



CHALMERS
UNIVERSITY OF TECHNOLOGY



Designing a Remote Supervisory Operation Interface for Autonomous Urban Passenger Ferries

A Human-Centred HMI Design Study for Shore-Based Monitoring, Communication, and Supervisory Command

Master's thesis in Industrial Design Engineering

PENGRUI YANG

DEPARTMENT OF INDUSTRIAL AND MATERIALS SCIENCE

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

MASTER'S THESIS 2026

Designing a Remote Supervisory Operation Interface for Autonomous Urban Passenger Ferries

A Human-Centred HMI Design Study for Shore-Based Monitoring,
Communication, and Supervisory Command

PENGRUI YANG



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Industrial and Materials Science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Designing a Remote Supervisory Operation Interface for Autonomous Urban Passenger Ferries
A Human-Centred HMI Design Study for Shore-Based Monitoring, Communication, and Supervisory Command
PENGRUI YANG

© PENG RUI YANG, 2026.

Supervisor: Fredrick Ekman, Industrial and Materials Science
Examiner: Lars-Ola Bligård, Industrial and Materials Science

Master's Thesis 2026
Department of Industrial and Materials Science
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Rendering of a conceptual Remote Operation Centre overlooking Gothenburg's riverfront, showing a multi-screen maritime control interface.

Typeset in L^AT_EX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Designing a Remote Supervisory Operation Interface for Autonomous Urban Passenger Ferries

A Human-Centred HMI Design Study for Shore-Based Monitoring, Communication, and Supervisory Command

PENGRUI YANG

Department of Industrial and Materials Science

Chalmers University of Technology

Abstract

This thesis explores the design of a Remote Operation Centre interface for supervising autonomous passenger ferries in Gothenburg. It argues that remote ferry operation is not continuous remote driving, but supervisory work based on mediated information, communication, and bounded intervention. Using a Research through Design approach, the project combined literature review, expert and stakeholder input, scenario reasoning, and iterative prototyping. The final outcome is a high-fidelity multi-screen ROC interface. The left screen supports fleet-level monitoring and priority recognition. The right screen supports selected-event diagnosis, communication, automation recommendation review, and supervisory response. The 24-inch vessel screens provide read-only awareness of individual vessels. The thesis contributes an interface-level design proposal for organising monitoring, communication, and supervisory command in autonomous urban ferry operation. The prototype is not a validated operational system. However, a formative internal review with two Cstrider participants indicated that the overall interface logic was understandable, while also identifying areas for refinement. The prototype can therefore serve as an evaluation-informed foundation for future expert review, simulator-based testing, and ROC interface development.

Keywords: Remote Operation Centre, autonomous ferries, maritime HMI, supervisory control, situation awareness, interaction design.

Acknowledgements

This master's thesis was carried out during the spring term of 2026 at the Department of Industrial and Materials Science at Chalmers University of Technology. I am a student from the Master Programme of Industrial Design Engineering. The project was conducted in relation to the Cstrider project and focused on the design of a remote operation centre interface for autonomous passenger ferry operations in Gothenburg.

I would like to begin by expressing thanks to my supervisor and examiner, Fredrick Ekman and Lars-Ola Bligård, for their support, guidance, and valuable feedback throughout the entire project. Their advice helped me clarify the direction of the thesis, improve the structure of the report, and strengthen the connection between the research findings and the final design outcome.

I would also like to thank Tobias Husberg for sharing valuable knowledge about the Cstrider project, autonomous ferry operation, and the practical context of remote operation centres. His feedback helped me better understand the operational logic behind the system and supported the development of a more realistic design proposal.

I extend my gratitude to all participants who took part in interviews, discussions, user tests, and evaluation sessions. I would also like to acknowledge the OpenBridge project for providing an open-source maritime design system that supported the visual and interaction design of the prototype. Finally, I would like to thank Sylvia Xie for her support, discussion, and encouragement during the thesis process.

Thank you!

Pengrui Yang, Gothenburg, June 2026

Contents

List of Figures	xiii
List of Tables	xv
1 Introduction	1
1.1 Background	1
1.2 Aim and Research Questions	2
1.3 Scope	3
1.4 Delimitations	4
2 Background and Literature Review	7
2.1 Autonomous and Remote Maritime Operation	7
2.2 Human Factors in Remote Operation	9
2.3 Challenges in Shore Control Centres	11
2.4 Literature-Based Design Implications	16
3 Methodology	19
3.1 Project Context	19
3.2 Design Problem	20
3.3 Research Through Design Approach	22
3.4 Expert and Stakeholder Input	22
3.5 Interview Procedure	23
3.6 Data Analysis	24
3.7 Design Process Overview	25
3.8 Methodological Limitations	26
4 Findings and Design-Oriented Analysis	29
4.1 Analytical Framework and Theme Development	29
4.2 Theme 1: Remote Supervision Requires Integrated Situation Recon- struction	31
4.3 Theme 2: Fleet Operation Requires Dynamic Priority Management .	32
4.4 Theme 3: Automation Must Be Explainable Before It Becomes Ac- tionable	33
4.5 Theme 4: Routine Operation Needs Clear Responsibility Boundaries .	34
5 Design Requirements and Principles	37

6	Design Development	41
6.1	Wireframes and Iterations	41
6.2	Information Architecture	43
6.3	Final Prototype	46
7	Evaluation and Design Review	61
7.1	Evaluation Aim	61
7.2	Evaluation Method	63
7.3	Evaluation Results	67
7.4	Design Revisions after Evaluation	69
8	Discussion	71
8.1	Design Contribution	71
8.2	Reflection on the Design and Research Process	73
8.3	Reflection on Remote Operation as Cognitive Work	75
8.4	Implications for ROC Interface Design	76
8.5	Ethics and Sustainability Reflections	78
8.6	Use of AI in the Thesis Process	79
8.7	Limitations	80
8.8	Future Work	81
9	Conclusion	85
	Bibliography	89
A	Appendix 1 - Interview and Stakeholder Input Material	I
A.1	Semi-Structured Interview Guide	I
A.2	Participant-Driven Scenario Reconstruction Template	II
A.3	Sketch from Participants	IV
A.4	Researcher Reflection on Interview Limitations	VII
B	Appendix 2 - Data Analysis and Theme Development	IX
B.1	Data Analysis Procedure	IX
B.2	Initial Coding Schema	X
B.3	Cross-Participant Coding Matrix	X
B.4	Revised Coding Schema and Design Translation	XII
B.5	Transition from Codes to Findings Themes	XIII
C	Appendix 3 - Design Process and Iterations	XV
C.1	Previous Project Version	XV
C.2	Wireframes	XVII
C.3	Early Concept Exploration	XVII
C.4	Prototype Screen Iteration	XVIII
D	Appendix 4 - Information Architecture and Command Logic	XXIII
D.1	Information Architecture Overview	XXIII
D.2	Command Scope and Logic	XXV

E	Appendix 5 - Visual Design System Support	XXIX
E.1	Colour and Status System	XXIX
E.2	Typography and Information Hierarchy	XXXI
E.3	Components and Interaction States	XXXIII
E.4	Icon Usage	XXXIV
F	Appendix 6 - AI Use Documentation	XXXVII
F.1	Prompting Approach	XXXIX
F.2	Use of Prompt Outputs	XL

List of Figures

3.1	Design process overview	19
3.2	Problem framing from literature-based challenges to the ROC inter- face design problem.	21
6.1	Early multi-screen ROC interface draft	41
6.2	Initial sketch wireframe	43
6.3	Multi-display screen strategy of the ROC prototype	50
6.4	Left-screen elements and functions	52
6.5	Right main screen: command workspace	56
6.6	Actor and interaction flow around the ROC interface	57
C.1	Previous version 1	XV
C.2	Previous version 2	XVI
C.3	Wireframes	XVII
C.4	AI-assisted visual exploration	XVII
C.5	AI-assisted final concept	XVII
C.6	Left screen iteration	XIX
C.7	Right screen iteration	XX
C.8	Screen layout iteration	XXI
D.1	Command scope in the ROC interface	XXV
E.1	Colour palette used in the final prototype, including base interface colours and semantic status colours for information, caution, warning, critical, and running states.	XXX
E.2	Typography styles used in the final prototype, including title, subtitle, underline, body, button, active body, label, and active label text styles.	XXXII
E.3	Selected component examples used in the final prototype, including status labels, vessel cards, command queue items, evidence panels, camera panels, information cards, and response bars.	XXXIV
E.4	Icon set used in the final prototype, including vessel, route, commu- nication, sensor, status, and command-related icons.	XXXV

List of Tables

1.1	Thesis scope and delimitations	5
2.1	Distinction between autonomy, remote operation, and thesis scope . .	9
3.1	Interview structure	24
4.1	Design-oriented coding framework	30
4.2	Scenario analysis timeline	31
5.1	Design requirements and principles	38
6.1	Information architecture of the final ROC prototype	45
6.2	Final prototype structure	46
6.3	Screen strategy of the final ROC prototype	48
6.4	Left-screen elements and functions	52
6.5	Right-screen components and functions	54
7.1	Evaluation aims	62
7.2	Scenario types used in the evaluation	65
7.3	SAGAT-inspired situation awareness questions	66
7.4	Summary of design revisions after evaluation	69
A.1	Semi-structured interview guide, Part 1.	I
A.2	Semi-structured interview guide, Part 2.	II
A.3	Participant-driven scenario reconstruction template.	II
A.4	Cognitive gap summary template.	IV
A.5	Researcher reflection on interview limitations.	VII
B.1	Initial coding schema developed from the interview material.	X
B.2	Cross-participant coding matrix.	X
B.3	Revised coding schema and design translation.	XII
B.4	Mapping from revised codes to Chapter 4 findings themes.	XIV
D.1	Information architecture of the final ROC interface prototype.	XXIV
D.2	Command level logic	XXVI
E.1	Colour and status system used in the final prototype.	XXXI
E.2	Typography hierarchy and application in the final prototype.	XXXIII

F.1	Summary of prompt use during the thesis process, Part 1.	XXXVIII
F.2	Summary of prompt use during the thesis process, Part 2.	XXXIX

1

Introduction

This chapter introduces the context, purpose and scope of the thesis. It first presents the background of autonomous passenger ferries and remote operation centres, with a focus on the emerging role of shore-based operators in future urban waterborne mobility. It then defines the design problem, the aim of the project, and the research questions that guided the work. Finally, the chapter clarifies the project scope and outlines the structure of the report.

1.1 Background

Urban mobility systems are under pressure to become more sustainable, accessible, and flexible. In Gothenburg, this challenge is closely connected to the river Göta älv. The river supports the identity and structure of the city, but it also creates a physical barrier between important urban districts. For this reason, cross-river mobility is not only a transport issue, but also an accessibility issue. Gothenburg's long-term transport strategy emphasises a shift towards walking, cycling, and public transport, while previous work on electrified and highly automated passenger ferries identifies ferry services as one possible way to improve cross-river connections and support a more flexible public transport system (City of Gothenburg, 2014; Ekman & Wallgren, 2024).

Highly automated and electric passenger ferries may make it possible to operate smaller vessels, increase service flexibility, and support both fixed shuttle services and more demand-responsive operation. However, automation does not remove the need for human involvement. Instead, it changes where human work takes place and what kind of work is required. In remote and autonomous maritime operations, part of the operational responsibility is moved from the vessel to a shore-based Remote Operation Centre (ROC). The operator is no longer physically located on the vessel and must understand the situation through mediated information, such as sensors, cameras, maps, alarms, system status, and communication channels (Man et al., 2015; Man et al., 2018; Sharma et al., 2025).

This shift creates a specific interaction design challenge. A ROC interface should not simply reproduce a traditional ship bridge on screens. The work domain is different. The operator may need to supervise several vessels, identify which situation requires attention, understand the behaviour and limits of automation, communicate with relevant actors, and make accountable decisions when the system needs human support. Research on remote maritime operation shows that situation awareness,

information quality, workload, automation transparency, and human-machine coordination are central challenges in this setting (Kari et al., 2018; Man et al., 2018; Sharma et al., 2025).

For small autonomous passenger ferries in an urban waterway, these issues become especially important because operation takes place close to passengers, docks, public spaces, other vessels, and changing traffic conditions. The operator’s role is therefore not treated as continuous remote driving. In this thesis, the role is understood as supervisory: monitoring fleet and vessel status, understanding abnormal events, communicating when needed, and issuing limited supervisory commands. Full manual teleoperation is outside the scope of this thesis.

This thesis responds to this challenge through the design of a ROC interface concept for Cstrider’s autonomous passenger ferry context. The design explores how information architecture, alert prioritisation, vessel status presentation, communication access, and command support can help a shore-based operator maintain an overview of the system and respond to abnormal events. The aim is not to design a complete autonomous ferry system, route planning system, or regulatory safety case. Instead, the thesis focuses on the human-machine interface as a critical layer between autonomous vessel behaviour and human supervisory responsibility.

1.2 Aim and Research Questions

The aim of this thesis is to design a Remote Operation Centre (ROC) interface concept for Cstrider’s autonomous urban passenger ferry system. The interface is intended to support shore-based operators in supervising multiple highly automated passenger ferries in an urban waterway context.

The thesis focuses on three connected operator activities: monitoring, communication, and supervisory command. Monitoring refers to maintaining awareness of fleet-level operation, selected vessel status, service mode, passenger-related states, and abnormal events. Communication refers to contacting or coordinating with relevant actors when a situation requires it. Supervisory command refers to bounded operator actions, such as holding, stopping, acknowledging, escalating, or approving a limited system recommendation, without taking over full manual control of the vessel.

Therefore, this thesis does not aim to design a full teleoperation interface or a direct vessel control system. Instead, it focuses on how the ROC interface can help operators understand the situation, identify priorities, understand automation behaviour, communicate when needed, and make appropriate supervisory decisions.

The main aim is formulated as follows:

The aim of this thesis is to design a ROC interface concept that supports shore-based operators in maintaining situation awareness, communicating

with relevant actors, and carrying out bounded supervisory commands when supervising multiple autonomous passenger ferries in an urban ferry context.

To address this aim, the thesis is guided by the following research questions:

RQ1. What operational and cognitive challenges need to be addressed when autonomous urban ferry operation is supervised remotely from a ROC?

RQ2. How can the information architecture of a ROC interface support operators in understanding fleet status, selected vessel status, service mode, and operational priority?

RQ3. How can a ROC interface support communication and supervisory response by making automation status, system recommendations, possible actions, and expected outcomes understandable and actionable?

The proposed interface concept is prepared for scenario-based evaluation and expert review. However, a full operational validation of safety, workload, or real-world performance is outside the completed scope of this thesis.

1.3 Scope

This thesis focuses on the design of a Remote Operation Centre (ROC) interface for Cstrider’s autonomous urban passenger ferry system. The design object is the shore-based operator interface. The interface is studied as a support system for remote supervision, rather than as a vessel control system or a full traffic management system.

The main design scope includes three operator activities: monitoring, communication, and bounded supervisory command. Monitoring refers to supporting the operator in understanding fleet-level operation, selected vessel status, service mode, task progress, system health, passenger-related states, and abnormal events. Communication refers to supporting contact and coordination with relevant actors, such as onboard staff, passengers, service personnel, or external support. Bounded supervisory command refers to limited operator actions, such as holding, stopping, acknowledging, escalating, requesting assistance, or confirming a system-recommended action when a situation requires human judgement.

The thesis considers both shuttle mode and on-demand mode as operational contexts for the interface. Shuttle mode refers to repeated operation along fixed or semi-fixed crossing corridors. On-demand mode refers to request-based operation where vessels may be assigned according to passenger demand. The interface design explores how these service modes can be represented and supervised in the ROC.

The prototype is a high-fidelity interface concept. It is used to explore information architecture, screen distribution, alert logic, vessel status representation, communication access, and command response patterns. The contribution of this thesis is limited to the interface design layer: how a shore-based operator can maintain situation awareness, understand priorities, communicate, and carry out bounded supervisory command in a remote multi-vessel ferry operation context.

1.4 Delimitations

This thesis does not design the vessel itself, the autonomous navigation system, the passenger-facing service, or the complete traffic management system. System recommendations shown in the interface are treated as outputs from underlying system functions, not as algorithms developed in this thesis.

The thesis does not develop fleet dispatch algorithms, route optimisation, timetable planning, passenger demand prediction, or autonomous navigation logic. These functions are treated as background system capabilities that may inform the interface, but they are not designed or evaluated as technical systems in this project.

Full manual teleoperation and direct vessel control are outside the scope. This includes joystick-based control, remote manoeuvring logic, low-latency control systems, manual docking, and emergency takeover hardware. These functions would require technical, regulatory, safety, and simulator-based work beyond the scope of this interaction design thesis. The project instead focuses on how operators understand system status, communicate with relevant actors, and carry out bounded supervisory commands without directly steering the vessel.

The prototype is not a deployable operational system. It does not include backend implementation, real sensor integration, cybersecurity, communication protocols, multi-display synchronisation, system reliability calculation, or full software engineering delivery.

Finally, this thesis does not provide regulatory certification, staffing procedures, operator training requirements, or a complete safety case for real-world autonomous ferry operation. These issues are treated as background constraints and future work.

Table 1.1: Thesis scope and delimitations

Area	In scope	Out of scope
Interface design	ROC operator interface, multi-screen layout, information architecture	Full software implementation
Operator activities	Monitoring, communication, bounded supervisory command	Continuous remote driving, joystick control, direct manoeuvring
Service modes	Representation of shuttle and on-demand operation	Dispatch algorithm, demand prediction, timetable optimisation
Vessel system	Displayed vessel status and system recommendations	Autonomous navigation logic, collision avoidance algorithms
Data and infrastructure	Conceptual status, alert, and communication representations	Real sensor integration, backend, cybersecurity, communication protocols
Safety and regulation	Interface-level responsibility boundaries	Certification, legal framework, complete safety case

2

Background and Literature Review

2.1 Autonomous and Remote Maritime Operation

Autonomous and remote maritime operation changes how work, responsibility, and information are distributed between vessel and shore. In conventional vessel operation, navigation and safety work are carried out on board through direct environmental observation, bridge instruments, vessel motion cues, and crew coordination. In highly automated operation, selected functions such as navigation support, monitoring, collision avoidance, route following, and task execution may be supported or performed by onboard automation. The human role is therefore not removed, but relocated and reconfigured.

A useful distinction needs to be made between autonomous, remotely operated, and unmanned vessels. An autonomous vessel can perform selected navigation or operational tasks through automated systems, but it may still have crew or safety personnel on board. An unmanned vessel has no people on board, but it does not necessarily operate fully autonomously, since it may still be supervised or controlled from a shore-based centre. Kari et al. (2018) make this distinction clear and describe remote ship operation as a situation where an operator manages a vessel from a distance through sensor-based and technical information rather than direct sensory contact. For this thesis, this distinction is important because the Cstrider context is not treated as a case of complete human removal. It is understood as a human-automation system involving autonomous vessels, onboard safety roles, passengers, shore-based operators, and the surrounding urban waterway.

The development of Maritime Autonomous Surface Ships (MASS) also shows that remote operation is becoming a formal operational and regulatory topic, rather than only a speculative design concept. The IMO adopted the non-mandatory International Code of Safety for Maritime Autonomous Surface Ships (MASS Code) in May 2026. The Code currently applies to cargo ships under SOLAS Chapter I and takes effect from 1 July 2026. In this thesis, this development is used only as evidence that remote and autonomous maritime operation is becoming a more formal regulatory topic, not as a directly applicable certification framework for the Cstrider passenger ferry concept.

Remote operation also changes the operator's access to the operational situation. Since the operator is not physically located on the vessel, the vessel and its surroundings must be understood through mediated information, such as sensors, cameras, maps, alarms, communication systems, and status displays. This makes the interface a central part of remote operation. It must translate vessel state, environmental conditions, automation behaviour, and abnormal events into information that the operator can understand and act upon.

The Shore Control Centre or Remote Operation Centre is therefore more than a monitoring screen. Studies from the MUNIN project describe shore-based operators monitoring several vessels through dashboards, electronic sea charts, conning displays, weather information, alarm flags, and vessel mode indicators (Man et al., 2015). This example shows that remote operation requires fleet overview, vessel-level detail, alert management, mode awareness, and escalation support. It also shows that the ROC is a socio-technical work environment where human roles, digital interfaces, communication, and organisational responsibility need to be considered together.

Different levels of remote intervention also need to be separated. A shore-based operator may monitor vessel status, receive system recommendations, update high-level operational instructions, issue bounded commands, or, in more demanding cases, take over direct control. These levels are not the same. This thesis focuses on monitoring, communication, and bounded supervisory command. It therefore relates to high-level supervision and event-based intervention, rather than joystick-based teleoperation, manual docking, or continuous remote manoeuvring.

This distinction is central to the design scope. A ROC interface for autonomous passenger ferries should not simply copy a traditional ship bridge onto shore-based screens. It should support a different work domain: remote supervision of automated vessels, where the operator must maintain situation awareness, understand automation behaviour, identify operational priorities, communicate with relevant actors, and intervene only when human judgement is required.

Table 2.1: Distinction between autonomy, remote operation, and thesis scope

Concept	Meaning in this thesis	Relation to design scope
Autonomous operation	The vessel can perform selected navigation or operational tasks through onboard automation	Treated as system behaviour shown through the interface
Remote operation	Shore-based operators supervise or operate vessels through mediated information	Forms the work context of the ROC interface
Unmanned vessel	A vessel with no people on board. It may still be remotely controlled or supervised	Not assumed as the default Cstrider case
Supervisory command	Bounded operator action such as hold, stop, acknowledge, escalate, or confirm a recommendation	In scope
Direct teleoperation	Continuous remote manoeuvring, steering, joystick control, or manual docking	Out of scope

2.2 Human Factors in Remote Operation

Remote operation is not only a technical shift from vessel to shore. It also changes how operators perceive information, build situation awareness, trust automation, manage workload, and make decisions. In a traditional bridge environment, operators can combine visual observation, sound, vessel motion, environmental cues, instruments, and team communication. In a Remote Operation Centre (ROC), many of these cues are replaced or mediated by sensors, cameras, maps, alarms, status displays, and communication systems. The operator still needs to make safety-related judgements, but the information used for these judgements becomes indirect, digital, and distributed.

Situation awareness is one of the most important human factors in this context. Endsley (1995) defines situation awareness as the perception of relevant elements in the environment, the comprehension of their meaning, and the projection of their future status. In remote maritime operation, all three levels are affected. The operator must perceive vessel position, movement, system status, surrounding traffic, passenger-related states, and alerts through interface representations. The operator must then understand what this information means, for example whether an alert requires intervention or whether a vessel is deviating from its expected task. Finally, the operator must project what may happen next, such as whether a vessel will enter a risk area, whether a delay will affect service operation, or whether an automation recommendation may create later conflicts.

For this reason, the aim of the ROC interface should not be to display as much information as possible. It should support actionable situation awareness. The interface needs to help the operator understand the current state, why it matters, what may happen next, and what action is available. Man et al. (2018) argue that, in shore control centre contexts, interface design directly affects how operators obtain and maintain situation awareness and how they make subsequent decisions. This supports an information architecture where fleet overview, vessel status, alert logic, mode awareness, and decision support are connected rather than presented as isolated interface elements.

Remote operation also creates workload and attention challenges. Workload does not only come from the amount of information shown. It also comes from switching attention between sources, comparing different displays, and integrating distributed information into one coherent operational picture. This becomes more critical in multi-vessel supervision. Veitch et al. (2024) found that supervisory performance was reduced when operators supervised more vessels, had less available response time, or lacked decision support. Therefore, a ROC interface should not treat all vessels or events with equal visual priority. It needs to guide attention toward the vessel, event, or decision that matters most.

Automation further changes the operator's role. In highly automated maritime operation, the operator is usually not expected to control the vessel manually at all times. Instead, the operator supervises automation and intervenes when necessary. Supervisory control theory describes the human role as setting goals, monitoring automation behaviour, receiving feedback, and intervening when the system reaches its limits (Sheridan, 1992). For this thesis, this means the operator should be supported in understanding what the automation is doing, why it recommends an action, whether human confirmation is required, and what consequence an operator command may have.

Trust and transparency are closely connected to this supervisory role. If automation behaviour is unclear, the operator may over-rely on the system, under-rely on it, repeat unnecessary checks, or react too late. Calibrated trust requires the operator to understand system capability, limitation, uncertainty, and evidence quality (Lee & See, 2004). In interface terms, this means that automation status, system intent, recommendation rationale, risk level, available actions, and responsibility boundaries should be visible at the point of decision.

Decision-making in remote operation therefore involves both uncertainty and responsibility. Sharma et al. (2025) show that remote operators may face difficulties in judging information quality, dealing with data delay, and working without direct sensory cues. The interface should reduce the distance between detection, understanding, action selection, and action confirmation. In this thesis, this provides the theoretical basis for alert prioritisation, vessel detail views, automation status, communication access, and the decision response bar. These elements are intended to help operators identify what has happened, why it matters, what the system recommends, and what bounded supervisory action is possible.

2.3 Challenges in Shore Control Centres

The previous section introduced general human factors in remote operation. This section focuses more specifically on challenges that appear in Shore Control Centres and Remote Operation Centres for maritime operation. These challenges are important for this thesis because they translate directly into interface design problems. The interface must not only display vessel data. It must help the operator compensate for reduced direct perception, integrate fragmented information, prioritise alerts, supervise several vessels, and understand when automation requires human support.

2.3.1 Loss of Ship Sense

One fundamental challenge in Shore Control Centres and Remote Operation Centres is the loss of ship sense. Ship sense refers to the practical and embodied understanding that experienced ship handlers develop through physical presence, vessel motion, visual observation, sound, environmental change, instrument readings, and professional experience. It is not only knowledge of the vessel's position. It is an experienced understanding of how the vessel behaves in relation to the waterway, weather, traffic, task, and manoeuvring situation (Prison et al., 2013).

In a traditional bridge environment, ship sense is built through continuous interaction between the operator, the vessel, the environment, and the bridge team. Operators can combine instrument information with direct cues such as available water, wave direction, vessel movement, vibration, acceleration, engine response, and surrounding traffic. These cues help them judge whether the vessel is behaving normally, whether the current action is effective, and how the situation may develop.

In remote operation, many of these cues are weakened, delayed, transformed, or removed. Man et al. (2015) argue that shore-based operators are geographically separated from the vessel and must mainly rely on digital screen-based information. As a result, many cues used in onboard navigation and decision-making are missing, which can make it harder to maintain situation awareness and the sense of harmony between vessel, operator, and environment. Man et al. (2018) also identify deprivation of ship sense, perceptual limitations, decision-making latency, and automation bias as important human factor issues in remote ship monitoring.

Similar issues appear in remote pilotage. Sharma et al. (2025) show that remote pilots may have difficulty judging information quality, perceiving environmental change, handling data synchronisation, and working through two-dimensional screen-based information. Although remote pilotage is not the same as autonomous ferry supervision, the findings are relevant because both situations involve a shift from physical co-location to shore-based interpretation of mediated information.

For Cstrider's autonomous urban ferry context, loss of ship sense has specific implications. The vessels may be smaller, slower, and operating over shorter distances

than deep-sea vessels, but the environment is still dynamic. The ROC operator needs to understand vessel movement, docking and departure phases, passenger-related states, nearby traffic, berth conditions, service mode, and abnormal events. If the interface only presents isolated values, separate camera feeds, or disconnected status indicators, the operator may still need to reconstruct the situation manually. This can increase attention switching and make the operator more reactive.

Therefore, the design goal is not to recreate full onboard ship sense in the ROC. That would be unrealistic because bodily, auditory, and situated environmental cues cannot be fully reproduced through an interface. Instead, the ROC interface should support a mediated operational understanding. It should connect vessel position, motion, task phase, automation state, alert rationale, and near-future development in a structured way. In this thesis, loss of ship sense therefore informs the need for integrated situation views, temporal information, automation transparency, and a clear link between fleet-level overview and vessel-level detail.

2.3.2 Cognitive Workload and Fragmented Information

Another central challenge in Shore Control Centres is the cognitive workload created by fragmented information. Remote operators do not build an operational picture from one single source. They need to interpret maps, vessel status panels, sensor feeds, camera views, weather information, passenger-related data, route information, alarms, communication channels, and automation recommendations. Even when each source is accurate, the operator still needs to understand how these pieces relate to each other and what they mean for the current task, risk level, service impact, and possible action.

This creates an integration workload. In the MUNIN Shore Control Centre prototype, operators monitored several autonomous vessels through multiple dashboards, an electronic sea chart, conning displays, weather information, alarm flags, and vessel mode indicators (Man et al., 2015). Such an environment requires the operator to switch between display areas, compare information layers, and connect local vessel data with the wider operational situation. The main issue is therefore not only information overload. It is also the effort needed to turn distributed information into a coherent and actionable understanding.

This problem becomes more important in remote operation because direct perception is reduced. Sharma et al. (2025) show that remote operators may need to rely on technical systems and communication to judge information quality, confirm conflicting information, and compensate for the lack of physical co-location. Remote operation may also include long periods of low activity followed by short periods of high-pressure judgement. The interface must therefore support both continuous monitoring and rapid attention recovery when an abnormal event occurs.

In Cstrider’s urban ferry context, fragmented information also includes service-related information. The operator needs to understand not only vessel safety, but also service mode, passenger demand, berth state, fleet distribution, boarding sta-

tus, and abnormal events. If these elements are separated across unrelated interface areas, the operator may only see local data and fail to form a fleet-level or service-level understanding quickly. This is especially problematic in on-demand operation, where vessel assignment and service priority can change more frequently than in a fixed shuttle service.

Therefore, the ROC interface should organise information according to the operator's cognitive task. When an event occurs, the interface should connect the vessel, location, cause, affected service, urgency, system recommendation, communication need, and available action. At the fleet level, it should help the operator understand which vessels are operating normally, which are delayed, which are assigned to tasks, and which require attention. In this thesis, this challenge informs the need for clear screen distribution, consistent status hierarchy, event-centred information grouping, and direct links between fleet overview and vessel-level detail.

2.3.3 Alarm and Alert Design

Alarm and alert design is a critical issue in Shore Control Centres and Remote Operation Centres. Since remote operators cannot continuously perceive field conditions through direct sensory cues, abnormal events often need to be communicated through alarms, warnings, notifications, or decision-support prompts. The purpose of an alert is therefore not only to indicate that something is wrong. It should help the operator understand what has happened, why it matters, how urgent it is, what it affects, and what response is possible.

In the MUNIN Shore Control Centre prototype, colour-coded flags were used to distinguish between normal and abnormal vessel states. A yellow flag indicated a non-critical abnormality that required operator attention, while a red flag indicated a critical situation that required immediate investigation and corrective action (Man et al., 2015). This demonstrates the value of alert prioritisation in remote monitoring. However, colour or severity level alone is not sufficient. The operator also needs to understand why the alert appears, whether the information is reliable, what operational layer is affected, whether the situation is developing, and what response is expected.

Alarm design is therefore closely linked to decision support. Man et al. (2018) argue that alerting the operator and providing relevant resources for decision support are key aspects of automation in remote ship operation. In a ROC interface, an alert should support the operator's path from detection to diagnosis, action selection, and action verification. A useful alert should answer several practical questions: what happened, where it happened, what caused it, what it affects, how urgent it is, what the system recommends, and what the operator can do next.

This is also consistent with the broader maritime concern for structured alert handling. Bridge Alert Management standards emphasise the need to improve the handling, distribution, and presentation of alerts on the bridge (International Maritime Organization, 2010). Although this thesis does not design a certified bridge

alert management system, this principle is relevant for ROC interface design. Alerts should be prioritised, understandable, and connected to the operator’s task rather than presented as isolated technical messages.

In the Cstrider context, alerts may concern different operational layers. Some alerts relate to immediate vessel safety, such as obstacle detection, unsafe departure conditions, docking uncertainty, sensor degradation, or critical battery state. Others relate to service operation, such as delayed departure, berth congestion, passenger request surge, or on-demand assignment conflict. If all alerts are presented with the same visual logic, the operator may not quickly understand the nature of the problem or the expected response.

For this thesis, alert design is therefore organised around severity, cause, impact, and actionability. Severity indicates the risk level and urgency. Cause explains why the alert was triggered. Impact clarifies what vessel, service, passenger state, or operational process may be affected. Actionability clarifies whether the operator needs to monitor, acknowledge, communicate, escalate, or carry out a bounded supervisory command. In the final prototype, alert cards and command actions should form one continuous decision-support flow rather than two separate interface elements.

2.3.4 Multi-Vessel Monitoring

Multi-vessel monitoring is a central challenge in autonomous ferry ROC design. At first, it may seem to be a display problem: how to show several vessel cards, dashboards, camera feeds, or routes at the same time. However, the more important issue is how the operator can understand which vessel needs attention, why it matters, and how to move from fleet-level overview to vessel-level detail under limited time and attention.

Early Shore Control Centre concepts already explored one-to-many supervision. In the MUNIN Shore Control Centre prototype, one operator was expected to monitor several unmanned vessels through multiple dashboards, electronic charts, conning displays, weather information, alarm flags, and mode indicators (Man et al., 2015). This illustrates the operational ambition behind shore-based fleet supervision. However, it does not mean that adding more vessels is only a matter of adding more display panels. The number of vessels affects how attention is distributed, how quickly abnormal states are detected, and how easily the operator can maintain an overview of the system.

This issue is also supported by recent empirical work on remote supervisory control. Veitch et al. (2024) note that the question of “how many vessels per operator” has existed since the early stages of autonomous ship research, while empirical evidence remains limited. Their simulator study found that supervisory performance was affected by the number of vessels, available response time, and the availability of decision support. In particular, performance was reduced when operators supervised three vessels instead of one. This suggests that multi-vessel supervision is not only

a display capacity issue, but also an attention allocation and decision-support issue.

In the Cstrider context, multi-vessel monitoring includes both safety and service dimensions. The operator needs to understand which vessels are operating in shuttle corridors, which are responding to on-demand tasks, which are docking, boarding, departing, repositioning, or waiting, and which vessels require human attention. These states should not be treated as equal, because they have different effects on safety, passenger service, and fleet balance.

Therefore, the core design challenge is priority structure. A ROC interface should help the operator identify which vessels are operating normally, which are delayed, which are at higher risk, which affect service operation, and which require a decision. This requires more than repeating the same information for each vessel. It requires a hierarchy that connects fleet status, vessel comparison, alert priority, and selected-vessel detail.

A multi-vessel interface should also avoid giving all vessels the same information density. Normal vessels should remain visible but low-interruption. Vessels with abnormal states should become more visually prominent and should show cause, impact, and recommended action. Vessels that may need future attention should show trends or upcoming events before they become urgent. In this way, multi-vessel monitoring moves from static display to dynamic prioritisation.

For this thesis, multi-vessel monitoring supports the need for a clear fleet overview, visible vessel priority, distinguishable service modes, and a strong connection between overview and selected-vessel detail.

2.3.5 ROC as a New Work Domain

A Shore Control Centre or Remote Operation Centre should not be understood as an onshore copy of a traditional ship bridge. If the ROC is treated only as a “remote bridge”, the interface may simply reproduce bridge instruments, camera views, radar displays, and control panels. This would ignore the fact that remote operation changes the operator’s work pattern, information access, responsibility boundaries, and relationship with automation.

Kari et al. (2018) argue that remote ship operations involve complex human-human and human-machine relationships. These relationships affect system performance and require the human to be treated as an integrated part of the remote operation system. Man et al. (2018) also describe remote ship monitoring as a distributed ship-shore context, where situation awareness, observability, understandability, and decision-making are shaped by both system structure and interface design. This means that ROC design is not only about transmitting vessel information to shore. It is about supporting human understanding and judgement in a distributed system.

This new work domain also changes the operator's role. The ROC operator is not a traditional onboard officer and not a continuous remote helmsman. Instead, the operator supervises automated vessels, monitors multiple operational states, interprets system recommendations, communicates with relevant actors, and intervenes when human judgement is required. The work is therefore closer to supervisory control than direct manual control.

For Cstrider's autonomous urban ferry context, the ROC also includes a service-system dimension. The operator needs to understand not only vessel safety, but also shuttle operation, on-demand tasks, berth status, passenger-related information, fleet balance, and service disruption. Therefore, the interface must support both maritime safety reasoning and urban mobility service reasoning.

For this thesis, treating the ROC as a new work domain supports the central design direction: the interface should help the operator maintain overview, understand priority, communicate in context, and carry out bounded supervisory commands without becoming a direct vessel control interface.

2.4 Literature-Based Design Implications

The literature review shows that a ROC interface should not be designed as a remote copy of a traditional ship bridge. Remote operation changes how operators perceive the vessel, maintain situation awareness, supervise automation, communicate with others, and intervene when needed. Therefore, the design challenge is not only to present vessel data, but to support human understanding and judgement in a remote, multi-vessel, and highly automated work context.

First, the interface should support actionable situation awareness. Vessel position, movement, task phase, service mode, alerts, automation behaviour, and available actions should not be shown as isolated data points. They should form an operational picture that helps the operator understand what is happening, why it matters, and what may happen next.

Second, the interface should reduce manual situation reconstruction. Since direct ship sense is weakened in remote operation, the operator should not need to combine separate camera feeds, status values, alarms, maps, and service information manually. The interface should group information around the vessel, event, task phase, and required response.

Third, the interface should prioritise attention across vessels. Multi-vessel supervision is not only a display capacity problem. It is an attention and decision-support problem. The fleet view should help the operator distinguish normal vessels, delayed vessels, vessels at risk, service-critical vessels, and vessels requiring human action.

Fourth, the interface should make automation behaviour understandable. The operator needs to know what the system is doing, why a recommendation is made, how urgent the situation is, and what consequence an operator action may have.

Automation should therefore be presented with status, rationale, uncertainty, and clear responsibility boundaries.

Fifth, alerts should be connected to response. Alert design should not stop at colour labels or severity levels. A useful alert should explain what happened, where it happened, what caused it, what it affects, and what can be done next. In this thesis, alert cards and command actions are therefore treated as one decision-support flow.

Finally, for Cstrider's ROC, this system means representing vessel safety, shuttle operation, on-demand tasks, berth status, passenger-related information, fleet balance, and service disruption. These implications inform the later design requirements and final ROC interface concept.

3

Methodology

This chapter describes the research and design process used in this thesis. The work was conducted as a design-oriented study, where literature, user research, synthesis, concept development, prototyping, and evaluation were used together to develop and assess a high-fidelity ROC HMI concept. The purpose of the methodology was not to test a fully operational system, but to build a grounded design proposal for remote supervision, communication, and supervisory command in the context of autonomous passenger ferry operation.

The process followed six main phases: literature review, user research, synthesis, concept development, prototyping, and evaluation. These phases were not completely linear. Some design decisions were revisited when new insights emerged from stakeholder feedback, literature, or prototype development. However, the structure provides a clear overview of how the project moved from domain understanding to final design evaluation.



Figure 3.1: Design process overview

3.1 Project Context

This thesis is conducted in collaboration with Cstrider and focuses on the design of a Remote Operation Centre (ROC) interface for small autonomous passenger ferries. The project is situated in the context of future urban water transport in Gothenburg, where electrified and highly automated ferries have been explored as a possible way to improve cross-river mobility, increase accessibility, and support more flexible public transport services. Previous work on automated passenger ferries in Gothenburg identifies central river connections, such as Lindholmen, Stenpiren, Järntorget, and Operan, as relevant areas for future ferry development (Ekman & Wallgren, 2024). These local mobility questions form the background of the thesis, but they are not the main design object.

The design object is the shore-based operator interface. The thesis therefore does not focus on route planning, transport policy, ticketing, passenger service design, or the design of the autonomous vessel itself. Instead, it examines how a ROC interface can support human supervision when several autonomous ferries are operating in an urban waterway. This includes maintaining a fleet-level overview, understanding selected vessel status, noticing abnormal events, communicating with relevant actors, and issuing limited supervisory commands when needed.

The operational concept considered in the project includes both shuttle and on-demand ferry services. Shuttle operation refers to repeated service between selected stops. In this mode, the ROC needs to support awareness of regular vessel movement, dock readiness, boarding state, and repeated crossings. On-demand operation refers to a more flexible service logic where vessels may respond to passenger demand. In this mode, the ROC also needs to support awareness of request distribution, vessel assignment, service priority, and changes in fleet availability.

A key project assumption is that the ROC operator is not a continuous remote driver. Based on the current project understanding, onboard crew or safety personnel are responsible for local departure readiness. In normal operation, the ROC operator does not need to approve every departure. Instead, the operator supervises the wider system and intervenes when vessel status, traffic conditions, dock conditions, passenger-related states, or system alerts indicate that human attention is required. This means that the operator's role is closer to supervision, coordination, and exception handling than direct vessel control.

The final design outcome is therefore a multi-screen ROC interface concept. It supports fleet-level monitoring, selected-vessel detail, alert prioritisation, communication access, and bounded command support. The interface is not intended to reproduce a traditional ship bridge on shore. It is designed to support the specific work of supervising autonomous passenger ferries from a remote operation setting.

3.2 Design Problem

The design problem of this thesis concerns how a Remote Operation Centre (ROC) interface can support human supervision of a small autonomous passenger ferry fleet. In conventional vessel operation, operators build situation awareness through direct visual, auditory, bodily, environmental, and instrument-based cues. In a remote operation context, this direct connection is reduced. The ROC operator must instead understand the vessel and its surroundings through mediated information, such as sensors, cameras, maps, alarms, vessel status data, automation feedback, and communication systems. This makes the interface a central part of the operator's ability to understand what is happening, judge the quality of information, and decide whether action is needed (Man et al., 2018; Sharma et al., 2025).

For Cstrider's autonomous urban ferry context, the problem is not simply to show more data. The operator needs to understand several vessels, different service modes,

passenger-related states, dock situations, traffic conditions, system health, and automation behaviour. If these elements are fragmented across screens or hidden in separate layers, the operator may need to reconstruct the situation manually. This can increase attention switching, delay comprehension, and make it harder to decide which vessel or event requires priority.

Multi-vessel supervision makes this issue more critical. Each vessel may have a different task, risk level, operational state, service impact, and need for action. Research on remote supervisory control shows that performance can be affected by the number of vessels, available response time, and the availability of decision support (Veitch et al., 2024). Therefore, the ROC interface should not only present several vessels side by side. It must help the operator understand which vessel needs attention, why it matters, and what action is possible.

Automation also changes the operator’s work. The operator is not expected to control the vessel continuously, but must still monitor, verify, interpret, communicate, and intervene when necessary. If automation behaviour is unclear, the operator may over-rely on the system, under-rely on it, repeat unnecessary checks, or react too late. This means that the interface must make automation status, system intent, uncertainty, and possible supervisory actions visible at the point of decision (Lee & See, 2004).

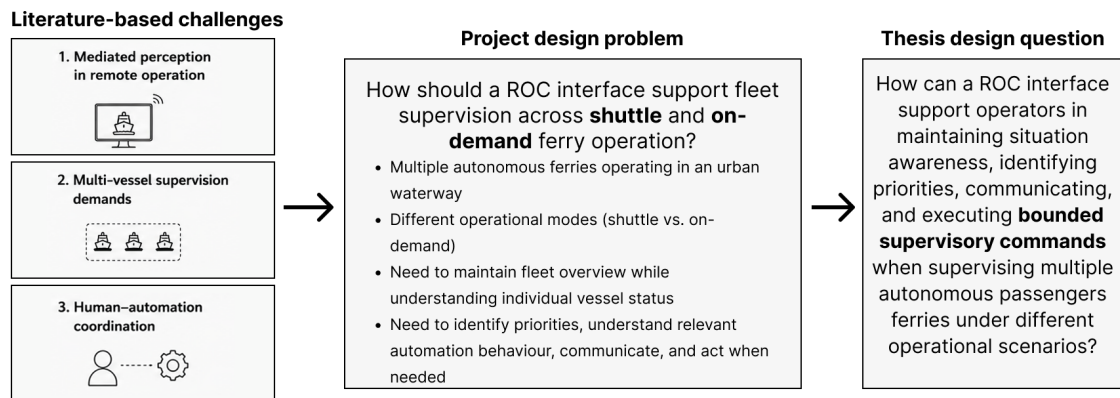


Figure 3.2: Problem framing from literature-based challenges to the ROC interface design problem.

Therefore, the central design problem is:

How can a ROC interface for an autonomous urban ferry fleet support operators in maintaining situation awareness, identifying operational priorities, understanding automation behaviour, and carrying out monitoring, communication, and supervisory command in a remote, multi-vessel, and highly automated operational context?

This thesis therefore does not focus on full manual teleoperation or direct vessel control. Instead, it focuses on supervisory interaction: fleet monitoring, vessel-

level understanding, abnormal event handling, communication access, and bounded command support.

3.3 Research Through Design Approach

This thesis adopts a Research through Design approach as its overall research orientation. In this approach, design is not treated only as the production of a final solution. It is also used as a way to explore an under-defined problem space, make design questions more concrete, and generate design knowledge through artefacts and reflection. Frayling (1993) distinguished between research into, for, and through art and design, while later work in interaction design describes prototypes and design artefacts as important means for inquiry in complex design situations (Stappers & Giaccardi, 2017; Zimmerman et al., 2007).

This approach is suitable for this thesis because the ROC interface for autonomous urban ferries is still an emerging design context. The operator role, service modes, information structure, and supervisory command logic are not yet stable industry patterns. The thesis therefore does not aim to optimise an existing interface. Instead, it uses literature review, project context analysis, expert input, scenario reasoning, and iterative prototyping to explore what kind of interface may support remote multi-vessel supervision.

The main design artefacts in this thesis include the information architecture, interaction flows, wireframes, screen layout decisions, and high-fidelity prototype. These artefacts are used to investigate how fleet-level information, selected-vessel detail, alert handling, communication access, and bounded supervisory commands can be organised across a multi-screen ROC environment. Through this process, the design problem moves from a broad question of remote operation support to more specific questions about priority recognition, automation transparency, situation reconstruction, and action support.

The contribution of this approach is interface-level design knowledge. The prototype is not a deployable operational system and does not include real vessel-control algorithms, backend dispatch logic, real sensor integration, regulatory certification, or full manual teleoperation. Instead, it is used to clarify how a ROC interface can organise information and support supervisory decision-making in the context of autonomous urban ferry operation.

3.4 Expert and Stakeholder Input

This thesis used expert interviews, stakeholder discussions, and formative design feedback to understand the practical design needs of Cstrider’s autonomous urban ferry ROC interface. Since this is an emerging work context, there are few users with direct hands-on experience of operating this exact type of system. Therefore, the study combined input from maritime professionals, Cstrider stakeholders,

supervisors, and HMI/design experts to explore possible operator tasks, cognitive challenges, service constraints, and interface needs.

The input was collected through semi-structured interviews and discussions. A semi-structured format was used because it allowed predefined topics to be covered while also leaving room for participants to introduce issues based on their own professional experience. This is appropriate when the research topic is exploratory and when the researcher needs both comparability across sessions and flexibility during the discussion (Kallio et al., 2016).

The discussions focused on concrete operational situations rather than abstract feature preferences. Key topics included departure readiness, docking, boarding, route monitoring, service delay, on-demand request handling, alert response, automation recommendations, passenger communication, and abnormal event handling. These topics were selected because they connect directly to the thesis scope: monitoring, communication, and bounded supervisory command.

In total, four interviews or discussions were conducted. Each session lasted approximately 45 minutes and was conducted in person. Depending on participant consent and the situation, the collected material included notes, audio recordings, screenshots, and design feedback. Participant information was anonymised or described in general terms in the report.

The purpose of this input was not to produce statistically generalisable findings. Instead, it was used to identify recurring design concerns and translate them into interface requirements and prototype decisions. Particular attention was given to how operators may build understanding from multiple information sources, how they may identify vessel priority across a fleet, how automation recommendations should be explained, and how communication and supervisory command should be supported.

The input helped connect the literature review to the Cstrider context. It clarified several practical interface needs, including cross-checking between information sources, visible departure-readiness evidence, fleet-level priority, alert explanation, service-mode awareness, communication access, and bounded supervisory action. These insights informed the later design requirements, information architecture, scenarios, and high-fidelity prototype.

3.5 Interview Procedure

The interview procedure was designed to move from participants' professional experience to the Cstrider ROC context and prototype feedback. Since the project concerns an emerging work domain, the interviews were not used to test a finished operational system. Instead, they were used to explore future operator needs, operational reasoning, and formative design feedback.

Before each interview, the researcher introduced the thesis background, the Cstrider

project context, the purpose of the session, the expected duration, and how the collected material would be used. Participants were also informed that the study focused on ROC interface design rather than the evaluation of their personal competence. Depending on participant consent and the situation, notes, audio recordings, screenshots, and design feedback were collected.

Table 3.1: Interview structure

Part	Focus	Purpose
1. participant background	Maritime, ferry operation, automation, remote operation, HMI, or design experience	To understand the participant's professional perspective
2. Scenario reconstruction	Real or typical situations such as departure, docking, boarding, route monitoring, or abnormal events	To understand how participants recognise situations, confirm information, make decisions, and follow up actions
3. Projection to ROC operation	How the same situation would change if the operator were located on shore	To identify lost sensory cues, required interface support, and cross-source confirmation needs
4. Cstrider-specific scenarios	Shuttle monitoring, on-demand request handling, docking, service mode changes, alert response, and passenger communication	To explore what information should be visible and when the operator should monitor, communicate, hold, escalate, or carry out a supervisory command
5. Prototype feedback	Early screen layouts or high-fidelity interface concepts, depending on design phase	To collect feedback on priority recognition, automation understanding, communication access, and command consequences

This structure allowed the interviews to connect practical operational experience with the specific design scope of the thesis: monitoring, communication, and bounded supervisory command in a remote multi-vessel ferry context.

3.6 Data Analysis

The interview notes, meeting summaries, prototype feedback, screenshots, and recordings were analysed through a hybrid thematic analysis. The analysis combined deductive and inductive coding. Deductive coding was guided by the literature review and the thesis focus, including situation awareness, workload, decision-making, human-automation interaction, trust, alert handling, and supervisory control. Inductive coding was used to capture issues that emerged from the Cstrider context, stakeholder discussions, scenario reconstruction, and prototype feedback.

Thematic analysis was chosen because it supports the identification of patterns across qualitative material while remaining flexible enough for exploratory design research (Braun and Clarke, 2006). A hybrid approach was suitable because the analysis needed to connect existing human factors literature with project-specific design concerns. This allowed the study to use theory-informed concepts while still remaining open to issues that appeared during interviews and design feedback (Fereday and Muir-Cochrane, 2006).

The analysis followed two coding cycles. In the first cycle, the material was organised according to the structure of the interview procedure: participant background, scenario reconstruction, projection to ROC operation, Cstrider-specific scenarios, and prototype feedback. Attention was given to how participants described situation recognition, information confirmation, decision-making, action, and follow-up. This helped identify what information may be needed before, during, and after a supervisory decision.

In the second cycle, the initial codes were grouped into design-oriented themes. These themes focused on recurring interface problems and opportunities, such as fragmented information, cross-source confirmation, weak prioritisation, unclear automation state, uncertain command consequence, departure-readiness evidence, passenger communication, alert-to-action logic, and multi-vessel priority judgement. Themes that could not be connected to the thesis scope or the current prototype were not treated as core design findings. Instead, they were treated as limitations or future work.

The purpose of the analysis was not to produce statistically generalisable results. The number of participants was limited, and the ROC for autonomous urban ferries is still an emerging work domain. Therefore, the findings should be understood as design-oriented insights for conceptual interface development. To improve credibility, the analysis was checked against three sources: the literature review, Cstrider stakeholder feedback, and prototype iteration. If a finding could not be represented in the current interface concept, it was not claimed as an implemented design contribution.

3.7 Design Process Overview

The design process connected the literature review, expert and stakeholder input, thematic analysis, and iterative prototyping. The process followed a human-centred and Research through Design logic: the problem was first framed through literature and project context, then translated into design requirements, explored through interface concepts, and refined into a high-fidelity multi-screen prototype.

The process was organised into four connected phases:

3.7.1 Problem framing

The first phase identified the main operational and cognitive challenges of remote ferry supervision. This included literature on situation awareness, loss of ship sense, fragmented information, multi-vessel monitoring, alert design, automation transparency, and supervisory control. It also included project-specific assumptions about Cstrider’s shuttle and on-demand ferry operation.

3.7.2 Design translation

The second phase translated the literature review and qualitative input into design implications and requirements. These focused on fleet-level overview, vessel priority, selected-vessel detail, alert-to-action logic, communication access, automation explanation, and bounded supervisory command.

3.7.3 Iterative prototyping

The third phase explored these requirements through sketches, wireframes, screen layout experiments, and high-fidelity interface prototypes. The iterations were used to test how information could be distributed across a multi-screen ROC environment and how the operator could move from fleet overview to vessel-level understanding and response.

3.7.4 Prototype Finalisation and Evaluation

The final phase assessed the prototype against the thesis aim, research questions, design requirements, and planned evaluation criteria. Since full operational testing was outside the completed scope, the prototype was prepared for scenario-based evaluation and expert review rather than validated as a deployable system.

This process reflects the human-centred design principle that interactive systems should be developed with attention to users, tasks, requirements, and human factors knowledge (International Organization for Standardization, 2019). In this thesis, the process was also informed by recent ROC design research, where remote operation centre concepts are developed through iterative, case-based, and prototype-driven methods (Gernez et al., 2026).

3.8 Methodological Limitations

The methodology has several limitations. First, the number of interviews and stakeholder discussions was limited. This was partly because the ROC for autonomous urban passenger ferries is still an emerging work context, and there are few people with direct experience of operating this exact type of system. The participants therefore provided informed professional and project-based perspectives, but they should not be treated as a representative sample of future ROC operators.

Second, the interviews and prototype feedback were formative rather than evaluative. They were used to identify design concerns, explore operational reasoning, and refine the interface concept. They did not test whether the final prototype improves situation awareness, reduces workload, increases decision confidence, or improves operational performance. Such claims would require a separate evaluation with defined scenarios, participants, measures, and comparison criteria.

Third, the analysis was conducted by a single researcher. Although the coding was guided by the literature review, project scope, interview material, stakeholder input, and prototype iterations, the interpretation of qualitative material still involves researcher judgement. This means that the findings should be understood as design-oriented insights rather than objective operational evidence.

Fourth, the design process was closely connected to the current prototype. This helped ensure that the findings remained relevant to the final design output, but it also means that broader issues outside the prototype scope were not fully developed. Topics such as full manual teleoperation, operator training, staffing models, regulatory certification, multi-operator collaboration, and technical system integration were treated as future work rather than core design contributions.

These limitations define how the results of the thesis should be understood. The thesis provides a conceptually grounded ROC interface proposal and interface-level design knowledge. It does not provide a validated operational system, safety case, or generalisable model of ROC operator performance.

4

Findings and Design-Oriented Analysis

4.1 Analytical Framework and Theme Development

This chapter presents the main findings from expert and stakeholder input, scenario reconstruction, and formative design feedback. The findings are presented as design-oriented insights rather than statistically generalisable user study results. The purpose of the chapter is to explain what operational and interaction problems were identified, and how these problems informed the later design requirements and prototype decisions.

To keep the findings connected to the interface design, the analysis was organised through a design-oriented coding framework. The framework combined literature-informed codes with project-specific codes from the Cstrider context. The literature-informed codes helped identify general human factors issues in remote operation, such as situation reconstruction, workload, trust, decision-making, and automation ambiguity. The project-specific codes captured issues that were especially relevant for autonomous urban ferry supervision, such as departure readiness, ROC veto logic, cross-source confirmation, passenger communication, multi-vessel priority, alert-to-action logic, and shuttle/on-demand mode awareness.

Table 4.1: Design-oriented coding framework

Coding area	Analytical focus	Examples from the material	Design relevance
Operational context	Conditions that shape the task	Docking, departure, boarding, environmental uncertainty, service disruption	Defines the operational situation that the interface must represent
Cognitive integration	How the operator builds understanding	Attention switching, mental reconstruction, cross-source confirmation, lack of direct sensory cues	Supports selected-vessel detail, event timeline, and integrated context views
Decision and responsibility	How judgement becomes action	Hold departure, wait and monitor, contact crew, escalate, approve or reject recommendation	Supports supervisory command logic and response feedback
Interface mediation	How information is represented through the interface	Weak visual priority, unclear alert meaning, limited feedback visibility	Supports information hierarchy, alert cards, and status structure
Trust and automation	How the operator understands automation behaviour	Automation ambiguity, recommendation rationale, uncertainty, command consequence	Supports automation explanation and bounded action support
Project-specific ferry operation	Cstrider-specific operational needs	Departure-readiness evidence, ROC veto, passenger/onboard communication, berth state, shuttle/on-demand mode	Supports service mode labels, communication access, and fleet-level priority

In addition to the coding areas, selected scenarios were analysed through a simple T0–T4 timeline. This helped the analysis focus on the process of operator reasoning rather than only collecting isolated design suggestions.

Table 4.2: Scenario analysis timeline

Timeline phase	Analysis focus	Example in the Cstrider context
T0. Before the situation	What is the current task and system state?	A vessel is approaching a dock in a shuttle corridor
T1. First cue or signal	What first indicates that the situation has changed?	Alert, camera view, radar/AIS target, crew message, or vessel-state change
T2. Diagnosis and confirmation	How does the operator confirm the situation?	Cross-checking map, radar/AIS, camera, crew checklist, and route clearance
T3. Action or decision	What supervisory action is taken?	Hold departure, wait, contact crew, escalate, or accept a system recommendation
T4. Verification and follow-up	How is the result verified after action?	Alert cleared, vessel state changed, crew confirmed, route reopened, or event timeline updated

The coding framework and scenario timeline led to four design-oriented themes. These themes are developed in Sections 4.2–4.5:

1. Remote supervision requires integrated situation reconstruction.
2. Fleet operation requires dynamic priority management.
3. Automation must be explainable before it becomes actionable.
4. Routine operation needs clear responsibility boundaries.

The findings in this chapter are therefore not treated as a theoretical summary disconnected from the design outcome. Each theme is carried forward only when it can be connected to the current thesis scope and prototype. Issues that are important but not implemented, such as full manual teleoperation, motion simulation, complete fail-safe recovery, multi-user collaboration, and operator training infrastructure, are treated as limitations or future work rather than design contributions.

4.2 Theme 1: Remote Supervision Requires Integrated Situation Reconstruction

The first theme is that remote supervision should be understood as integrated situation reconstruction. In the Cstrider ROC context, the operator does not experience the vessel, dock, passengers, traffic, and environment directly. Instead, the situation

must be built from mediated information, such as map data, vessel status, camera views, alerts, route context, communication, and automation feedback.

The analysis showed that the main challenge is not the absence of information, but the separation of information across different sources. A single vessel event may require the operator to check the map, selected-vessel status, camera evidence, route clearance, passenger or boarding state, alert reason, and possible crew confirmation. If these elements are presented as separate fragments, the operator must mentally assemble them before deciding whether the situation is normal, uncertain, or requires intervention.

This finding was especially relevant for docking, departure, and abnormal event scenarios. These situations require the operator to understand both the immediate vessel state and the wider operational context. For example, a departure situation is not only about whether the vessel is technically ready to move. It also depends on onboard readiness, route clearance, berth condition, nearby traffic, passenger state, and whether the ROC has any reason to hold or stop the operation. Therefore, the interface needs to support cross-source confirmation rather than force the operator to search through unrelated panels.

The finding also affected how camera information was interpreted. Camera views can provide useful contextual evidence, but they should not become the only basis for judgement. In a remote ROC setting, camera information needs to be read together with map position, traffic context, vessel state, automation status, and communication. A camera-heavy interface may appear visually rich but still fail to support situation reconstruction if it does not connect what is seen with what the system knows and what action is possible.

For the design, this theme means that the ROC interface should organise information around the operator's reasoning process. The selected-vessel workspace should connect status, route context, alert rationale, evidence, communication, timeline, and response options. The purpose is to help the operator move from distributed signals to a coherent operational picture, and from that picture to a bounded supervisory response.

4.3 Theme 2: Fleet Operation Requires Dynamic Priority Management

The second theme is that fleet operation requires dynamic priority management. In the Cstrider ROC context, the operator does not only monitor one vessel at a time. The operator needs to maintain an overview of several autonomous passenger ferries, different service modes, multiple berths, passenger requests, boarding and departure states, and abnormal events. The design challenge is therefore not simply to show several vessels on the same screen. It is to help the operator understand which vessel needs attention first, why it matters, and whether a supervisory response is required.

The analysis showed that identical vessel cards are not enough for multi-vessel supervision. If all vessels are shown with the same visual weight and information density, the operator must compare them manually. This creates a risk that safety-critical or service-critical situations are missed, especially when several vessels are moving, docking, boarding, or responding to on-demand requests at the same time. A fleet interface therefore needs a priority structure, not only a vessel list.

Priority in this context is not based only on safety severity. It also depends on time pressure, service impact, passenger state, berth availability, task phase, and actionability. For example, a vessel with a minor technical deviation may require monitoring, while another vessel may need immediate attention because it is about to depart, has unclear route clearance, or affects an on-demand assignment. These situations should not appear equal in the interface.

This finding also shows why service mode must remain visible in the fleet overview. Shuttle, on-demand, repositioning, waiting, docking, boarding, and departing states create different expectations for the operator. In shuttle mode, the operator needs to see whether regular crossings and dock readiness are stable. In on-demand mode, the operator also needs to understand request distribution, vessel assignment, passenger waiting time, and fleet availability.

For the design, this theme means that the ROC interface should support priority recognition before detailed diagnosis. The fleet overview should make normal vessels visible but low-interruption, while vessels with abnormal, delayed, uncertain, or action-relevant states should become more prominent. Queue cards should help the operator select what to inspect first by showing alert level, cause, affected vessel, service impact, and whether action is needed. Detailed diagnosis and supervisory command should then move to the selected-vessel workspace rather than being handled directly from the fleet overview.

4.4 Theme 3: Automation Must Be Explainable Before It Becomes Actionable

The third theme is that automation must be explainable before it becomes actionable. In the Cstrider ROC context, automation may support vessel assignment, route monitoring, risk detection, alert prioritisation, and action recommendation. However, the ROC operator still carries supervisory responsibility. This means that a system recommendation cannot be treated as a command that the operator simply accepts. It must be presented in a way that allows the operator to understand the reason, impact, and consequence of the proposed action.

The analysis showed that automation support becomes problematic when it only presents a final recommendation. A message such as “hold departure”, “reroute vessel”, or “accept assignment” is not enough by itself. The operator also needs to know what triggered the recommendation, which vessel or service is affected, how urgent the situation is, and what may happen if the recommendation is accepted or

rejected. Without this context, the operator may either follow the system without enough understanding or spend extra time checking separate information sources before acting.

This finding is especially important for alert handling and supervisory command. In an abnormal event, the alert card should explain the situation, while the decision response area should show the available actions. These two parts should not be separated. If the interface shows an alert without a clear response path, the operator may understand the problem but not know what action is expected. If the interface shows a command without enough explanation, the operator may act without understanding the operational consequence.

Command labels also need to describe operational meaning. Vague labels such as “confirm”, “review”, or “proceed” are weak because they do not explain what will change in the system. Clear labels such as “Hold Departure”, “Contact Crew”, “Escalate”, or “Accept Assignment” are more useful because they describe the action and make its operational consequence easier to understand.

For the design, this theme means that automation explanation, alert rationale, and supervisory command should be treated as one connected flow. The interface should help the operator understand what the system has detected, why it recommends an action, what the action will affect, and how the result can be verified after the command is carried out.

4.5 Theme 4: Routine Operation Needs Clear Responsibility Boundaries

The fourth theme is that routine operation needs clear responsibility boundaries. In the Cstrider ROC context, routine tasks such as departure, docking, boarding, and route monitoring involve several actors and systems. The onboard crew or safety personnel may confirm local readiness, the autonomous vessel system may monitor route and vessel conditions, and the ROC operator may supervise the wider operation and intervene when needed. If these responsibilities are not clearly represented in the interface, the operator may become uncertain about when to monitor, when to communicate, and when to issue a supervisory command.

This finding was especially important for departure logic. A normal departure should not be treated as a repeated approval task for the ROC operator. Based on the project understanding, onboard crew or safety personnel are responsible for local safety checks, while the ROC operator keeps a supervisory veto role. This means that the ROC should not ask the operator to approve every normal departure. Instead, the interface should show the evidence needed to understand whether departure is normal, uncertain, or blocked. The operator should only be pulled into action when route clearance, dock condition, onboard readiness, passenger state, or system alerts indicate that human attention is required.

The same logic applies to docking and boarding. These are routine but safety-relevant phases. If the interface only shows that a vessel is “docking” or “boarding”, the operator may not know whether the process is normal or whether a check is missing. The interface therefore needs to show role-specific evidence: what the vessel system has confirmed, what the onboard crew has confirmed, what remains uncertain, and whether the ROC has any required action.

The analysis also showed that unclear responsibility boundaries can lead to experience-based coping. When system representation is incomplete or ambiguous, operators may rely on personal judgement, cross-checking, and informal communication to compensate. This may work for experienced operators, but it should not become the main operating logic of a future ROC. A supervisory interface should reduce unnecessary interpretation by making responsibilities, evidence, and available actions explicit.

For the design, this theme means that routine operation should be represented through clear responsibility states rather than generic status labels. The interface should distinguish between onboard crew confirmation, automation monitoring, system recommendation, and ROC supervisory authority. The decision response area should only show actions that belong to the ROC operator’s responsibility, such as hold, stop, contact crew, acknowledge, escalate, or accept a bounded recommendation.

5

Design Requirements and Principles

The design requirements were developed by translating the literature review, expert and stakeholder input, scenario analysis, and prototype feedback into concrete interface directions. They define what the ROC interface should support and what it should avoid. The requirements are specific to this thesis context: a multi-screen ROC interface for supervising autonomous passenger ferries in Gothenburg.

The requirements also define the design boundary. The interface should not copy a traditional ship bridge, present the ROC operator as a full-time remote driver, or add manual confirmation steps only to create a sense of control. Instead, it should support the actual supervisory work of the ROC operator: monitoring several vessels, understanding exceptions, communicating with relevant actors, and carrying out bounded supervisory commands.

Table 5.1: Design requirements and principles

ID	Design requirement	Rationale	Interface implication
R1	Support situation construction, not only data display	Remote operators need to build understanding from mediated and distributed information. Isolated map, camera, status, and alert data increase manual reconstruction.	Connect fleet map, vessel status, route context, event cards, evidence views, and timeline into a coherent operational picture.
R2	Use layered attention for multi-vessel supervision	The operator should not inspect every vessel with the same detail at all times. Multi-vessel supervision requires attention hierarchy.	Provide fleet-level overview, selected-vessel detail, persistent vessel monitoring, and event-focused inspection as separate but connected layers.
R3	Make exceptions more visible than routine operation	In normal operation, many tasks are handled by automation. Operator attention is most valuable when the situation moves away from the expected state.	Use alert queue, priority indicators, vessel states, and visual hierarchy to make delayed, degraded, uncertain, or action-relevant events more visible.
R4	Explain cause, impact, and available response	Alerts should not only state that something is wrong. The operator needs to know why it matters, what it affects, and what can be done.	Structure queue cards around event, cause, affected vessel or service, urgency, system recommendation, and possible response.
R5	Keep commands supervisory and context-specific	The prototype should not imply direct steering or full teleoperation. Commands should reflect the ROC operator's bounded responsibility.	Present actions such as hold, stop, acknowledge, contact crew, escalate, request assistance, or accept a limited recommendation only in relevant vessel or event contexts.
R6	Embed communication in operational context	Communication should not be a separate generic tool. It should support the current vessel, event, actor, and decision need.	Link communication actions to selected vessel, alert, crew confirmation, passenger issue, service disruption, or escalation path.
R7	Represent shuttle and on-demand operation as service context	Cstrider's ROC supervises both vessel safety and urban ferry service operation. Shuttle and on-demand modes create different monitoring priorities.	Show service mode, task state, request assignment, berth status, passenger-related state, and service impact in the fleet and selected-vessel views.

These requirements are not independent interface features. They work together. For example, situation construction depends on layered attention and event visibility. Alert explanation depends on cause, impact, and available response. Supervisory command depends on clear responsibility boundaries and contextual communication. The requirements therefore support one overall design direction: an integrated, priority-based, and action-oriented ROC interface.

The requirements were also used as a scope filter. Design elements that could not be represented in the current prototype were not treated as implemented contributions. Full manual teleoperation, joystick control, real route optimisation, back-end dispatch logic, operator training infrastructure, multi-operator collaboration, regulatory certification, and a complete safety case are outside the current design contribution. They are treated as future work or background constraints.

6

Design Development

6.1 Wireframes and Iterations

The design development started with a set of early wireframes and interface drafts for a multi-screen Remote Operation Centre. These early versions explored how vessel status, route information, camera views, alerts, and operator actions could be distributed across several displays. The previous version shown in Figure 6.1 already established an important direction for the project: the ROC interface should not be a single dashboard, but a multi-screen workspace where fleet monitoring, vessel awareness, and event handling are separated across different display areas.

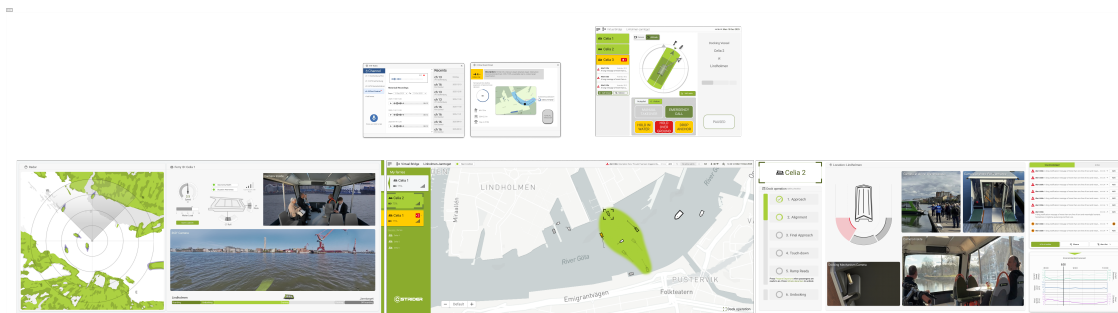


Figure 6.1: Early multi-screen ROC interface draft

In this previous version, several interface ideas were tested at the same time. The design included fleet-level maps, vessel lists, radar-style views, route and docking information, camera feeds, event timelines, alert lists, and action buttons. This was useful because it made the overall operational space visible. It also helped clarify that the operator would need to move between different levels of information: from fleet overview, to selected vessel context, to specific event response. However, the same version also revealed several structural problems.

The first problem was that the screen roles were not yet clear enough. Some screens mixed fleet monitoring, vessel inspection, camera monitoring, and command actions within the same visual space. This made the interface appear comprehensive, but it also risked making the operator's task ambiguous. It was not always clear whether a screen was intended for passive monitoring, selected-vessel inspection, or active supervisory response. For a safety-critical ROC interface, this ambiguity is problematic because the operator needs to understand not only what information is shown,

but also what kind of action is expected from each screen.

The second problem was information fragmentation. The previous version contained many useful elements, but the relationship between them was not always explicit. For example, alerts, vessel status, camera evidence, route context, and response actions were sometimes shown as separate interface areas without a strong visual or logical connection. This could force the operator to reconstruct the situation manually by looking across different panels. Since remote operation already depends on mediated information, the interface should reduce this reconstruction burden rather than increase it.

The third problem was that monitoring and command were not sufficiently separated. Some early screens included action buttons close to monitoring information, while other screens also implied command functions. This created a risk of distributing command responsibility across too many displays. During the iteration process, this became a key design decision: monitoring information can be distributed across multiple screens, but supervisory command should remain concentrated in a dedicated decision workspace. This separation supports a clearer boundary between observing a vessel, interpreting an event, and issuing an accountable command.

The fourth problem concerned the map and spatial information. The previous version used map-based views to show vessel positions and routes, but the map still behaved too much like a general geographic display. For the Cstrider context, the most important spatial object is not the full city map, but the operational river corridor, including vessels, berths, route segments, service modes, alert areas, and local traffic constraints. This led to a later shift from a generic map view towards a more operational corridor-based representation.

The fifth problem was the balance between camera information and decision support. The previous version included multiple camera views, which were important for remote visual confirmation. However, the iteration showed that camera feeds should not dominate the interface or become the main organising structure. In remote supervisory operation, cameras are better understood as evidence that supports interpretation and trust calibration. They need to be connected to the selected event, vessel status, route context, and system recommendation.

These issues led to several changes in the later design. The interface was reorganised around clearer screen responsibilities. Fleet-level awareness was separated from focused command handling. Persistent vessel monitoring was treated as read-only background awareness rather than as an additional command surface. The command workspace was developed as a focused area where event explanation, evidence, automation recommendation, communication, and response actions could be reviewed together. The map was also reframed as an operational corridor view rather than a general city map.

The iteration process therefore did not simply refine the visual style of the interface. It changed the underlying organisation of the ROC system. The previous version

helped expose the risk of treating the ROC as a collection of useful panels. The later design instead treats the ROC as a structured supervisory workspace, where each screen has a defined role and where the operator can move from monitoring, to interpretation, to bounded supervisory response. This design logic forms the basis for the final information architecture presented in the next section.

In addition to the digital previous version, the early hand-drawn wireframes were also important in the design process. These sketches were not used to define detailed visual components, but to explore the basic spatial logic of the ROC workstation. They helped test how monitoring, vessel selection, camera views, event information, and command-related areas could be arranged before the interface was developed into higher-fidelity screens. The sketches also made it possible to compare different layout ideas quickly and identify the need for clearer separation between fleet overview, selected-vessel information, and supervisory response.

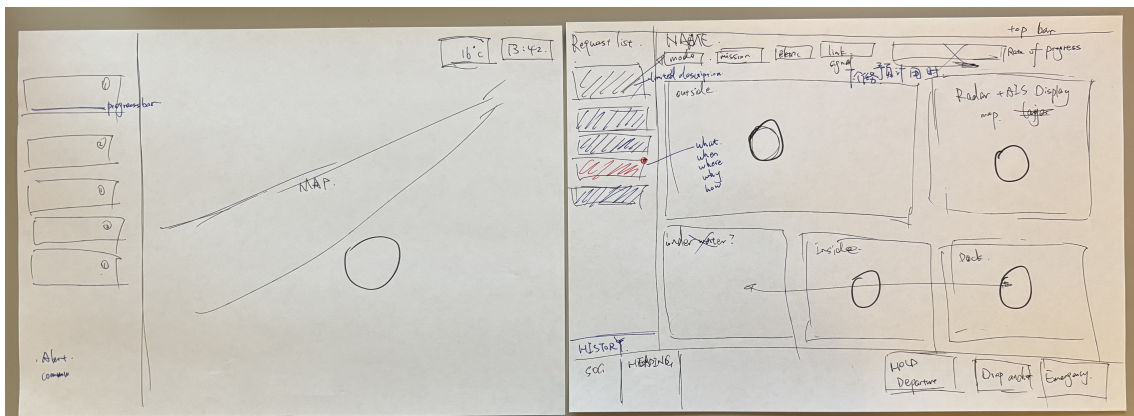


Figure 6.2: Initial sketch wireframe

The sketch wireframe shows that the design process first focused on the overall relationship between screens and information areas, rather than on detailed visual styling. This supports the transition from an exploratory layout to the later digital prototype. Together with the previous version shown above, the sketch demonstrates how the final information architecture was developed through iteration: from rough spatial organisation, to a multi-screen interface draft, and finally to a clearer supervisory workspace with defined screen roles.

6.2 Information Architecture

The final prototype is organised as a multi-display ROC interface for remote supervisory operation. Its information architecture is based on three connected work layers: monitoring, communication, and supervisory command. These layers are not treated as equal interface functions. Monitoring forms the base layer, communication becomes relevant when clarification or coordination is needed, and supervisory command appears only when a selected vessel or event requires operator action.

This hierarchy reflects the final scope of the thesis. The ROC operator is not de-

signed as a continuous remote driver. The interface does not support manual steering, joystick control, throttle control, or full teleoperation. Instead, it supports fleet-level awareness, selected-vessel understanding, contextual communication, and bounded supervisory response.

The information architecture can be summarised as a movement from overview, to focus, to response. The operator first reads the fleet situation, then selects a vessel, route area, or event, and then moves into a focused workspace where evidence, system rationale, communication options, and available responses are shown together. This structure prevents command actions from becoming generic buttons detached from operational context.

The architecture is divided into three information layers.

The left main screen works as the fleet-level entry point. It is based on a 34-inch ultrawide layout and provides a corridor-based operational overview of the Gothenburg ferry area. The main purpose of this screen is not to provide detailed control, but to help the operator understand where the vessels are, which vessel needs attention, what service mode is active, and whether the fleet is operating as expected. The map, vessel list, status indicators, and selected-vessel summary are therefore organised around fleet orientation and priority recognition.

The right main screen functions as a case-focused command workspace for system-initiated exceptions and supervisory responses. It is not used as a general fleet monitoring dashboard. Instead, it organises information around the currently selected command case from the command queue. The screen brings together the information needed for deliberate judgement, including the event explanation, relevant vessel status, tactical context, supporting camera and sensor evidence, automation recommendation, communication options, and response actions. Therefore, the right screen supports exception handling and bounded supervisory command, rather than continuous monitoring or direct vessel control.

The five 24-inch vessel screens provide persistent per-vessel monitoring. Each screen is assigned to one vessel and remains visible while the operator works on the two main screens. Their role is deliberately limited to passive awareness. They may show vessel identity, route progress, operating mode, speed, battery, passenger-related state, warning level, camera view, and recent events. However, they do not include command buttons, approval controls, emergency actions, or hidden command menus. If a vessel requires action, the operator should handle that action on the right main screen, where the event can be interpreted with its context, evidence, and response options.

This separation is important because the ROC workstation should not become either one overloaded dashboard or five separate vessel-control stations. The left screen keeps the operator oriented at fleet level. The right screen supports focused understanding, communication, and supervisory response. The smaller vessel screens maintain persistent awareness without adding extra control surfaces. In this way, the information architecture supports the thesis aim: a remote supervisory interface for

Table 6.1: Information architecture of the final ROC prototype

Layer	Main operator question	Main screen location	Interface role
Fleet-level monitoring	What is happening across the ferry operation?	Left main screen	Supports spatial overview, vessel distribution, route context, service mode, vessel priority, and abnormal-state recognition.
Selected-vessel / selected-event focus	What is happening with this vessel or event?	Right main screen	Supports diagnosis through vessel status, event reason, tactical map, sensor state, camera evidence, automation recommendation, and impact information.
Contextual communication	Who needs to be contacted or confirmed?	Mainly right main screen, linked to selected vessel or event	Supports contact with the vessel, crew, passenger, local support, or external actor when the event requires clarification or coordination.
Supervisory command	What high-level response is available?	Right main screen response area	Supports bounded actions such as hold, monitor, request confirmation, continue with restrictions, contact, escalate, E-stop, or other protected emergency responses.
Persistent vessel monitoring	Is each individual vessel still in an acceptable state?	Five 24-inch vessel monitoring screens	Supports continuous read-only awareness of each vessel's movement, route progress, system health, camera view, passenger state, and recent events.

monitoring, communication, and bounded command, rather than a remote-driving cockpit.

This architecture is also consistent with recent ROC design thinking, where the ROC is treated as an integrated workstation and workplace architecture rather than a single-screen interface (Gernez et al., 2026).

6.3 Final Prototype

The final prototype presents a multi-screen ROC interface for supervising autonomous passenger ferries in an urban waterway context. It brings together the information architecture, screen strategy, interaction flow, and design iterations described in the previous sections. The prototype is not presented as a certified operational system or a complete technical implementation. It is a high-fidelity design proposal that explores how a ROC interface can support monitoring, communication, and bounded supervisory command.

The prototype consists of two main interactive screens and five smaller vessel monitoring screens. The left main screen supports fleet-level monitoring and prioritisation. The right main screen supports selected-event diagnosis, communication, and supervisory response. The five 24-inch screens support persistent read-only awareness of individual vessels. Together, these screens form a workstation that allows the operator to move from fleet-level overview to focused event handling without losing awareness of the wider fleet.

Table 6.2: Final prototype structure

Prototype part	Main role	Design contribution
Left main screen	Fleet monitoring and prioritisation	Provides a corridor-based overview of the Gothenburg ferry operation, vessel distribution, service mode, status, and attention priority.
Right main screen	Command workspace	Connects command queue, selected event, tactical context, system recommendation, evidence, communication, and response actions.
Five 24-inch vessel screens	Persistent vessel monitoring	Keeps each vessel visible through read-only state, route progress, camera, system health, passenger state, and recent events.

The left screen represents the fleet-level layer of the prototype. Its role is to help the operator understand the operational field before entering detailed event handling. It combines a Gothenburg waterway map, vessel positions, route or corridor context,

wharf labels, traffic or operational state, vessel cards, status filtering, and selected-vessel summary. The design direction is not a generic city map, but an operational waterway corridor view. This supports spatial awareness and priority recognition across the fleet.

The right screen represents the focused decision layer. It is organised around a command queue, selected command workspace, tactical or route view, system recommendation, vessel status, sensor and camera evidence, communication access, and response bar. Its purpose is to help the operator understand what happened, why it matters, what the system recommends, what evidence supports the recommendation, and what bounded response is available. The right screen is therefore a command workspace, not a general monitoring dashboard or camera wall.

The 24-inch vessel screens represent the persistent monitoring layer. Each screen is assigned to one vessel and provides continuous read-only awareness. They may show vessel state, route progress, local map, camera feeds, passenger status, system health, active alert, and recent events. They do not include command buttons, approval controls, emergency stop controls, route recalculation, sensor check, or hidden command menus. If a vessel requires action, the operator should handle the case in the right command workspace.

The final prototype also clarifies the command boundary of the system. Routine autonomous actions, such as ordinary route following, routine docking, routine departure, and normal obstacle avoidance, are not treated as repeated approval tasks for the ROC operator. They can be logged and monitored, but they should not become command queue items by default. The operator becomes active when autonomy reaches uncertainty, service impact, operational boundary, passenger safety, or safety-critical escalation. This keeps the interface aligned with remote supervisory operation rather than remote driving.

The visual design uses an OpenBridge-inspired maritime HMI language to make the prototype credible within the remote maritime operation context. The interface relies on structured panels, status badges, semantic alert colours, compact cards, map controls, sensor indicators, and consistent spacing. The purpose of the visual design is not decoration. It supports scanability, status recognition, priority judgement, and trust in a safety-critical supervision context.

The final design can be understood through four design outcomes. First, it separates overview, focus, and response across screens. Second, it makes abnormal and action-relevant states more visible than routine operation. Third, it connects system recommendations with evidence, impact, communication, and response. Fourth, it prevents the 24-inch screens from becoming fragmented command surfaces. These outcomes directly respond to the thesis aim of designing a ROC interface for multi-vessel supervision under different operational scenarios.

The prototype also has clear limitations. It does not implement real-time vessel control, backend dispatch optimisation, certified route planning, live sensor integration, regulatory compliance logic, or full manual teleoperation. It also does not validate that the interface improves workload, situation awareness, or safety performance. These aspects require further technical development and evaluation. The contribution of the prototype is therefore interface-level design knowledge: it shows how information, attention, communication, and bounded command can be organised in

a future ROC workstation for autonomous urban ferry operation.

6.3.1 Screen Strategy and System Layout

The screen strategy of the final prototype is based on a division of labour between displays. The ROC operator needs to maintain fleet-level awareness, inspect selected vessels or events, communicate with relevant actors, and issue bounded supervisory responses when necessary. These tasks require different levels of detail and different levels of interaction. For this reason, the prototype does not place all functions on one large dashboard. Instead, it uses two main interactive screens and five read-only vessel monitoring screens.

The final workstation is organised into three display groups.

Table 6.3: Screen strategy of the final ROC prototype

Display group	Screen format	Main role	Interaction type
Left main screen	34-inch ultrawide, 21:9, 3440 × 1440	Fleet monitoring, map-based overview, route context, service mode, vessel priority, and vessel or area selection	Active overview and selection
Right main screen	27-inch, 16:9, 2560 × 1440	Selected-vessel or selected-event inspection, command queue, evidence, recommendation, communication, and supervisory response	Focused inspection and response
Five vessel monitoring screens	24-inch, 16:9, 1920 × 1080	Persistent per-vessel status, route progress, camera view, system health, passenger state, and recent events	Read-only passive monitoring

The left main screen is the starting point of the operator’s visual workflow. It provides the fleet-level overview and shows the operational field across the Gothenburg waterway. Its role is to help the operator understand where the vessels are, which vessel needs attention, what service mode is active, and whether the ferry operation is developing normally. The left screen therefore works as an orientation and prioritisation space rather than a detailed control panel. In the final design, it uses a map-left and vessel-panel-right layout, with the map taking the dominant area and the vessel panel supporting selection, comparison, and quick inspection. This follows the final left-screen design decision that the operator’s primary spatial awareness should come from the map, while the vessel panel supports vessel comparison and selection.

The right main screen is the focused workspace. It becomes relevant after the

operator selects a vessel, area, alert, or command-related event. This screen brings together the information needed for deliberate judgement: command queue, selected command header, tactical or route decision map, system recommendation rationale, risk and impact information, vessel status, sensor state, camera evidence, nearby traffic, command alternatives, and the bottom response bar. The right screen is therefore not a general monitoring dashboard. It is the place where the operator reviews what happened, why it matters, what the system recommends, what evidence supports the case, and what response is available.

The five 24-inch vessel monitoring screens support persistent awareness. Each screen is assigned to one vessel and shows its current state without requiring the operator to open and close detailed views on the main screens. These screens may show vessel identity, route progress, local map, camera feeds, passenger state, system health, active alerts, and recent events. However, they are deliberately read-only. They do not include command buttons, approval controls, emergency stop controls, route recalculation, sensor check, or hidden command menus. If a vessel requires action, the operator should move to the right main screen, where the case can be interpreted with its reason, impact, evidence, and available response.

This layout creates a clear attention flow. The operator first scans the left screen to understand the fleet situation. When a vessel or event requires attention, the operator moves to the right screen to inspect the selected case. Communication and supervisory commands are then shown in the selected context. During this process, the five smaller screens keep each vessel visible in the background. This allows the operator to focus on one event without losing peripheral awareness of the rest of the fleet.

The screen strategy also avoids two weaker alternatives. The first would be a single overloaded dashboard where map, vessel detail, alert queue, communication, cameras, and command actions all compete for attention. The second would be five repeated vessel-control interfaces, where each vessel has its own command panel. Both alternatives would make prioritisation harder and increase the risk of fragmented interaction. In the final design, the left screen keeps the operator oriented, the right screen supports deliberate inspection and response, and the 24-inch screens maintain background awareness without becoming extra control surfaces. This screen strategy supports the thesis scope: monitoring, communication, and supervisory command.

This division is also consistent with remote ship operation research. Earlier Shore Control Centre studies used multi-display workstations to separate alarm dashboards, chart information, radar or conning information, fleet overview, and camera support. The lesson is not that the Cstrider prototype should copy those workstations directly. The relevant lesson is that remote supervision requires both overview and detail, and these layers need to be organised spatially so that the operator can move from monitoring to diagnosis and response without losing situational awareness.

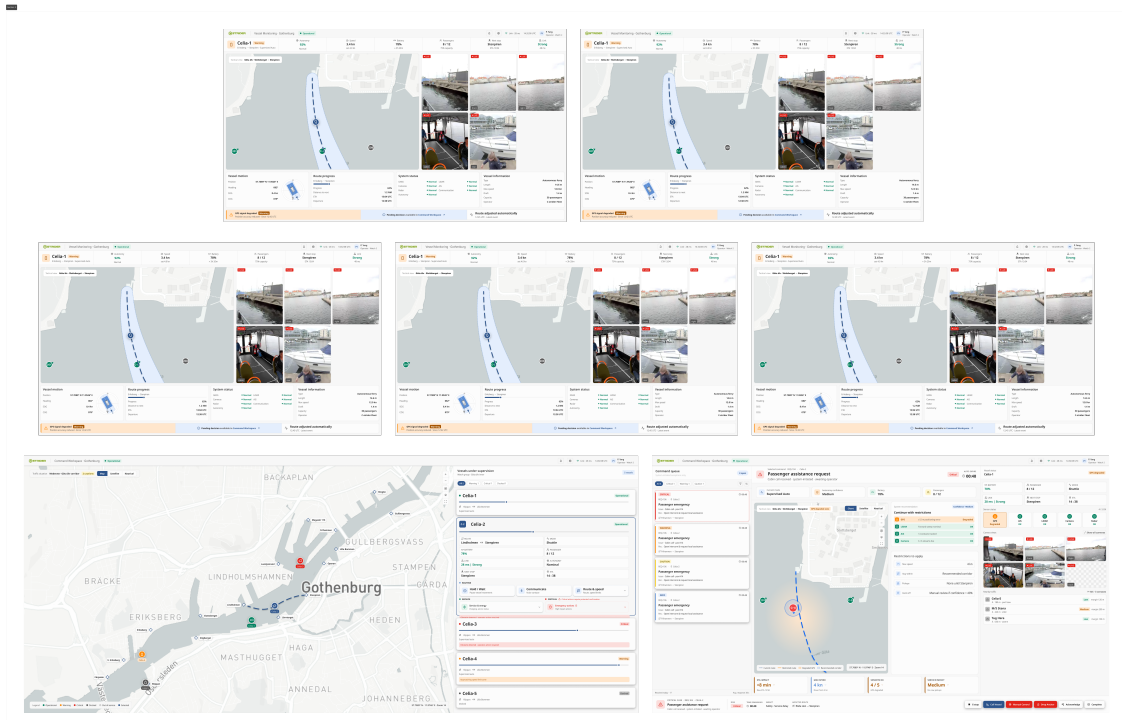


Figure 6.3: Multi-display screen strategy of the ROC prototype

6.3.2 Left Screen: Fleet Monitoring Interface

The left screen is the fleet monitoring interface of the final ROC prototype. It is designed for a 34-inch ultrawide display and works as the operator's primary entry point into the ferry operation. Its purpose is to support fleet-level situation awareness, spatial orientation, vessel selection, service mode awareness, and priority recognition before the operator enters detailed event handling on the right screen.

The left screen is not designed as a full command workspace. It does not aim to support detailed diagnosis, evidence comparison, or final response execution. Instead, it helps the operator answer the first operational questions: where the vessels are, which vessel needs attention, what route or service context is relevant, what abnormal state is visible, and whether the fleet is operating as expected.

The final layout places the operational map on the left and the vessel supervision panel on the right. This arrangement gives priority to spatial awareness. In the Cstrider context, vessel position, route direction, wharf relationship, and corridor state are central to fleet supervision. The vessel panel then supports comparison, filtering, and selected-vessel inspection without taking over the main spatial view.

The main map is not intended to function as a generic city map. Because the design context is the narrow Göta älv waterway, a normal city-map representation can waste screen space on land, roads, and urban labels that are not central to ROC work. The map should therefore be understood as an operational corridor map. It should emphasise the river corridor, vessel locations, direction of movement, wharfs,

service routes, traffic context, caution zones, and alert-related spatial areas. Land remains useful as context, but it should not dominate the operator's attention.

The final prototype shows this direction by placing vessel markers, wharf labels, route context, status legend, and traffic situation information on the Gothenburg map. The visible markers and labels allow the operator to relate vessel status to real service locations such as Lindholmen, Stenpiren, Järntorget, Operan, Frihamnen, and Lilla Bommen. This supports the local urban ferry context rather than treating the vessels as isolated moving objects.

The right-side vessel supervision panel supports fleet comparison and priority recognition. It shows the watch group, the number of supervised vessels, status filters, vessel cards, and one expanded selected-vessel card. The selected-vessel card provides compact information such as route, mode, battery, passengers, communication link, next stop, autonomy state, and ETA. This allows the operator to inspect one vessel without leaving the fleet overview.

The vessel list also supports priority recognition. Vessels should not appear as equal cards when their operational states are different. A vessel with a warning, critical issue, docking state, or service delay should be easier to detect than a vessel operating normally. The purpose is not to overload the list with detailed data, but to show which vessel may require attention first and why.

Service mode is also part of the left-screen logic. Shuttle and on-demand operations create different expectations for the operator. In shuttle mode, the operator expects repeated movement along a known corridor and stable arrival or departure cycles. In on-demand mode, the operator needs to understand changing request locations, assigned vessel movement, and possible service imbalance. The left screen should therefore make service mode visible at fleet level, but it should not suggest that the operator manually dispatches every vessel. Dispatch and route optimisation remain outside the implemented prototype scope.

The selected-vessel action area on the left screen needs a narrow interpretation. It can provide stable access to common vessel-related actions, such as communication, route or speed information, service and energy information, hold or wait state, and protected emergency access. However, it should not become the place where the operator completes high-impact command decisions. Actions that require evidence review, consequence understanding, or system recommendation should be handled through the right command workspace.

This distinction is important. If the left screen allowed full command execution, the interface would blur the boundary between fleet monitoring and event response. The final design therefore treats the left screen as a selection and orientation layer. It can open or prepare an action path, but detailed response should remain connected to the selected vessel or event on the right screen.

The bottom bar supports operational continuity. It shows operator identity, watch status, link state, time, and basic system state. This information is useful because

the left screen is always visible during supervision. However, the bottom area should remain secondary. It should not compete with the map and vessel panel unless there is a high-priority system decision or warning that requires attention.

Overall, the left screen supports the first stage of ROC work: maintaining fleet awareness and identifying what deserves attention. It gives the operator a structured overview of the operational corridor, vessels, service mode, abnormal states, and selected-vessel summary. Detailed diagnosis, communication, and supervisory response are then continued on the right screen. In this way, the left screen keeps the operator oriented without turning the ROC workstation into a manual remote-driving interface.

Table 6.4: Left-screen elements and functions

Left-screen element	Function
Operational corridor map	Shows vessel positions, routes, wharfs, traffic context, and abnormal areas.
Vessel supervision panel	Supports fleet comparison, filtering, and selected-vessel inspection.
Selected-vessel card	Shows compact route, mode, battery, passengers, link, autonomy, next stop, and ETA.
Vessel action area	Provides stable or protected access to vessel-related actions.
Bottom status bar	Shows operator, watch, link, time, and system continuity information.

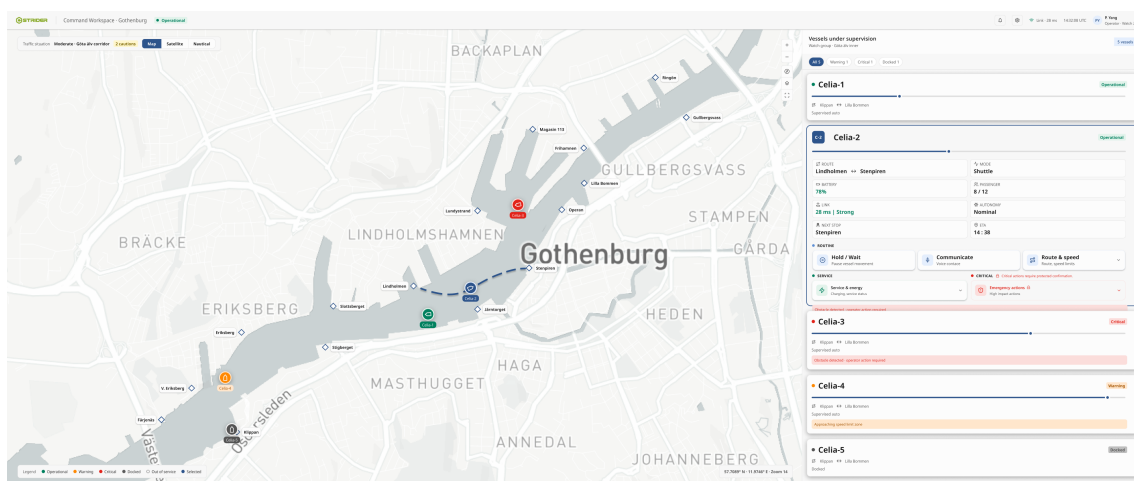


Figure 6.4: Left-screen elements and functions

6.3.3 Right Screen: Command Workspace

The right screen is the command workspace of the final ROC prototype. It supports focused inspection and response for one selected event of selected vessel. While the left screen keeps the operator oriented at fleet level, the right screen helps the operator understand what is happening in a specific case, why it matters, what the system recommends, what evidence supports the recommendation, and what supervisory response is available.

The right screen is not a general monitoring dashboard. It is also not a manual driving console. It does not support joystick control, continuous steering, throttle control, or manual docking. Its function is to support bounded supervisory command in selected operational situations. This keeps the operator role aligned with remote supervision rather than continuous remote vessel handling.

The screen is organised into four main areas: a command queue, a selected command workspace, an evidence and vessel status panel, and a bottom response bar. The command queue helps the operator identify which system-initiated cases require attention. The selected command workspace explains the active case and its operational impact. The evidence panel provides supporting vessel, sensor, traffic, and camera information. The response bar presents the available supervisory actions for the selected case.

The command queue is limited to cases that need operator awareness or possible intervention. It should not contain every routine automation update. Routine departure, routine docking, route following, and ordinary obstacle avoidance should normally be logged as events rather than treated as command requests. The queue therefore focuses on abnormal, uncertain, time-sensitive, service-relevant, or safety-relevant situations.

The queue uses four alert levels: Info, Caution, Warning, and Critical. These levels describe more than severity. They also indicate urgency, expected operator role, response time, service impact, and actionability. Info means that the operator should know the event. Caution means that the operator should watch the event. Warning means that the operator should decide within a time window. Critical means that the operator should act immediately. This structure prevents low-priority updates from competing visually with time-critical cases.

The selected command workspace is the centre of the screen. It should make the selected event understandable without forcing the operator to search through unrelated panels. For example, if the selected event concerns degraded positioning, the workspace should connect the affected vessel, degraded area, system recommendation, confidence level, route restriction, nearby traffic, ETA impact, and available response. If the selected event concerns a passenger assistance request, the workspace should prioritise cabin evidence, communication access, passenger state, local assistance, and possible escalation.

The system recommendation panel is a key part of the right screen. It should not only state the recommended action. It should also explain the reason for the recom-

Table 6.5: Right-screen components and functions

Component	Function
Command queue	Organises system-initiated exceptions by severity, urgency, affected vessel, reason, recommendation, and route or service context.
Selected command header	Identifies the selected event, vessel, alert level, time window, and current automation state.
Tactical / route view	Shows the selected vessel, current route, restricted route, degraded area, traffic context, and recommended corridor when relevant.
System recommendation panel	Explains what the system recommends, why the recommendation is made, confidence level, known limitations, and expected restrictions.
Evidence panel	Shows relevant vessel status, sensor state, camera views, passenger status, nearby traffic, or docking context.
Communication access	Allows contact with the relevant vessel, crew, passenger support, or external actor in the context of the selected event.
Response bar	Provides context-specific supervisory actions such as monitor, hold, continue with restrictions, request confirmation, call vessel, acknowledge, complete, escalate, or protected emergency action.

mendation, the confidence level, known constraints, and the expected operational consequence. This is necessary because the operator remains responsible for supervisory judgement. The operator should not be asked to accept a recommendation without understanding what will change after the response.

The evidence area provides support for this judgement. Camera views are retained because they give direct visual confirmation and help compensate for the lack of physical presence in remote operation. However, the camera area should not dominate the whole screen. Cameras are treated as a contextual evidence layer. They support the interpretation of the selected event, but they do not replace the tactical map, recommendation rationale, sensor state, or response logic.

The final right-screen design may keep multiple camera thumbnails visible, including views such as front, rear, port, starboard, cabin, and dock. Their relative importance depends on the event. A passenger case should prioritise the cabin view. A docking case should prioritise dock and side views. A route or obstacle case should prioritise front view, tactical map, and nearby traffic. This allows visual evidence to remain available without turning the right screen into a surveillance wall.

Communication is also embedded in the selected event context. The operator should not open a generic communication tool and then manually reconstruct which vessel

or issue the call belongs to. Instead, communication actions should be connected to the selected vessel, missing confirmation, passenger issue, degraded state, or escalation path. This supports clarification and coordination without separating communication from decision-making.

The bottom response bar is the main supervisory command area. Its actions should change according to the selected event. A caution-level advisory may offer actions such as monitor, request status update, notify passengers, acknowledge, or complete. A warning-level degraded-state case may offer continue with restrictions, hold, request confirmation, call vessel, or escalate. A critical case may provide immediate protected actions such as E-stop, call vessel, request local assistance, or other emergency escalation. These actions should be specific enough to show operational meaning. Generic labels such as “confirm” or “review” are weaker because they do not explain what will happen after the operator acts.

The response bar should not be interpreted as a manual control panel. Manual steering, throttle control, joystick docking, detailed path drawing, sensor calibration, and autonomy-parameter editing are outside the current prototype scope. If an entry such as manual control appears in the mock-up, it should be treated only as a protected emergency escalation or future-work placeholder, not as a normal implemented function. Similarly, protected emergency actions such as E-stop and drop physical anchor should be visually separated from routine supervisory actions.

Overall, the right screen supports the second and third stages of ROC work: focused situation interpretation and bounded response. It links alert priority, event explanation, automation rationale, evidence, communication, and action into one selected-event workflow. This prevents the operator from acting on isolated alerts or generic controls, and it keeps command actions tied to the operational context in which they are meaningful.

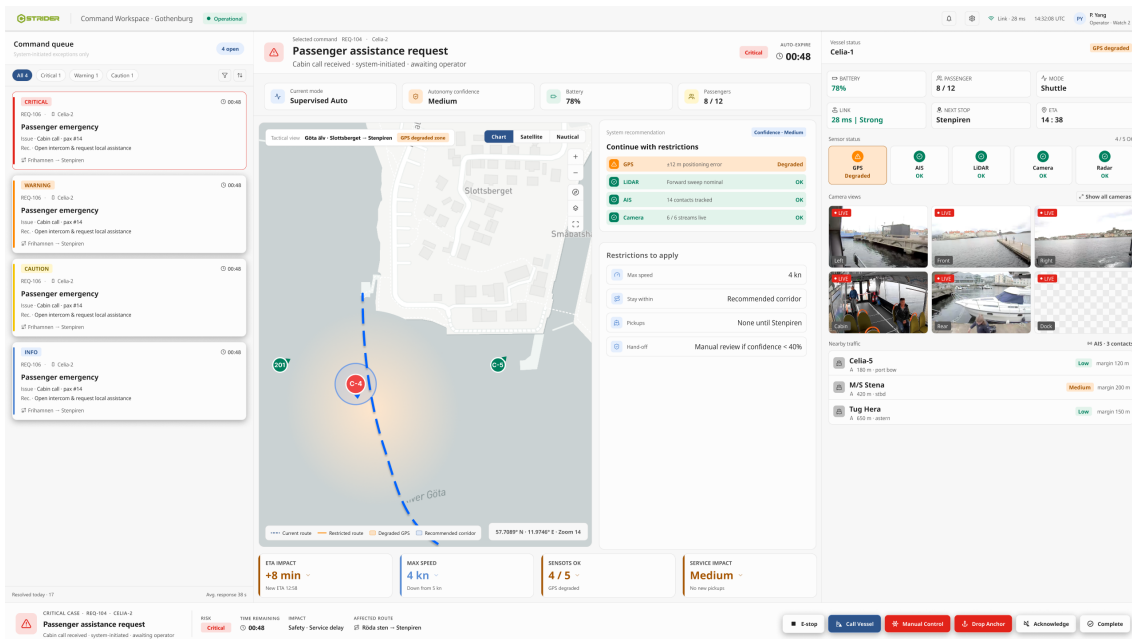


Figure 6.5: Right main screen: command workspace

6.3.4 Interaction Flow

The interaction flow of the final prototype describes how the ROC operator moves between fleet monitoring, vessel or event inspection, communication, and supervisory command. It does not describe full vessel control. The purpose is not to show how the operator manually drives a vessel, but to show how the interface supports remote supervisory work when routine autonomy, service operation, or onboard confirmation requires human attention.

The interaction model contains two connected levels. The first level is the wider system interaction between passengers, the booking system, fleet dispatch, autonomous vessels, onboard safety crew, the ROC interface, the ROC operator, and the external environment. The second level is the screen-based interaction inside the ROC workstation, where the operator moves from the left fleet screen, to the right command workspace, while maintaining passive awareness through the 24-inch vessel monitoring screens.

The wider system flow is important because the ROC interface is not an isolated dashboard. It receives and presents information from several sources. Passenger requests may enter through the booking system. Fleet assignments and service updates may come from the fleet dispatch system. Vessel state, route progress, sensor status, camera feeds, and automation events come from the autonomous vessels. Local readiness or passenger-related confirmation may come from onboard safety crew. External context, such as traffic, dock state, weather, and restricted areas, also shapes the operational situation. The ROC interface integrates these sources and presents them to the operator as fleet overview, selected-event evidence, communication options, and response actions.

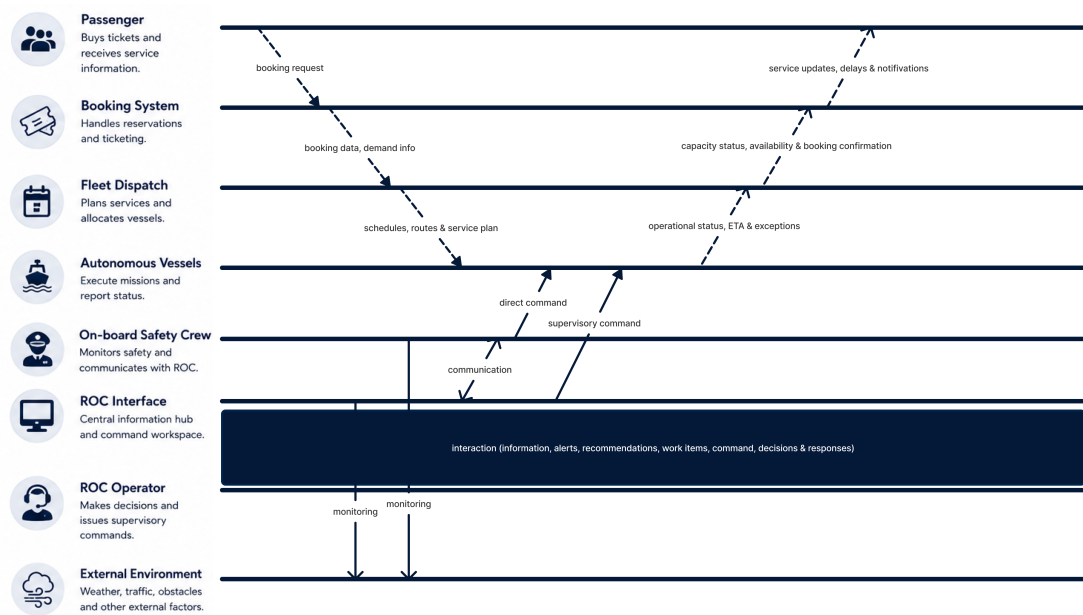


Figure 6.6: Actor and interaction flow around the ROC interface

This actor-level flow explains why the ROC interface must support both service awareness and operational safety awareness. In normal operation, the system should handle passenger request processing, vessel assignment, route following, docking, and routine obstacle avoidance without continuous operator approval. These events can be shown as status updates or event logs. They should not become command queue items by default.

The operator becomes more active when the system detects an abnormal, uncertain, service-critical, passenger-related, or safety-critical situation. In this case, the interaction flow moves from monitoring to focused review. The interface should show what happened, why it matters, what evidence is available, what the system recommends, who may need to be contacted, and what supervisory response is possible. This matches the final design direction that the right screen should connect alerts, evidence, automation recommendation, communication, and response options rather than showing them as separated panels.

The screen-based flow starts from the left screen. The operator first scans the fleet map, vessel list, service mode, and abnormal state indicators. The left screen supports the first question: whether the fleet is operating normally and which vessel or area requires attention. If the situation is normal, the operator does not need to intervene. Routine updates remain visible through the left screen, event logs, and the 24-inch vessel monitoring screens.

When a vessel, area, or event needs attention, the operator selects it from the left screen or from a command queue item. The interaction then moves to the right screen. The right screen presents the selected case with event reason, vessel status, tactical map, route or service impact, sensor state, camera evidence, automation recommendation, and available response. The command action is therefore not

detached from context. It appears only after the operator has selected a vessel or event and reviewed the relevant information. This is consistent with the intended attention flow from fleet scan, to selected-event inspection, to communication or supervisory command.

Communication is treated as part of the event flow, not as a separate generic tool. If the issue concerns missing crew confirmation, the communication action should connect to the onboard safety crew. If the issue concerns passenger assistance, the action should connect to the vessel, cabin, passenger support, or local assistance. If the issue concerns an operational escalation, the communication path should connect to the relevant support actor. This avoids forcing the operator to open a separate communication system and manually reconstruct which event the call belongs to.

Supervisory command appears at the end of the selected-event flow. The response options should depend on the type and severity of the event. For example, an information-level event may only need to be viewed or acknowledged. A caution-level event may require monitoring or a status update. A warning-level event may require the operator to continue with restrictions, hold, request confirmation, or escalate. A critical event may require immediate protected actions such as E-stop, calling the vessel, requesting local assistance, or another emergency escalation. The four alert levels therefore guide the operator role: Info means know, Caution means watch, Warning means decide, and Critical means act.

After a response is selected, the system should show execution state and follow-up information. The operator should be able to see whether the command is pending, accepted, executed, rejected, or requires further confirmation. The event timeline should then record the system detection, recommendation, communication, operator response, and resulting state. This supports later verification and handover because the operator can reconstruct not only the current state, but also how the situation developed.

The 24-inch screens support the flow by keeping each vessel visible in the background. They do not interrupt the main workflow with command buttons. Their role is to show vessel identity, operating mode, route progress, camera or sensor views, system health, passenger state, active alert, and recent event summary. If a vessel requires action, the screen may indicate that a pending decision is available in the command workspace, but the operator should handle the action on the right screen. This keeps command decisions centralised and avoids turning the five vessel screens into five separate control stations.

The final interaction flow can therefore be summarised as follows:

1. Monitor fleet state on the left screen.
2. Notice a vessel, area, service issue, or system-initiated exception.
3. Select the vessel or event.

4. Review the selected case on the right screen.
5. Check evidence, recommendation, impact, and time window.
6. Communicate with the relevant actor if clarification is needed.
7. Choose a bounded supervisory response.
8. Verify execution and follow-up state through the event timeline.
9. Continue fleet monitoring while 24-inch screens maintain vessel awareness.

This flow supports the central design argument of the thesis. The ROC operator is not placed in continuous direct control. Instead, the interface supports a transition from distributed operational signals to a selected event, from the selected event to contextual understanding, and from contextual understanding to accountable supervisory response.

7

Evaluation and Design Review

7.1 Evaluation Aim

The evaluation was planned as a design review of the final ROC prototype. Its purpose was to examine whether the interface concept supports the operator's intended work in the proposed autonomous urban ferry context. The evaluation was not designed to certify the interface for real operation, test vessel autonomy, validate route optimisation, assess collision avoidance algorithms, or evaluate full manual teleoperation. Instead, it focused on the interface-level question of whether the prototype supports monitoring, communication, and bounded supervisory command.

This framing is important because the final prototype is a high-fidelity interface concept, not a fully functional simulator or operational ROC system. The evaluation therefore focuses on how participants understand the interface, interpret the operational situation, identify priority, understand automation recommendations, and judge possible supervisory responses. It does not claim to prove operational safety, workload reduction, or performance improvement.

The main evaluation question was:

Does the final ROC prototype help the operator understand the fleet situation, identify what needs attention, understand automation behaviour, and choose an appropriate supervisory response?

To answer this design-level question, the evaluation focused on six aspects of the prototype.

Table 7.1: Evaluation aims

Evaluation focus	Aim of evaluation	Related prototype area
Situation awareness	To examine whether participants can understand what is happening across the fleet and with a selected vessel.	Left fleet screen, right command workspace, 24-inch monitoring screens
Priority recognition	To examine whether participants can identify which vessel, event, or service issue needs attention first.	Vessel list, alert queue, status badges, event priority
Automation recommendation understanding	To examine whether participants understand what the system recommends, why it recommends it, and what consequence the response may have.	System recommendation panel, rationale, impact information, response bar
Mode awareness	To examine whether participants can distinguish shuttle mode and on-demand mode and understand how these modes change the supervision task.	Mode labels, route context, task state, demand or service indicators
Supervisory command confidence	To examine whether participants understand which supervisory actions are available and whether those actions feel appropriate for the selected event.	Response bar, communication actions, protected actions, event context
Perceived workload	To examine whether the multi-screen structure feels manageable or visually and cognitively overloaded.	Screen layout, information hierarchy, map/camera/sensor density

Situation awareness is central because the ROC operator is separated from direct physical cues. In this prototype, situation awareness refers to the operator's ability to understand the fleet state, the selected vessel state, and the likely development of an event. The evaluation should therefore examine whether the left screen, right screen, and 24-inch vessel screens together support a coherent operational picture rather than fragmented information.

Priority recognition is also critical because the operator supervises multiple vessels and should not inspect all vessels with equal depth at all times. The evaluation should examine whether the interface helps participants identify which vessel or event requires attention, and whether alert levels, vessel status, service impact, and queue structure make this priority understandable.

Automation recommendation understanding concerns whether participants can interpret the system's suggested response. The prototype should not ask the operator to accept a recommendation without context. The evaluation should therefore examine whether participants understand the reason, confidence, restriction, impact,

and expected result of a recommendation before choosing a response.

Mode awareness concerns the distinction between shuttle and on-demand operation. These modes create different expectations for route regularity, request handling, vessel assignment, passenger waiting time, and service impact. The evaluation should examine whether participants can understand the active mode and use it correctly when interpreting vessel behaviour.

Supervisory command confidence concerns whether participants understand the available action and its operational meaning. The evaluation should not test manual driving ability. Instead, it should examine whether participants can distinguish actions such as monitor, hold, request confirmation, continue with restrictions, communicate, acknowledge, complete, or escalate. It should also check whether protected emergency actions are clearly separated from routine responses.

Perceived workload concerns whether the multi-screen layout feels manageable. The prototype separates fleet overview, selected-event response, and persistent vessel monitoring across different screens. This structure may reduce view switching, but it may also increase visual load if information hierarchy is unclear. The evaluation should therefore examine whether participants can move between the left screen, right screen, and 24-inch screens without confusion.

Overall, the evaluation aim is to review whether the final interface concept is understandable, coherent, and defensible as a remote supervisory interface. The evaluation does not validate the system as operationally safe or technically complete. It reviews whether the design responds to the thesis requirements and whether it can support scenario-based discussion of monitoring, communication, and supervisory command.

7.2 Evaluation Method

The evaluation was designed as a lightweight design review of the final ROC prototype. It combined expert review, scenario-based walk-through, and SAGAT-inspired freeze questions. This approach was selected because the prototype is a conceptual high-fidelity interface, not a fully functional simulator, certified ROC system, or operational vessel-control environment. The method therefore focused on interface understanding, situation interpretation, priority recognition, automation recommendation understanding, and supervisory response judgement.

This method follows the evaluation direction defined for the thesis: expert review, scenario-based walk-through, SAGAT-inspired questions, and perceived workload assessment. These methods are suitable because the thesis evaluates the clarity and defensibility of an interface concept, rather than measuring real vessel-control performance.

SAGAT was used as an inspiration rather than as a full experimental method. In the original SAGAT method, a simulation is paused at unpredictable moments, displays are hidden, and participants answer questions about the current situation.

The accuracy of the answers is then used as a measure of situation awareness. In this thesis, the method was simplified because the prototype did not include a live simulation environment. Instead, freeze-style questions were used during scenario review to check whether participants could identify key information, understand the selected situation, and predict what may happen next.

The method was also informed by recent discussion of human factors measurement in remote shipping. Grundmann et al. (2026) argue that measurement methods such as SAGAT may need to be adapted for remote-operation contexts, including multiple-vessel monitoring, sensor fusion, and mediated perception. This is relevant for the present thesis because the interface does not represent a traditional ship bridge. It represents a distributed ROC workstation where the operator must interpret mediated information across several screens.

7.2.1 Participants

The evaluation was intended for participants with relevant knowledge of maritime operation, autonomous transport, HMI design, UX design, or safety-critical interface design. Since autonomous urban ferry ROC operation is still an emerging professional role, the study did not require participants to be experienced ROC operators. Instead, the review focused on whether the interface logic was understandable, credible, and defensible for people who could evaluate remote supervision, maritime operation, or complex interface design.

The evaluation was conducted internally within the Cstrider project with two participants. Both participants were familiar with the project context and had relevant knowledge of autonomous ferry operation, remote supervision, and the practical development context of the Cstrider system. They were recruited through the project network and took part as internal experts and stakeholders rather than as statistically representative end users.

Due to the small number of participants, the evaluation should be understood as a formative expert and stakeholder review. Its purpose was not to produce generalisable results or statistically validate the prototype, but to examine whether the interface concept was understandable, coherent, and relevant for the Cstrider operational context. The review focused on the clarity of the multi-screen layout, the distinction between monitoring and supervisory command, the visibility of critical information, and the logic of operator response.

The high-fidelity prototype was used as the main evaluation material during the review session. Participant feedback was documented through researcher notes and was later analysed as design-level feedback, focusing on confirmed design logic, unclear interface elements, and possible future improvements.

7.2.2 Evaluation Material

The evaluation material consisted of the final high-fidelity prototype and scenario prompts covering normal fleet monitoring, shuttle operation, on-demand operation,

warning-level degraded state, passenger assistance, and protected action cases. The scenarios were selected to represent the main design scope of the thesis: monitoring, communication, and supervisory command. They should cover both normal and abnormal situations. The scenarios may include:

Table 7.2: Scenario types used in the evaluation

Scenario type	Purpose
Normal fleet monitoring	To check whether participants understand vessel positions, service mode, and general fleet state.
Shuttle operation	To check whether participants understand repeated route operation, departure / arrival cycle, and dock-related state.
On-demand operation	To check whether participants understand request-based service, assigned vessel, waiting time, and service impact.
Warning-level degraded state	To check whether participants understand alert reason, evidence, recommendation, and available response.
Passenger assistance case	To check whether communication access and passenger-related evidence are understandable.
Critical or protected action case	To check whether emergency response options are clearly separated from routine supervisory actions.

The scenario prompts should not ask participants to manually steer or control the vessel. They should ask participants to interpret the interface and explain what they would do next as a ROC operator.

7.2.3 Procedure

Each evaluation session followed four steps.

First, the participant was introduced to the project context, the thesis scope, and the purpose of the evaluation. The participant was informed that the prototype is a design concept for monitoring, communication, and supervisory command, not a complete operational system.

Second, the participant was shown the screen structure of the prototype. This included the left screen for fleet monitoring, the right screen for selected-event review and command response, and the 24-inch screens for read-only vessel monitoring.

Third, the participant went through scenario-based walk-through. For each scenario, the participant was asked to inspect the relevant screens and explain what they understood from the interface. The researcher then asked questions about situation awareness, priority, automation recommendation, communication need, and possible supervisory response.

Fourth, the participant gave feedback on the overall interface. This included comments on layout, visual hierarchy, information density, terminology, alert levels, command labels, camera use, service mode representation, and perceived workload.

7.2.4 SAGAT-Inspired Freeze Questions

The freeze questions were used to check whether participants could understand the situation from the interface at specific moments in a scenario. They were inspired by the three levels of situation awareness: perception, comprehension, and projection.

Table 7.3: SAGAT-inspired situation awareness questions

SA level	Example question	Evaluation focus
Perception	Which vessel is currently selected? What alert level is active? What mode is the vessel in?	Whether key information is visible and noticed.
Comprehension	Why does this vessel need attention? What is the system recommending? What is the service impact?	Whether participants understand the meaning of the information.
Projection	What may happen if no action is taken? What should the operator check next?	Whether participants can reason about the near-future situation.

The questions were not used as a strict quantitative SAGAT score. Instead, they were used as structured prompts during the design review. This is more appropriate for the current prototype because there is no live simulation, no hidden display condition, and no controlled comparison between interface versions.

7.2.5 Feedback and Workload Questions

After the scenario walk-through, participants were asked to comment on perceived workload and interface manageability. The focus was not on producing a full NASA-TLX workload score. Instead, the questions were used to collect design-level feedback on whether the interface felt visually dense, cognitively manageable, and logically structured.

The questions included:

- Was it clear which vessel or event needed attention first?
- Was it clear where to look for detailed information?
- Was the relationship between alert, evidence, recommendation, and response understandable?

- Did the multi-screen layout feel manageable?
- Were any areas visually overloaded?
- Were any command labels unclear or too risky?
- Was it clear that the 24-inch vessel screens were monitoring screens rather than command screens?

This helped connect participant feedback to specific design issues rather than only collecting general opinions.

7.3 Evaluation Results

The evaluation results should be understood as formative design feedback rather than as statistical validation. Since the review was conducted with two internal Cstrider participants, the findings mainly indicate whether the prototype was understandable, coherent, and relevant within the project context.

Overall, the participants were able to understand what was happening in the prototype and how different parts of the interface related to different types of information. The general division between fleet monitoring, selected-vessel information, camera evidence, alerts, and supervisory response was considered understandable. This suggests that the multi-screen structure and the basic information architecture were broadly readable. The participants could follow the intended logic from identifying a situation, checking related information, and considering an appropriate response.

At the same time, the review revealed several areas that need further refinement. One issue was visual legibility. Some text was perceived as too small, especially in information-dense areas. This could become problematic in a real ROC setting, where the operator may need to read information quickly across multiple screens. Some colour mappings were also unclear or inconsistent. For example, the relationship between alert colours, vessel status colours, and interface highlights was not always immediately obvious. This indicates that the colour system needs to be more consistent and more tightly connected to operational meaning.

The participants also pointed out that the amount of information was generally sufficient, but that some information did not fully match between the left and right screens. This means that the prototype provided enough content for understanding the situation, but cross-screen consistency needs improvement. When the same vessel, event, or alert appears on different screens, the naming, status level, colour coding, and displayed priority should be aligned. Otherwise, the operator may need to spend extra effort checking whether the screens refer to the same situation.

Another issue concerned time-related information. The current time predictions

were not intuitive enough. Although the interface included time or ETA-related data, the meaning of these values was not always easy to interpret at a glance. This suggests that time prediction should be presented more visually and actionably, for example through clearer progress indicators, countdowns, expected delay labels, or impact statements. Instead of only showing a time value, the interface should help the operator understand what the time means for the vessel, passengers, service reliability, or required response.

The review also identified some repeated information. Some status details appeared in more than one location without adding new meaning. Repetition can be useful when it supports cross-screen awareness, but unnecessary repetition may increase visual load and reduce clarity. The prototype should therefore distinguish between purposeful redundancy and avoidable duplication. Critical information can be repeated across screens, but secondary information should be placed only where it supports the operator's decision-making process.

A further finding was that some technical values should be translated into more evaluative information. For example, showing a signal delay as a raw number, such as 28 ms, may be less useful than showing whether the connection is strong, acceptable, degraded, or critical. For supervisory operation, the operator does not always need the raw technical value first. In many cases, the interface should first communicate the operational meaning of the value, and then allow the detailed number to be accessed if needed. This applies to signal latency, sensor confidence, positioning quality, communication status, and other system health indicators.

Finally, the review showed that the distinction between clickable buttons and non-clickable information cards was not always clear enough. Some cards looked interactive, while some buttons did not stand out sufficiently as available actions. This is an important issue because the prototype separates monitoring from supervisory command. If the visual difference between information display and action control is weak, the operator may be uncertain about where action is possible. Future refinement should therefore strengthen interaction affordance through clearer button styling, hover or active states in the interactive prototype, disabled states, and more consistent card hierarchy.

In summary, the evaluation confirmed that the overall interface concept was understandable and that the main screen division supported the intended supervisory logic. However, the results also showed that the prototype requires further refinement in visual legibility, colour consistency, cross-screen alignment, time representation, information redundancy, qualitative system-status presentation, and interaction affordance. These findings do not invalidate the design direction. Instead, they define the next level of development needed before the interface could be tested in a more realistic simulation or expert evaluation setting.

7.4 Design Revisions after Evaluation

The formative internal review also identified several interface details that should be improved in the next design iteration. These issues do not change the overall direction of the prototype, but they indicate where the interface needs further refinement before more realistic simulator-based testing or expert evaluation. The next stage should focus on improving readability, cross-screen consistency, colour semantics, time representation, information hierarchy, and the distinction between interactive and non-interactive elements.

Table 7.4: Summary of design revisions after evaluation

Evaluation finding	Design revision
Some text was too small	Increase the size of key labels, status text, and alert information; reduce secondary details in dense areas.
Colour meanings were not always clear	Standardise colour use for alert level, vessel status, selected state, active state, and disabled state.
Left and right screens did not fully match	Align vessel names, alert levels, status indicators, event labels, and priority coding across screens.
Time prediction was not intuitive enough	Present time as action-oriented information through countdowns, progress indicators, delay labels, and impact statements.
Some information was repeated	Keep purposeful redundancy for critical information, but remove or reduce repeated secondary details.
Raw technical numbers were not always meaningful	Translate technical values into evaluative status categories, such as stable, degraded, weak, or critical.
Buttons and cards were visually too similar	Strengthen the difference between clickable actions and passive information cards through clearer styling and states.

8

Discussion

This chapter begins by reflecting on whether and how the aim of the thesis was achieved. The aim of this thesis was to explore how a Remote Operation Centre interface could support shore-based supervision of autonomous urban passenger ferries, with a focus on monitoring, communication, and supervisory command. This aim was addressed through a Research through Design process that combined literature review, expert and stakeholder input, scenario-based reasoning, iterative prototyping, and a formative design review.

Overall, the aim was achieved at the level of an interface concept. The final prototype demonstrates how ROC work can be organised across a multi-screen environment, where fleet-level monitoring, selected-event reasoning, vessel-specific awareness, communication access, and bounded supervisory responses are separated but connected. In this sense, the project responds to the aim by translating the abstract problem of remote ferry supervision into a concrete HMI proposal.

However, the aim was not achieved as operational validation. The prototype should not be understood as a certified control system or as proof that the interface improves situation awareness, reduces workload, or increases safety in real operation. The evaluation indicated that the overall interface logic was understandable and helped identify areas for refinement, but it was limited in scale and formative in character. Therefore, the main achievement of the thesis is a research-grounded, design-grounded, and evaluation-informed interface proposal that can support further expert review, simulator testing, and development of ROC workstations for autonomous urban ferry operation.

8.1 Design Contribution

This thesis contributes to a conceptual ROC interface proposal for autonomous urban ferry operation. The contribution is not a deployable control system, navigation algorithm, certified safety case, or complete remote-control station. Instead, it is an interface-level design contribution that shows how monitoring, communication, and bounded supervisory command can be organised for shore-based supervision of multiple autonomous passenger ferries.

The first contribution is the definition of the ROC operator's role as a supervisory role rather than a remote-driving role. The final design does not position the operator as someone who continuously steers, throttles, docks, or manually controls the

vessel. Instead, the operator is supported in monitoring fleet state, understanding selected vessel or event context, communicating with relevant actors, and issuing limited supervisory responses when automation requires human attention. This role boundary is important because an unclear operator role can easily lead to an interface that is either too passive or too control-heavy. In this thesis, supervisory command is therefore treated as a bounded action layer, not as full manual teleoperation.

The second contribution is a layered multi-screen information architecture. The final prototype separates the operator's work into fleet-level monitoring, selected-event command handling, and persistent single-vessel awareness. The left screen supports fleet overview and priority recognition. The right screen supports event diagnosis, evidence review, communication, and response. The five 24-inch vessel screens support read-only per-vessel monitoring. The contribution is not simply the use of several screens, but the distribution of attention and interaction across these screens. This structure helps distinguish overview, focus, and response without turning every screen into a command surface.

The third contribution is a mode-aware supervision logic for urban ferry service. The design treats shuttle and on-demand operation as different service contexts that create different monitoring needs. Shuttle mode is represented as a more regular corridor-based operation, where the operator needs to understand route continuity, dock state, departure or arrival phase, and delay. On-demand mode is represented as a more flexible request-based service logic, where the operator needs to understand vessel assignment, passenger demand, waiting time, and service impact. This connects the maritime safety layer with the urban mobility service layer. The ROC interface is therefore not only a vessel monitoring tool, but also a service supervision interface.

The fourth contribution is a prioritised multi-vessel supervision concept. The prototype does not repeat a traditional bridge interface for each vessel. Instead, it supports attention management through fleet status, vessel cards, alert levels, command queue, selected-vessel detail, and persistent vessel monitoring. This responds to the problem that multi-vessel supervision is not only a matter of display capacity. The operator needs to know which vessel or event requires attention first, why it matters, and whether action is needed. The design therefore uses priority, urgency, service impact, and actionability as organising principles.

The fifth contribution is an alert-to-action logic for supervisory command. In the final design, alerts are not treated as isolated warnings. They are connected to cause, impact, system recommendation, evidence, communication options, and possible response. This is especially important because remote operation depends on mediated information and system recommendations. The operator should not be asked to act on a warning without understanding what happened, what the system recommends, what evidence supports it, and what consequence the response may have. The right-screen command workspace and response bar are therefore designed as a connected decision flow rather than as separate panels or generic buttons.

The sixth contribution is the deliberate limitation of command surfaces. The 24-inch vessel screens are defined as read-only monitoring displays, not as small command stations. Left-screen actions are treated as access points or protected entry points, while detailed command review remains on the right screen. Routine autonomous actions, such as ordinary route following, routine docking, normal departure, and ordinary obstacle avoidance, are not treated as repeated approval tasks for the ROC operator. They can be logged and monitored, but they should not become command queue items by default. This keeps the interface aligned with remote supervision rather than remote driving.

Together, these contributions define the design logic of the thesis. The final prototype proposes a way to organise ROC supervisory work across screens, information layers, service modes, alert structures, and response actions. Its value lies in making the operator's remote supervisory task more explicit: maintaining fleet awareness, identifying priority, understanding automation, communicating in context, and responding through bounded commands.

The contribution should be understood within the limits of the project. A formative internal evaluation was conducted with two Cstrider participants, and the feedback indicated that the overall interface logic was understandable while also identifying several areas for refinement. However, due to the limited number of participants and the formative nature of the review, the thesis does not claim that the prototype improves situation awareness, reduces workload, improves safety, or increases operational performance in real operation. These claims would require further evaluation with a larger and more diverse participant group, realistic scenarios, simulator-based tasks, and formal human factors measures. The contribution of this thesis is therefore a research-grounded, design-grounded, and evaluation-informed interface concept that can support future expert review, simulator testing, and ROC workstation development for autonomous urban ferry operation.

8.2 Reflection on the Design and Research Process

The thesis process combined literature review, expert and stakeholder input, interview-based analysis, iterative prototyping, and formative evaluation. This combination was useful because the project addressed an emerging work context where fixed requirements were not available from the beginning. The Remote Operation Centre for autonomous urban passenger ferries is not yet a mature and standardised workplace. Therefore, the process needed to move between research, interpretation, and design exploration rather than follow a linear problem-solving model.

The literature review was important for framing the project. It helped define remote operation as supervisory work rather than direct remote driving, and it clarified several human factors challenges such as situation awareness, loss of ship sense, fragmented information, automation transparency, alerting, and multi-vessel monitoring. However, the literature alone could not define the final interface. Much

of the existing research focuses on larger vessels, remote pilotage, cargo ships, or general ROC operation. The specific context of small autonomous passenger ferries in an urban public transport system required further interpretation through the Cstrider project context.

Expert and stakeholder input helped connect the literature to a more realistic operational setting. These discussions clarified that the ROC operator should not approve every routine vessel action. Instead, the operator should mainly monitor the fleet, understand system-initiated exceptions, communicate when needed, and intervene when safety, service continuity, passenger welfare, or operational boundaries are affected. This input strongly influenced the final command scope and helped prevent the interface from becoming a manual teleoperation cockpit.

The interview and coding process supported the transition from broad research themes to design-oriented findings. The analysis made it possible to organise issues such as trusted information sources, vessel-state understanding, departure readiness, passenger communication, alert-to-action logic, and command responsibility. This was useful because it created a bridge between qualitative input and concrete interface requirements. At the same time, the process was limited by the number and type of participants. The findings should therefore be understood as design-oriented insights rather than a complete model of future ROC operator work.

Prototyping was not only used to visualise a solution, but also to clarify the problem. Early sketches and wireframes revealed that the main design challenge was not simply where to place information on screens. The more important issue was how to structure the operator's movement from fleet overview, to selected-event understanding, to supervisory response. Iteration also helped clarify the roles of the different screens. The left screen became focused on fleet-level monitoring and priority recognition, the right screen became the command workspace, and the 24-inch vessel screens became read-only awareness displays for individual vessels. In this way, prototyping functioned as a method for reasoning about work structure, not only as a method for producing visual output.

The evaluation was designed as a lightweight design review because the prototype was a high-fidelity interface concept rather than a functioning simulator or operational control system. This was appropriate for the maturity of the project. The evaluation could check whether the interface logic was understandable, whether participants could follow the scenario-based reasoning, and whether the relationship between alerts, evidence, recommendations, and response options was clear. However, this method could not validate operational performance. It did not measure actual workload reduction, safety improvement, or situation awareness improvement in a live ROC environment.

Overall, the process was coherent because each method had a different role. The literature review framed the problem, stakeholder input grounded the operational assumptions, interviews and coding translated concerns into design themes, prototyping tested how these themes could become an interface, and evaluation provided

formative feedback on clarity and coherence. The main limitation of the process was that it remained concept-driven and formative. Future work would need closer involvement with actual operators, simulator-based scenarios, more systematic usability testing, and stronger integration with technical system constraints.

8.3 Reflection on Remote Operation as Cognitive Work

A central reflection from this thesis is that remote operation should not be understood as simply moving the ship bridge to shore. The work itself changes. In an onboard bridge context, operators are physically situated in the operating environment and can use visual, auditory, bodily, environmental, instrumental, and social cues to understand the vessel and its surroundings. In a ROC context, many of these cues become indirect, mediated, delayed, or unavailable. The operator must therefore reconstruct the operational situation through digital representations.

This means that remote operation is not only a technical relocation of control. It is a transformation of maritime work into distributed cognitive supervision. The operator's role shifts from direct situated control towards monitoring, interpretation, prioritisation, automation verification, communication, and accountable intervention. This supports the main argument of the thesis: the ROC is not a remote-driving cockpit, but a supervisory work environment.

This reflection is important for interface design. If the ROC is treated as a digital copy of the bridge, the design may focus on reproducing instruments, camera views, radar displays, and control panels. However, this does not solve the central cognitive problem. The operator still needs to understand which vessel matters, what the automation is doing, whether the information is reliable, what action is possible, and who is responsible for the next step. Therefore, the interface must organise information around the operator's reasoning process, not only around technical data sources.

The final prototype responds to this problem by structuring supervision across fleet, vessel, and event levels. The left screen supports fleet-level awareness and priority recognition. The right screen supports selected-event diagnosis, automation explanation, communication, and supervisory response. The 24-inch vessel screens support persistent read-only vessel awareness. Together, these layers aim to reduce hidden mental work by making the operator's supervisory reasoning more visible.

Automation should also be understood as redistributing work rather than removing work. The operator may not continuously steer the vessel, but still needs to understand automation behaviour, judge system recommendations, and intervene when a situation becomes uncertain, service-critical, passenger-related, or safety-critical. For this reason, the prototype connects alerts, evidence, recommendations, communication, and supervisory commands within one selected-event flow.

This also explains why the thesis uses the term supervisory command instead of control. Control can imply continuous manual handling, joystick operation, or direct vessel manoeuvring. The supervisory command is narrower. It refers to bounded high-level actions, such as holding an operation, requesting confirmation, contacting relevant actors, continuing with restrictions, acknowledging an event, or escalating an abnormal case. This distinction keeps the interface aligned with the operator's actual responsibility in the proposed Cstrider context.

In summary, remote operation reconfigures maritime work from situated physical control into distributed cognitive supervision. The design lesson from this thesis is that a ROC interface should support the operator as an interpreter, supervisor, communicator, and accountable decision-maker, rather than as a continuous remote driver.

8.4 Implications for ROC Interface Design

The design work in this thesis suggests that ROC interface design should be treated as a work-domain design problem, not only as a visual dashboard problem. Remote operation changes the operator's role, information access, sensory conditions, attention demands, and responsibility boundaries. Therefore, future ROC interfaces should be designed around how operators understand, prioritise, communicate, and intervene in a distributed autonomous system. The first implication is that ROC

interfaces should not directly copy the traditional ship bridge. Bridge instruments, radar displays, camera views, conning information, and nautical charts may still be useful, but they need to be reorganised around shore-based supervision. A shore-based operator does not have the same physical presence, sensory cues, or bridge-team interaction as an onboard officer. Remote pilotage research also shows that moving work from ship to shore changes information quality, sensory access, communication, and work strategy (Sharma et al., 2025). For this reason, ROC interface design should focus less on reproducing bridge layout and more on supporting remote situation reconstruction, exception handling, and automation supervision.

The second implication is that multi-vessel supervision needs a fleet-level attention structure. A ROC interface should not be designed only as a single-vessel control panel. The operator needs to compare vessels, understand fleet status, identify which vessel or event requires attention, and move into detail without losing the wider operational picture. This is consistent with remote supervisory control research, where multi-vessel supervision and reduced available time can affect intervention performance (Veitch et al., 2024). Therefore, fleet interfaces should make priority visible through alert level, vessel state, service impact, urgency, and actionability.

The third implication is that automation recommendations need to be explainable before they become actionable. Automation does not remove humans from the system. It changes the operator's work from direct control to supervision, verification, and accountable intervention. A recommendation should therefore not appear as a simple instruction to accept or reject. The interface should show what the system

recommends, why it recommends it, what evidence supports it, what uncertainty remains, and what consequence the response may have. This is especially important because decision support systems are meant to support expert judgement rather than replace it (Veitch et al., 2024).

The fourth implication is that command interfaces should be bounded and contextual. Supervisory commands should appear only when they are connected to a specific vessel, event, reason, and expected consequence. Routine autonomous actions should not become repeated approval tasks. Instead, they should be logged and monitored unless they create uncertainty, service impact, passenger risk, operational boundary issues, or safety escalation. This prevents the interface from becoming a remote-driving cockpit and keeps the operator's role closer to supervisory command.

The fifth implication is that communication should be embedded in an operational context. Remote operation increases the importance of coordination between the ROC operator, onboard safety crew, passengers, support actors, and the autonomous vessel system. If communication is separated from the selected event, the operator must manually reconstruct who to contact and why. A better design connects communication to the current vessel, missing confirmation, passenger issue, degraded state, or escalation path. This also reflects the wider human-interaction problem in remote ship operations, where human-human and human-machine relationships become part of system performance (Kari et al., 2018).

The sixth implication is that service context should be represented together with vessel safety. In the Cstrider case, the ROC operator does not only monitor technical vessel state. The operator also supervises an urban ferry service. Shuttle mode, on-demand mode, passenger waiting time, route continuity, dock state, and service disruption affect how an event should be interpreted. A warning on one vessel may matter more if it blocks a high-demand route, affects boarding, or delays an on-demand request. ROC interfaces for autonomous passenger ferries should therefore connect maritime supervision with service operation.

The seventh implication is that interface design should make responsibility boundaries visible. The operator needs to know what the autonomous system has handled, what the onboard safety crew has confirmed, what remains uncertain, and what belongs to the ROC operator. This is especially important for routine phases such as departure, docking, and boarding. If every routine step is shown as an operator approval task, the interface may increase workload and blur accountability. If too little is shown, the operator may lose confidence and situation awareness. The interface should therefore show evidence, status, and authority boundaries rather than simply adding more controls.

Overall, these implications point toward a specific direction for future ROC HMI design. The interface should not be organised around maximum data visibility or maximum control availability. It should be organised around the operator's supervisory reasoning process: maintaining overview, identifying priority, understanding

automation, communicating in context, and carrying out bounded intervention when needed.

8.5 Ethics and Sustainability Reflections

This thesis concerns the design of a remote operation centre interface for autonomous passenger ferries. Therefore, the design result should not only be discussed as an interface proposal, but also as part of a wider socio-technical system. The interface may influence how operators understand remote situations, how decisions are made, how passengers are supported, and how responsibility is distributed between humans and automation.

From an ethical perspective, a central issue is the responsibility of the remote operator. In the proposed system, the operator is not expected to manually control the vessel during normal operation. Instead, the operator supervises several vessels, reviews exceptions, communicates when needed, and gives high-level supervisory commands. This role can reduce continuous manual workload, but it can also create a new risk: the operator may be asked to intervene in complex situations after not being directly involved in the ongoing control process. For this reason, the interface should make the situation understandable before asking for action. This influenced the design of the command workspace, where alerts, system recommendations, sensor information, camera views, and response options are placed together to support decision-making.

Accountability is another important issue. If a system recommendation is accepted, rejected, or modified by the operator, the reason for the decision should be clear and traceable. This is important in safety-critical work, because decisions may need to be reviewed after an incident or near miss. In the prototype, supervisory commands are therefore treated as explicit actions. Routine autonomous behaviour is separated from operator-level decisions, so the operator is not asked to approve every normal action. Instead, the operator becomes involved when a case affects safety, passenger welfare, service state, or operational boundaries.

Passenger safety and trust also need to be considered. An autonomous passenger ferry is not only a technical vessel system; it is also a public transport service. Passengers may need help, information, or reassurance when there is no traditional bridge crew visible. For this reason, the interface includes passenger-related information and communication functions. However, this also raises a privacy concern. Camera views and cabin information should not be used as general surveillance. Their role should be limited to operational safety, passenger assistance, and situation confirmation.

From a sustainability perspective, electric and automated ferries may support more sustainable urban mobility by making waterborne public transport more flexible and scalable. In the Gothenburg context, this could help improve river crossings and reduce pressure on land-based transport. However, this thesis does not directly

prove environmental benefits. It does not include life cycle assessment, energy analysis, traffic simulation, or a quantitative comparison with buses, trams, cars, or conventional ferries. Therefore, the sustainability contribution of this thesis should be understood at the design and service-system level. The proposed ROC HMI explores how remote ferry operation could become more understandable, manageable, and operationally scalable.

Overall, the thesis does not claim that an interface alone can make autonomous ferry operation ethical or sustainable. These outcomes also depend on regulation, training, organisational routines, cybersecurity, maintenance, energy supply, and public acceptance. The contribution of this work is more limited but still relevant: it shows how interaction design can support clearer remote supervision, more transparent decision-making, and a more accountable relation between human operators and autonomous ferry systems.

8.6 Use of AI in the Thesis Process

Generative AI tools were used during this thesis as a supporting tool for reflection, writing, translation, and design development. The use of AI was not treated as a research method and did not replace the author's own design decisions, analysis, or evaluation. Instead, AI was used as a tool to support the working process, especially when organising report structure, refining English text, summarising design arguments, comparing alternative formulations, and checking whether the connection between research findings and design decisions was clear.

AI tools were also used during the design process to support ideation and critique. For example, they helped generate alternative ways of describing the ROC interface, question whether screen roles were clear enough, and identify possible problems in information hierarchy, command boundaries, and visual consistency. In some cases, AI-assisted image generation was used to explore visual presentation material and communicate the intended atmosphere of the final Remote Operation Centre concept. These outputs were used as design support and presentation material, not as empirical evidence.

The author remained responsible for selecting, modifying, and verifying all content included in the thesis. Literature-based claims were checked against actual sources, and AI-generated suggestions were not used as references unless they could be connected to real literature, project material, design decisions, or evaluation findings. This was important because generative AI can produce plausible but inaccurate statements, especially when discussing technical, regulatory, or research-specific topics.

The use of AI also had limitations. AI support may influence wording, framing, and the way design arguments are structured. Therefore, the thesis process required continuous checking to ensure that the final report still reflected the actual project scope, the developed prototype, and the evaluation results. In particular, claims

about situation awareness, workload, safety, and operational performance were kept limited to what the project could reasonably support. The thesis does not treat AI-generated text or images as independent evidence. They are understood as process support within a human-led design and writing process.

Overall, AI contributed to the efficiency and clarity of the thesis process, but the final academic argument, design direction, prototype decisions, and interpretation of evaluation feedback remained the responsibility of the author.

8.7 Limitations

This thesis has several limitations. These limitations define what can reasonably be claimed from the project and what should be investigated in future work. The final prototype should be understood as a conceptual ROC interface proposal. It is not a validated operational system, a certified safety-critical interface, or a complete remote-control solution.

The first limitation concerns the empirical input. The project used input from maritime professionals, Cstrider stakeholders, supervisors, and HMI/design experts. This was appropriate because autonomous urban ferry ROC operation is still an emerging work context, and there are few people with direct experience of operating this exact type of system. However, the participants were not actual Cstrider ROC operators working in a live operational environment. Therefore, the findings should be understood as informed design input rather than a complete representation of future operator needs.

The second limitation concerns evaluation. A formative internal evaluation was completed with two Cstrider participants. However, this review was limited in scale and should not be treated as statistical validation or operational performance testing. Therefore, the thesis cannot claim that the interface improves situation awareness, reduces workload, increases decision confidence, improves safety, or improves operational performance.

The third limitation concerns operational realism. The prototype does not use live AIS, radar, LiDAR, camera, weather, passenger request, vessel-control, or traffic data. It also does not simulate real vessel dynamics, communication delay, degraded sensor behaviour, passenger demand, dispatch optimisation, or traffic conflict. The shuttle and on-demand representations should therefore be understood as interface-level service concepts, not as validated operational or dispatch models.

The fourth limitation concerns technical implementation. The prototype is a high-fidelity interface design, not a working software system. It does not include back-end integration, real-time event processing, alarm logic implementation, cybersecurity, system reliability calculation, data-link management, or multi-display synchronisation. These technical layers are necessary for real ROC deployment, but they are outside the scope of this design thesis.

The fifth limitation concerns regulation, organisation, and responsibility. The thesis explores interface-level concepts such as onboard crew confirmation, ROC supervisory veto, protected emergency actions, and bounded supervisory command. However, it does not define the legal authority of the ROC operator, the certification process, the staffing model, the complete safety case, or the organisational procedures needed for real-world operation. These issues would need to be defined before the concept could be used in practice.

The sixth limitation concerns command scope. The thesis deliberately excludes full manual teleoperation. The prototype does not define joystick control, direct vessel handling, throttle control, manual docking, emergency takeover procedures, or fallback control modes. It focuses on monitoring, communication, and supervisory command. This boundary makes the design scope more precise, but it also means that severe failure recovery and direct remote-control operation remain unresolved design problems. This is consistent with the final project scope, where full manual teleoperation and direct vessel control are treated as out of scope or future work.

The seventh limitation concerns visual and interaction maturity. Although the final prototype uses an OpenBridge-inspired maritime HMI style, it remains a design prototype. Some interface elements, such as map abstraction, alert terminology, protected action behaviour, event taxonomy, and command confirmation logic, require further refinement. The prototype should therefore be treated as a structured design proposal rather than a finished product specification.

Overall, these limitations mean that the thesis should be read as a design-oriented exploration of ROC interface structure. It provides a grounded proposal for organising monitoring, communication, and bounded supervisory command, but it does not provide operational validation, technical implementation, regulatory approval, or proof of safety performance.

8.8 Future Work

Several directions for future work follow from the limitations of this thesis. Since the current project focuses on a conceptual ROC interface for monitoring, communication, and bounded supervisory command, future work should extend the prototype toward stronger evaluation, more realistic operation, technical integration, and broader organisational definition.

The first direction is formal evaluation. The final prototype should be tested with maritime professionals, HMI experts, Cstrider stakeholders, and, if available, future ROC operators. Future evaluation could use scenario-based walkthroughs, expert review, SAGAT-inspired questions, workload-related measures, response-time observations, and post-scenario interviews. This would make it possible to examine whether the interface supports situation awareness, priority recognition, automation recommendation understanding, mode awareness, command confidence, and perceived workload. Such evaluation should be clearly separated from operational

validation. It can assess interface clarity and usability, but it would not by itself certify the system for real ferry operation.

The second direction is operational data integration. The current prototype uses conceptual representations of vessel state, routes, alerts, service mode, passenger state, and system recommendations. Future work should explore how the interface behaves with live or simulated AIS, radar, LiDAR, camera, weather, battery, communication link, crew confirmation, docking, and passenger request data. This is important because remote supervision depends on mediated information, and such information may be delayed, incomplete, uncertain, or conflicting. Future work should therefore examine not only what information is shown, but also how information quality, confidence, delay, and conflict are represented.

The third direction is dynamic service and traffic simulation. The current prototype represents shuttle and on-demand operation at the interface level, but it does not implement real fleet dispatch logic, route optimisation, passenger demand prediction, or traffic conflict simulation. A more advanced simulation could test how operators understand vessel assignment, service mode change, passenger waiting time, fleet balance, berth availability, and route conflict under dynamic conditions. This would be especially useful for evaluating whether the interface can support both maritime safety supervision and urban ferry service supervision.

The fourth direction is a more detailed event, alert, and escalation model. The current design uses Info, Caution, Warning, and Critical levels to connect alert priority with operator role. Future work should define a more complete event taxonomy, including how events are detected, how severity is assigned, when the operator should be interrupted, what evidence should be shown, what response options are allowed, and when escalation is required. This should cover event types such as degraded positioning, route conflict, docking uncertainty, berth blockage, passenger assistance, onboard crew confirmation failure, communication loss, and emergency fallback.

The fifth direction is technical implementation. Future development would need to connect the interface to backend systems, real-time data processing, alarm logic, communication channels, vessel state models, service management systems, and multi-display synchronisation. Cybersecurity, data reliability, latency management, system redundancy, and failure-state design would also need to be addressed before the interface could become part of a real ROC system. These are engineering and safety-critical system issues, not only interaction design issues.

The sixth direction is full manual teleoperation and emergency takeover. This thesis deliberately excludes joystick control, throttle control, manual docking, detailed path drawing, and direct vessel handling. However, severe failure recovery may require some form of protected manual or semi-manual fallback. Future work should treat this as a separate design and research problem. It would require simulator testing, technical control architecture, regulatory analysis, training procedures, and clear responsibility allocation. It should not simply be added as a button inside the current supervisory command interface.

The seventh direction is organisational and regulatory development. The current thesis explores interface-level ideas such as onboard crew confirmation, ROC supervisory veto, contextual communication, and protected emergency actions. Future work should define how these concepts relate to legal authority, certification, staffing models, operator training, shift handover, liability, emergency procedures, and collaboration between multiple ROC operators. Without these organisational definitions, the interface logic cannot be fully translated into real operation.

The eighth direction is passenger-side and service-side integration. Since the Cstrider concept concerns autonomous passenger ferries in an urban mobility context, future work should also examine how passenger requests, passenger assistance, service disruption, boarding information, and public transport coordination are represented across the wider system. This may include passenger-facing applications, dock-side displays, crew tools, service management dashboards, and communication protocols between passengers, onboard personnel, and the ROC.

Together, these directions show that the final prototype should be treated as a foundation for further design and research rather than a completed system. The next step is not only to make the interface more visually refined. It is to test the interface in realistic scenarios, connect it to more reliable data and simulation environments, define event and responsibility models, and investigate how the ROC would operate as part of a larger socio-technical ferry system.

9

Conclusion

This thesis explored how a Remote Operation Centre interface could support the supervision of autonomous passenger ferries in an urban waterway context. The project focused on Cstrider’s future ferry operation in Gothenburg and addressed the interface needs of a shore-based operator who supervises multiple vessels rather than continuously drives one vessel manually.

The central argument of the thesis is that remote ferry operation should not be understood as a simple transfer of the ship bridge to shore. Remote operation changes the operator’s work. The operator no longer has direct physical access to the vessel, passengers, dock, traffic, weather, and surrounding environment. Instead, the operator must build understanding through mediated information, including maps, vessel data, alerts, camera views, system recommendations, communication, and event history. This makes the ROC a new supervisory work environment rather than a copied bridge interface. Remote pilotage research also highlights similar challenges, including indirect information, reduced physical co-location, communication demands, and changed work strategies in shore-based maritime operation (Sharma et al., 2025).

The thesis therefore focused on three interface functions: monitoring, communication, and bounded supervisory command. Monitoring is needed to maintain awareness of the fleet, selected vessels, service mode, and abnormal events. Communication is needed to clarify uncertain situations, coordinate with onboard safety crew, support passenger-related issues, and handle escalation. Supervisory command is needed when the system reaches uncertainty, service impact, operational boundary, passenger safety concern, or safety-critical escalation. Full manual teleoperation, joystick control, throttle control, manual docking, and certified emergency takeover were not included in the design scope.

The final design outcome is a high-fidelity multi-screen ROC interface concept. The prototype separates the operator’s work across two main interactive screens and five smaller read-only vessel monitoring screens. The left main screen supports fleet-level monitoring and prioritisation through a corridor-based Gothenburg map, vessel list, service mode visibility, status indicators, and selected-vessel summary. The right main screen supports selected-event diagnosis and supervisory response through command queue, tactical view, system recommendation, vessel status, camera and sensor evidence, communication access, and a response bar. The five 24-inch screens support persistent awareness of individual vessels, but deliberately avoid command buttons so that the workstation does not become fragmented into several small

control stations.

This screen structure is one of the main design contributions of the thesis. It organises the ROC interface around overview, focus, and response. The left screen helps the operator identify what requires attention. The right screen helps the operator understand the selected case and decide what response is appropriate. The smaller screens keep individual vessels visible without turning every vessel display into a command surface. This structure responds to the challenge of multi-vessel supervision, where the operator must identify priority rather than inspect every vessel with equal depth. Research on remote supervisory control also shows that number of vessels, available time, and decision support can influence intervention performance, which makes attention structure and decision support important design concerns (Veitch et al., 2024).

The prototype also contributes a specific command logic. Routine autonomous actions, such as ordinary route following, routine docking, routine departure, and normal obstacle avoidance, are not treated as repeated approval tasks for the ROC operator. Instead, they can be monitored and logged. Operator action becomes relevant when a situation requires interpretation, confirmation, communication, or escalation. This is reflected in the alert-to-action logic of the right screen, where alerts are connected to cause, impact, evidence, recommendation, communication, and possible response. The interface therefore avoids treating commands as generic buttons and instead ties them to selected vessel or event context.

Another contribution is the integration of service operation into the ROC interface. Because Cstrider’s concept concerns autonomous urban passenger ferries, the operator is not only supervising vessel safety. The operator also needs to understand shuttle operation, on-demand requests, passenger state, dock conditions, service delay, and fleet availability. The design therefore treats shuttle and on-demand modes as operational contexts that affect what the operator needs to notice and how an event should be interpreted.

The thesis also clarifies several limits. The prototype is not a working software system. It does not include live AIS, radar, LiDAR, weather, passenger request, traffic, dispatch, or vessel-control data. It does not implement backend route optimisation, fleet dispatch logic, certified alarm logic, cybersecurity, regulatory compliance, or real vessel-control functions. A formative internal evaluation was conducted with two Cstrider participants. The review indicated that the overall interface logic was understandable, but also identified issues related to text size, colour consistency, cross-screen alignment, time representation, information repetition, evaluative status presentation, and interaction affordance. Therefore, the thesis cannot claim that the prototype improves situation awareness, reduces workload, improves safety, or increases operational performance. Such claims require broader expert review, simulator-based testing, and formal human factors evaluation.

In conclusion, this thesis provides a design-grounded proposal for how a ROC interface for autonomous urban ferry operation can be structured. Its main contribution is not a finished operational system, but a clearer interface logic for remote supervisory work. The final prototype shows how fleet overview, selected-event diagnosis,

contextual communication, automation recommendation, and bounded supervisory response can be organised across a multi-screen workstation. This provides a foundation for further evaluation, technical development, and future research into ROC interfaces for autonomous passenger ferry services.

Bibliography

- [1] Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101. <https://doi.org/10.1191/1478088706qp063oa>
- [2] City of Gothenburg. (2014). Gothenburg 2035: Transport strategy for a close-knit city. https://goteborg.se/wps/wcm/connect/6c603463-f0b8-4fc9-9cd4-c1e934b41969/Trafikstrategi_eng_140821_web.pdf?MOD=AJPERES
- [3] Ekman, F., & Wallgren, P. (2024). *Opportunities and barriers of electrified and automated passenger ferries in Gothenburg: A pilot study in ElectriCity – WP2* [Pilot study report].
- [4] Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1), 32–64. <https://doi.org/10.1518/001872095779049543>
- [5] Fereday, J., & Muir-Cochrane, E. (2006). Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International Journal of Qualitative Methods*, 5(1), 80–92. <https://doi.org/10.1177/160940690600500107>
- [6] Frayling, C. (1993). Research in art and design. *Royal College of Art Research Papers*, 1(1), 1–5.
- [7] Gernez, E., Mallam, S. C., Nordby, K., Fauske, J. E., Johansen, B., Sandnes, B. K., Wildhagen, S. L., & Lockertsen, H. (2026). The ROC architecture framework for remote ship operations. In T. E. Kim, A. K. Sydnes, B.-M. Batalden, & L. P. Perera (Eds.), *Human factors in remote ship operations* (pp. 116–147). Routledge. <https://doi.org/10.4324/9781003586883-5>
- [8] Grundmann, R., Ujkani, A., & Hohnrath, P. (2026). A systematic analysis of measurement methods for human factors in remote shipping. In T. E. Kim, A. K. Sydnes, B.-M. Batalden, & L. P. Perera (Eds.), *Human factors in remote ship operations* (pp. 148–166). Routledge. <https://doi.org/10.4324/9781003586883-6>
- [9] International Maritime Organization. (2010). *Resolution MSC.302(87): Adoption of performance standards for bridge alert management*.

- [https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/MSResolutions/MSC.302\(87\).pdf](https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/MSResolutions/MSC.302(87).pdf)
- [10] International Organization for Standardization. (2019). *Ergonomics of human-system interaction—Part 210: Human-centred design for interactive systems* (ISO 9241-210:2019). <https://www.iso.org/standard/77520.html>
 - [11] Kallio, H., Pietilä, A.-M., Johnson, M., & Kangasniemi, M. (2016). Systematic methodological review: Developing a framework for a qualitative semi-structured interview guide. *Journal of Advanced Nursing*, 72(12), 2954–2965. <https://doi.org/10.1111/jan.13031>
 - [12] Kari, R., Gaspar, H. M., Gausdal, A. H., & Morshedi, M. (2018). Human interactions framework for remote ship operations. *2018 26th Mediterranean Conference on Control and Automation (MED)*, 581–588. <https://doi.org/10.1109/MED.2018.8442755>
 - [13] Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1), 50–80. https://doi.org/10.1518/hfes.46.1.50_30392
 - [14] Man, Y., Lundh, M., Porathe, T., & MacKinnon, S. (2015). From desk to field: Human factor issues in remote monitoring and controlling of autonomous unmanned vessels. *Procedia Manufacturing*, 3, 2674–2681. <https://doi.org/10.1016/j.promfg.2015.07.635>
 - [15] Man, Y., Weber, R., Cimbritz, J., Lundh, M., & MacKinnon, S. N. (2018). Human factor issues during remote ship monitoring tasks: An ecological lesson for system design in a distributed context. *International Journal of Industrial Ergonomics*, 68, 231–244. <https://doi.org/10.1016/j.ergon.2018.08.005>
 - [16] Prison, J., Dahlman, J., & Lundh, M. (2013). Ship sense: Striving for harmony in ship manoeuvring. *WMU Journal of Maritime Affairs*, 12, 115–127. <https://doi.org/10.1007/s13437-013-0038-6>
 - [17] Sharma, A., Praetorius, G., Weber, R., MacKinnon, S. N., & Sætrevik, B. (2025). From ship to shore: Understanding cognitive challenges in remote pilotage operations. *Advances in Human Factors of Transportation*, 186, 601–610. <https://doi.org/10.54941/ahfe1006550>
 - [18] Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control*. MIT Press.
 - [19] Stappers, P. J., & Giaccardi, E. (2017). Research through design. In M. Soegaard & R. Friis-Dam (Eds.), *The encyclopedia of human-computer interaction* (2nd ed., pp. 1–94). The Interaction Design Foundation.

- [20] Veitch, E., Alsos, O. A., Cheng, T., Senderud, K., & Utne, I. B. (2024). Human factor influences on supervisory control of remotely operated and autonomous vessels. *Ocean Engineering*, 299, Article 117257. <https://doi.org/10.1016/j.oceaneng.2024.117257>
- [21] Zimmerman, J., Forlizzi, J., & Evenson, S. (2007). Research through design as a method for interaction design research in HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 493–502). ACM. <https://doi.org/10.1145/1240624.1240704>

A

Appendix 1 - Interview and Stakeholder Input Material

A.1 Semi-Structured Interview Guide

This interview guide was used before the participant-driven scenario reconstruction. The purpose was to understand the participant's current onboard work practices, key information needs, uncertainty-handling strategies, and expectations for a remote operation interface.

Table A.1: Semi-structured interview guide, Part 1.

Section	Interview questions
A. Background	What is your role and what are your main responsibilities? What are the usual stages of a normal voyage?
B. Routine monitoring	What information do you mainly monitor during normal operation? How do you judge that everything is normal?
C. Departure readiness	What must you confirm before departure? What information would make you uncomfortable to depart if it was missing?
D. Docking / undocking	During docking or undocking, what takes most of your attention? Which information is most critical during this stage?
E. Onboard vs. remote	On a real bridge, what can you confirm that would be difficult for a remote system to replace? What can cameras replace, and what can they not replace?

Table A.2: Semi-structured interview guide, Part 2.

Section	Interview questions
F. Transition to uncertainty	Have you experienced situations that seemed normal at first but gradually became uncertain or unsafe? What were the earliest cues? How did you decide whether to intervene?
G. Concrete episode	Please recall one concrete example involving a critical or uncertain departure, docking, or monitoring process. What did you notice first? What information did you check? How did you confirm the situation? What did you finally do?
H. Prototype evaluation and interface feedback	Can this interface support your judgement of whether the vessel is ready? What is missing from the interface? Which information should be made more prominent? How should the interface be changed for a multi-vessel scenario?
I. Closing	If only one module could be added, what would you add? Which judgement must remain with the operator?

A.2 Participant-Driven Scenario Reconstruction Template

This appendix presents the template used to guide the participant-driven scenario reconstruction. The template was designed to help participants reconstruct a real onboard bridge scenario and then reflect on how the same situation might change in a Remote Operation Centre (ROC) context.

Table A.3: Participant-driven scenario reconstruction template.

Section	Prompt / item
Participant information	Participant ID Role Years of experience Vessel type / operational background
Part A: Real bridge scenario reconstruction	Daily work routine introduction Basic confidence in the reconstructed scenario Scenario description Decision timeline

Section	Prompt / item
T0: Situation begins	What was happening? What was the first information noticed? Which sensory cues were involved?
T1: Risk recognition	What triggered the awareness of risk? Which instrument or information source was critical? What was the initial interpretation of the situation?
T2: Risk confirmation	How was the risk verified? Was any cross-check performed? For example: radar, AIS, visual observation, team communication, or other sources. Confidence level: 1 / 2 / 3 / 4 / 5
T3: Decision point	What decision was made? What information was used to predict the next 5–10 minutes? Was there any uncertainty in the decision?
T4: Action and outcome	What action was taken? How was the correctness of the action confirmed? Was ship sense important in this situation? If yes, why?
Part B: Remote projection	The same scenario was discussed again, but this time in a remote operation context.
T0: Remote context	Where would the participant look first on the ROC interface? What information would be missing compared with being onboard?
T1: Remote risk recognition	Would the participant recognise the risk at the same time? Yes / No If recognition would be delayed, what would cause the delay?
T2: Remote risk confirmation	Would the participant need additional confirmation? Which interface element would increase confidence? Which interface element would cause hesitation?
T3: Remote decision	Would the participant make the same decision? Yes / No If the decision would be different, why? Trust in the system: 1 / 2 / 3 / 4 / 5
T4: Remote action and monitoring	Would the participant feel fully in control?

Section	Prompt / item
	What additional support would help?

Table A.4: Cognitive gap summary template.

Stage	Onboard strength	Remote limitation	Design opportunity
T0: Situation begins			
T1: Risk recogni- tion			
T2: Risk confirma- tion			
T3: Decision point			
T4: Action and outcome			

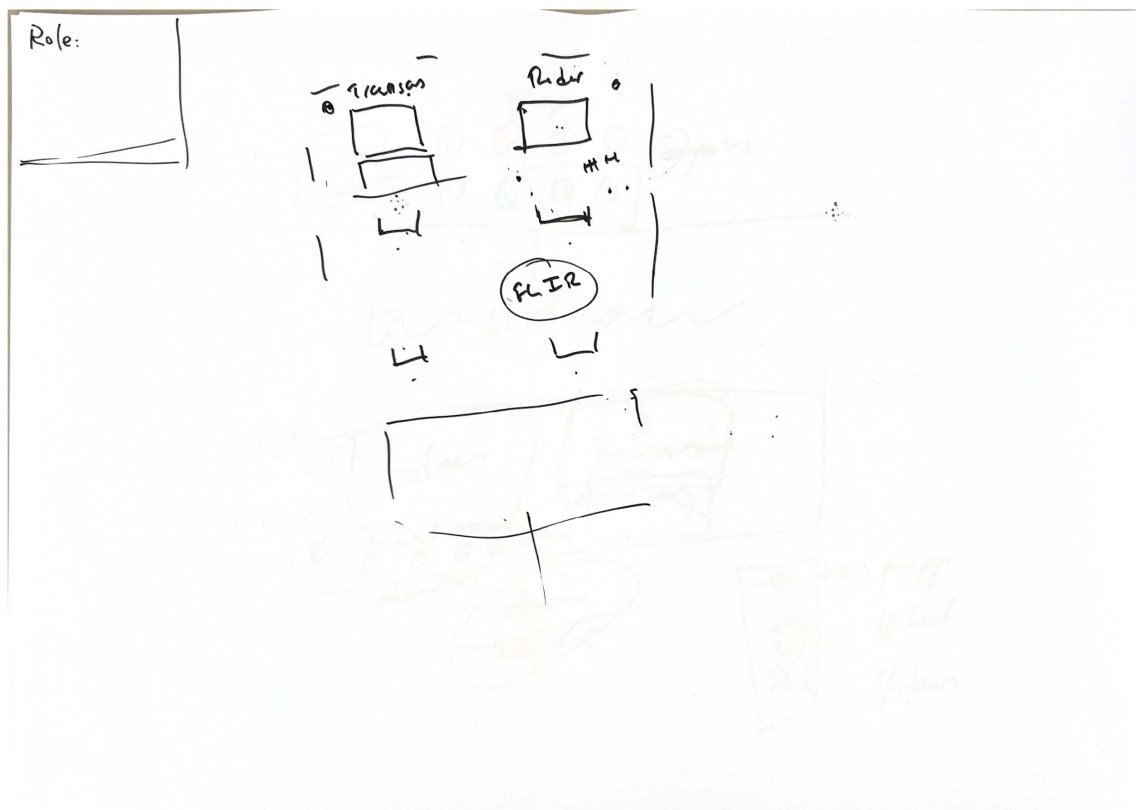
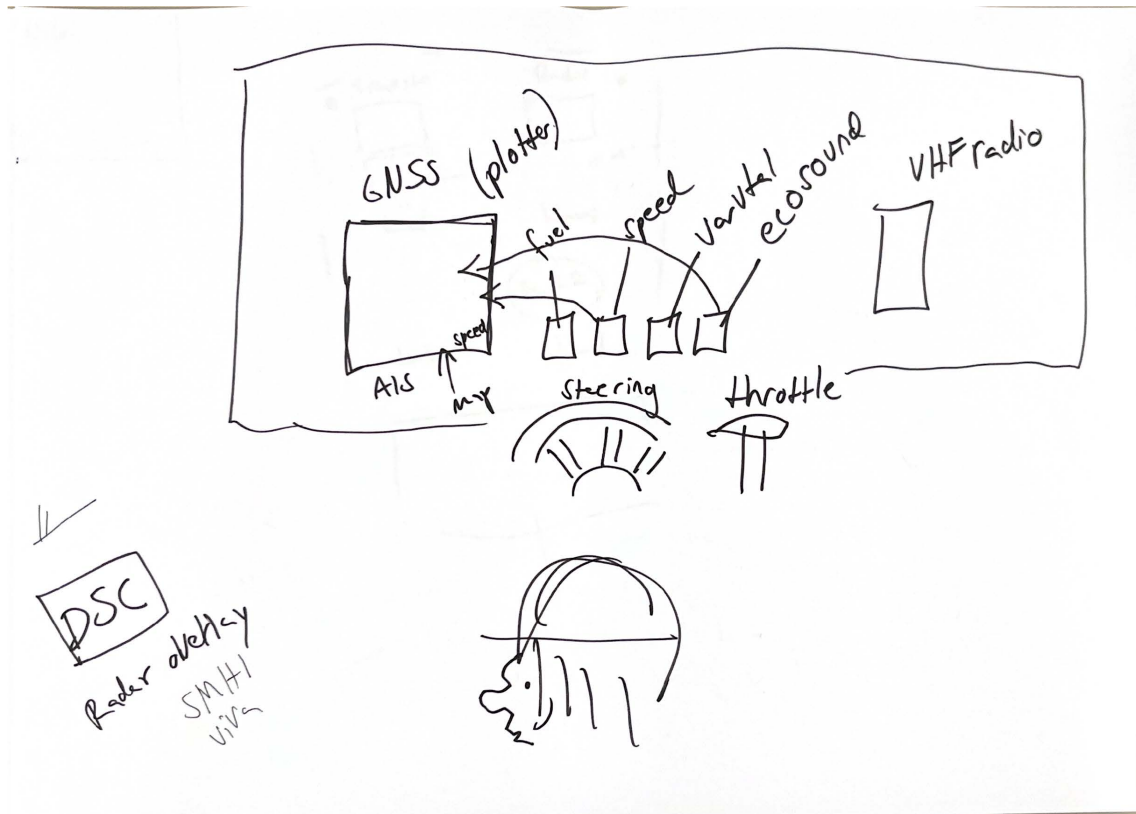
A.2.1 Researcher Notes

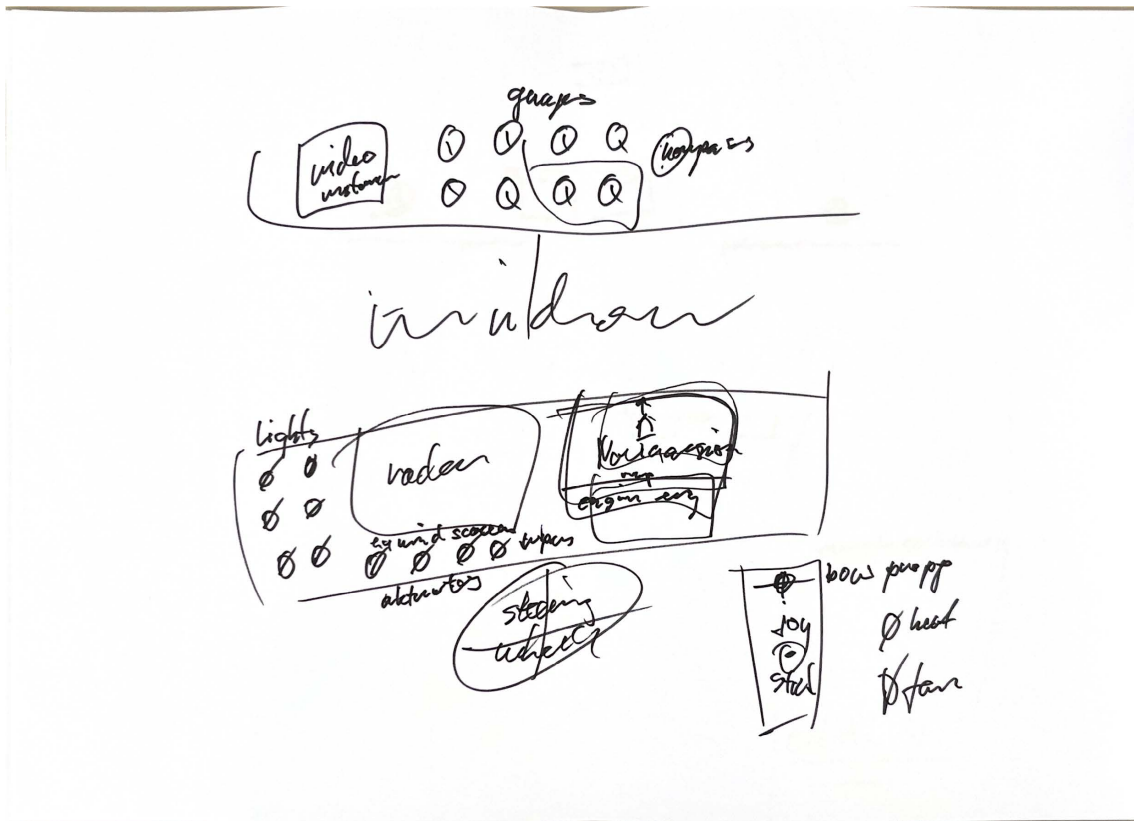
During the scenario reconstruction, the researcher recorded additional notes under the following categories:

- Hesitation moments
- Information search patterns
- Strong emotional reactions
- Notable quotes

A.3 Sketch from Participants

The following participant-generated bridge sketch was collected during the interview to support the reconstruction of onboard information use, spatial attention, and decision-making practices.





A.4 Researcher Reflection on Interview Limitations

The following reflection summarises limitations identified after preparing and conducting the interview. These limitations were not treated as failures of the interview, but as methodological boundaries that shaped how the findings were interpreted. They also helped clarify which topics required more direct prompting during the prototype evaluation and scenario reconstruction.

Table A.5: Researcher reflection on interview limitations.

Limitation	Issue identified	Implication for the study
Limited experience with the proposed ROC task	Participants had not directly worked in the exact task context proposed in this thesis, such as one captain or operator supervising multiple vessels, switching between on-demand service and a shuttle mode, or working mainly through exception-based supervision.	Their answers were grounded in existing onboard practice and professional judgement, rather than direct experience with the proposed ROC workflow. Therefore, the findings were interpreted as expert-informed expectations, not as validation of real ROC work performance.
Unclear system boundaries	Participants were not always forced to define operational boundaries, such as who should trigger mode switching, how responsibility changes after switching, when a situation must be escalated to the shuttle mode, and when the system can return to on-demand mode.	These questions required more explicit prompting during the interview. The limitation also showed that system boundaries, responsibility transfer, and escalation logic should be made clearer in the interface concept and future evaluation.
Limited discussion of organisational and service-level issues	Participants were more likely to discuss vessel handling, safety judgement, and onboard information use than broader organisational issues, such as shift work, staffing, role allocation, and service management.	The interview findings mainly supported interface-level and task-level design decisions. Organisational, institutional, and service-operation questions should be treated as limitations and possible future work, rather than fully resolved design requirements.

B

Appendix 2 - Data Analysis and Theme Development

Appendix B presents the data analysis process used to translate interview material into design implications. The appendix includes the initial coding schema, the cross-participant coding matrix, and the revised coding schema used to inform the final ROC interface concept.

B.1 Data Analysis Procedure

The interview data were analysed through an iterative coding process. The analysis started with open coding of each interview, where recurring issues, decision cues, information needs, and concerns about remote operation were extracted from the interview notes and scenario reconstruction material. The initial codes were then compared across participants to identify shared patterns and participant-specific insights.

The analysis was not intended to produce statistical results. Instead, it was used to support design synthesis by identifying repeated operational concerns and translating them into interface requirements. A participant-by-code matrix was used to check whether each code was strongly, moderately, weakly, or not clearly present across the interviews. This helped prevent the analysis from relying only on memorable individual comments, while still retaining critical issues that appeared strongly in only one or two interviews.

The coding schema was revised after the first round of synthesis. The initial schema mainly focused on cognitive and supervisory issues, such as operational picture, decision authority, ship sense, workload, and automation trust. However, the matrix comparison showed that several operational and service-level issues were under-represented. These included vessel condition, alerting, communication, workflow sequencing, weather and water-condition awareness, and service legitimacy. The final coding schema therefore combines supervisory cognition with broader socio-technical and operational dimensions.

B.2 Initial Coding Schema

Table B.1: Initial coding schema developed from the interview material.

Initial code category	Description
Operational picture and situation construction	How participants described the information needed to understand the vessel, route, surrounding traffic, and operational situation. This included trusted information sources such as radar, AIS, camera views, local knowledge, and vessel-state information.
Decision, authority, and release control	How participants described safety-related decisions, especially around departure readiness, docking, undocking, and the boundary between automatic operation and human judgement.
Embodied and situated knowledge	How participants described onboard cues that are difficult to replace remotely, including movement, sound, weather exposure, visual judgement, and tacit ship sense.
Workload, attention, and fleet supervision	How participants described the challenge of supervising more than one vessel, switching attention between vessels, and avoiding information overload or missed events.
Automation trust, failure, and safe-state logic	How participants described trust in automation, uncertainty about system correctness, failure handling, and the need for safe fallback states.
Passenger, emergency, and service implications	How participants described passenger safety, emergency handling, service continuity, and public acceptability of remote or autonomous ferry operation.

B.3 Cross-Participant Coding Matrix

The following matrix was used to compare how strongly each code appeared across the interview material. The matrix was used as an analytical aid, not as a quantitative measurement.

Table B.2: Cross-participant coding matrix.

Code	P1	P2	P3	P4	Interpretation
Trusted and integrated operational picture	H	H	H	H	Strong shared theme. The interface needs to integrate several sources into one usable operational picture.

Continued on next page

Code	P1	P2	P3	P4	Interpretation
Departure readiness as a human decision boundary	H	M	H	M	Stable theme. Departure was treated as a critical decision point where missing information reduces confidence.
Docking and undocking attention demand	H	H	M	M	Strong theme. Close manoeuvring requires clear spatial, visual, and vessel-state information.
Loss of ship sense in remote operation	H	M	H	M	Stable theme. Remote systems cannot fully replace embodied onboard cues.
Camera as partial visual replacement	M	H	M	H	Cameras were useful, but not sufficient alone. They support confirmation rather than complete situation understanding.
Automation trust and verification burden	H	H	M	H	Strong theme. Operators need to understand whether the system is correct, uncertain, or failing.
Workload and multi-vessel supervision	M	M	H	M	Important theme. Multi-vessel supervision requires prioritisation and alert management.
Vessel condition and technical readiness	M	H	M	M	Added in the revised schema because technical state was relevant to readiness and trust.
Communication and coordination	M	M	H	M	Added in the revised schema because remote operation depends on coordination with onboard crew, passengers, and other actors.
Weather and water-condition awareness	H	M	M	M	Added in the revised schema because environmental judgement affects confidence and operational safety.
Service legitimacy and passenger acceptability	L	M	M	H	Added in the revised schema because the system must be acceptable as a public transport service, not only as a vessel-control system.

H = strongly present; M = moderately present; L = weakly present; – = not clearly present.

B.4 Revised Coding Schema and Design Translation

After the cross-participant comparison, the coding schema was revised to better reflect both cognitive-supervisory issues and operational/service realities. The revised schema was then used to translate interview findings into design implications for the ROC interface.

Table B.3: Revised coding schema and design translation.

Revised code category	Analytical meaning	Design implication
Operational picture and situation construction	Operators need a coherent picture of vessel state, route, traffic, weather, cameras, and system interpretation.	The interface should integrate map, vessel state, sensor evidence, route context, and alert reason instead of showing fragmented data panels.
Decision, authority, and release control	Some decisions mark a boundary between routine automation and human responsibility, especially departure, docking, route restriction, and escalation.	The interface should clearly separate routine autonomous events from supervisory commands that require human acknowledgement, veto, or escalation.
Embodied and situated knowledge	Onboard judgement depends on physical and sensory cues that are only partly available remotely.	The interface should provide compensatory evidence, such as camera views, vessel motion indicators, environmental data, and confidence cues, while acknowledging that cameras cannot fully replace ship sense.
Workload, attention, and fleet supervision	Supervising multiple vessels creates attention-switching and prioritisation problems.	The interface should support fleet-level prioritisation, alert hierarchy, selected-vessel focus, and a clear separation between monitoring screens and command workspace.
Automation trust, failure, and safe-state logic	Operators need to know whether automation is acting correctly, uncertainly, or failing.	The interface should show automation mode, system confidence, reason for recommendation, fallback state, and required operator response.

Continued on next page

Revised code category	Analytical meaning	Design implication
Vessel condition and technical readiness	Readiness is not only a route or passenger question; it also depends on propulsion, battery, sensors, connection, doors, and other vessel-state information.	The interface should include technical readiness indicators and make missing or degraded data visible before departure or critical operation.
Alerting and attention management	Alerts should guide attention without turning every routine event into a command.	The interface should distinguish information, caution, warning, and critical states, and should reserve command queue items for situations that require operator judgement.
Communication and coordination	Remote operation depends on communication with onboard crew, passengers, and possibly external actors.	The interface should include communication access and show who needs to be contacted, what has been communicated, and whether confirmation has been received.
Operational workflow and task sequencing	Operators need to understand where the vessel is in the operational sequence, such as departure, transit, docking, passenger boarding, or service recovery.	The interface should show voyage phase, current task state, next expected action, and whether the system is waiting, acting, or requesting human input.
Weather and water-condition awareness	Environmental cues influence confidence, manoeuvring judgement, and interpretation of sensor data.	The interface should show wind, current, visibility, water conditions, and their operational relevance instead of displaying them as isolated weather data.
Service legitimacy, passenger, and emergency management	The ROC system must work as part of a public ferry service, not only as a technical control system.	The interface should support passenger-related awareness, service disruption handling, emergency escalation, and transparent operational decisions.

B.5 Transition from Codes to Findings Themes

The revised coding schema was used as an intermediate analytical structure between the interview material and the findings presented in Chapter 4. The final findings themes were not generated by counting code frequency alone. Instead, they were developed by grouping related codes according to their relevance to the thesis aim:

understanding what kind of information, judgement, and interaction support a ROC interface should provide for autonomous urban ferry supervision.

The coding categories were therefore consolidated into four broader themes for Chapter 4. These themes summarise the most relevant patterns across participants while also preserving operational issues that were important for the design concept, even when they were not equally strong in every interview.

Table B.4: Mapping from revised codes to Chapter 4 findings themes.

Chapter 4 findings theme	Related revised codes	Reason for grouping
Theme 1: Remote supervision requires integrated situation reconstruction	Operational picture and situation construction; embodied and situated knowledge; vessel condition and technical readiness; weather and water-condition awareness.	These codes describe how operators reconstruct the situation remotely by combining vessel state, route context, environmental cues, camera views, and experience-based judgement.
Theme 2: Fleet operation requires dynamic priority management	Workload, attention, and fleet supervision; alerting and attention management; operational workflow and task sequencing; communication and coordination.	These codes describe the challenge of supervising several vessels at once and deciding which vessel, event, or task should receive attention first.
Theme 3: Automation must be explainable before it becomes actionable	Automation trust, failure, and safe-state logic; operational picture and situation construction; alerting and attention management; vessel condition and technical readiness.	These codes describe the need for the system to explain what it detects, why it recommends an action, how confident it is, and what risk remains before the operator can act on it.
Theme 4: Routine operation needs clear responsibility boundaries	Decision, authority, and release control; operational workflow and task sequencing; communication and coordination; service legitimacy, passenger, and emergency management.	These codes describe the boundary between autonomous routine operation and human responsibility, including departure, docking, escalation, communication, and service-level accountability.

C

Appendix 3 - Design Process and Iterations

This appendix documents the design process and interface iterations that led to the final ROC prototype. The project was developed from an earlier course project, where two previous interface concepts were created for different operational modes. During the thesis, these earlier outputs were reviewed and revised according to the final thesis scope: monitoring, communication, and supervisory command for autonomous urban ferry operation.

The materials in this appendix are included to show how the final prototype evolved through previous versions, wireframes, early visual concept exploration, and later screen-level iterations.

C.1 Previous Project Version

The thesis continued from a previous project course, where two interface directions were developed for different operational modes. These earlier versions provided an initial visual and functional basis, but they were not directly used as the final thesis prototype. Instead, they were reviewed to identify which elements could support the revised thesis scope and which parts needed to be removed, simplified, or restructured.

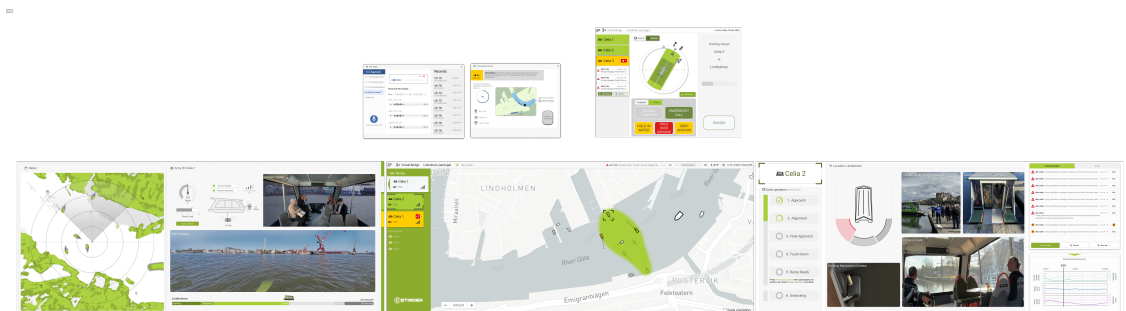


Figure C.1: Previous version 1

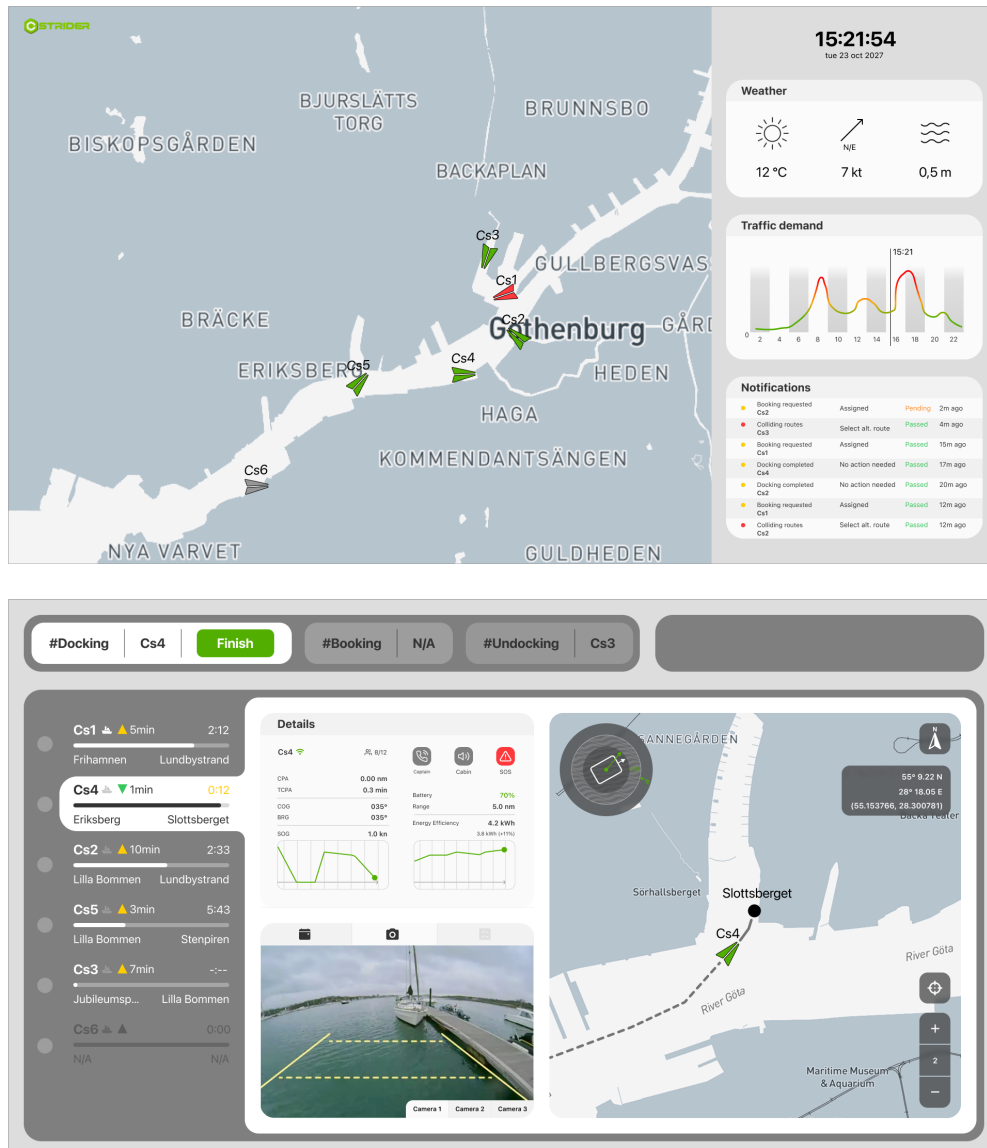


Figure C.2: Previous version 2

C.2 Wireframes

Wireframes were used to explore the basic structure of the ROC interface before visual styling and detailed components were developed. The focus at this stage was screen responsibility, information placement, and the relationship between fleet monitoring, command handling, and single-vessel monitoring.



Figure C.3: Wireframes

C.3 Early Concept Exploration

Early visual concept images were generated to explore possible workstation atmosphere, screen arrangement, and interface direction. These images were used as low-fidelity visual exploration material rather than final prototype screens.

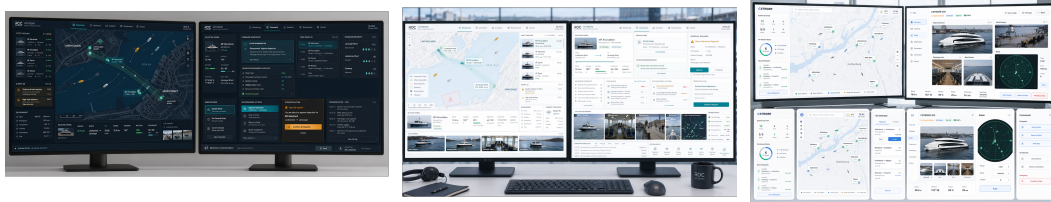


Figure C.4: AI-assisted visual exploration

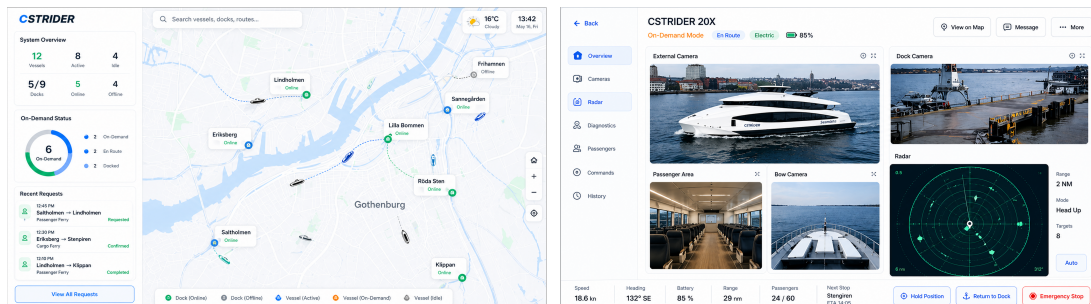
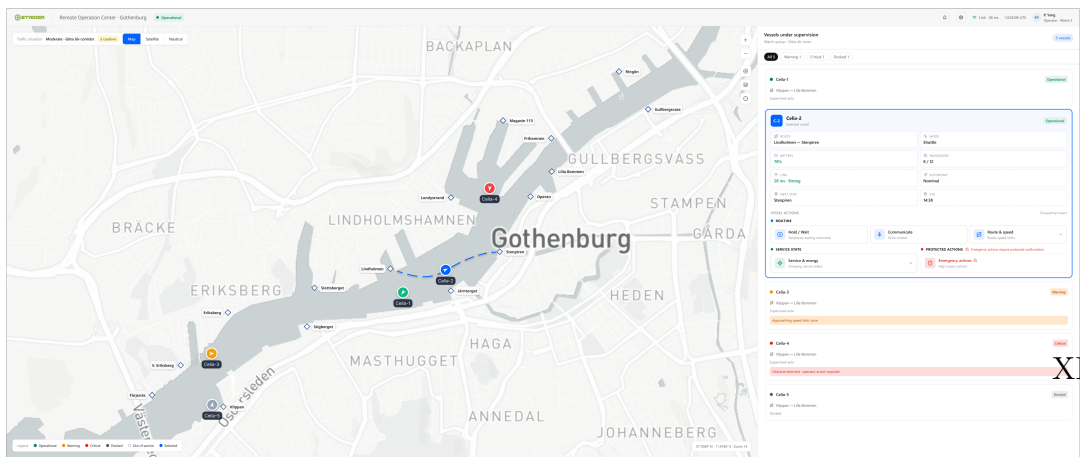
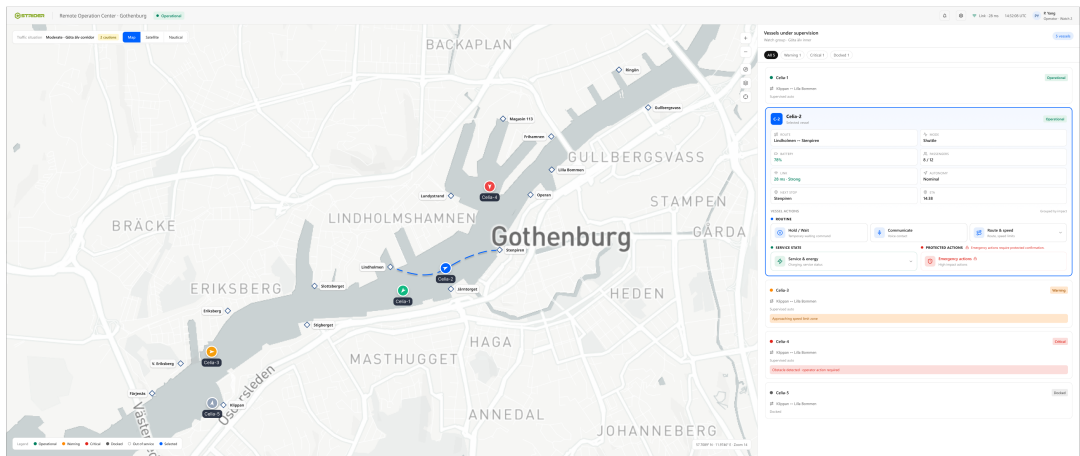
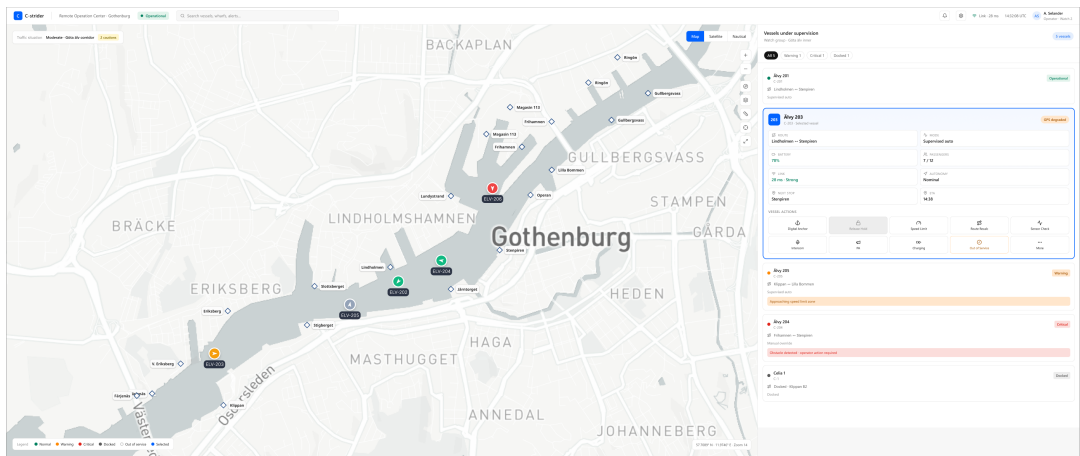
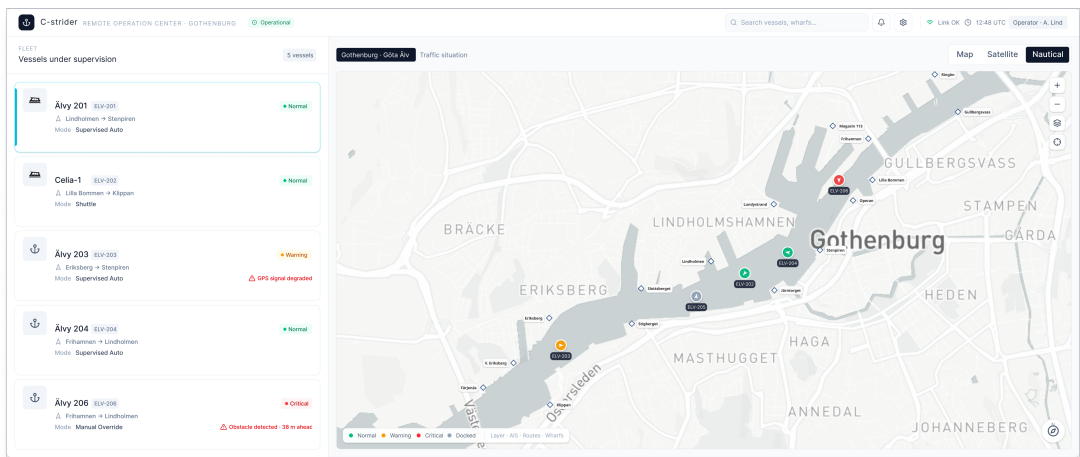


Figure C.5: AI-assisted final concept

C.4 Prototype Screen Iteration

The later prototype iterations focused on screen-level refinement. The main design work was to clarify the responsibility of each screen and to reduce ambiguity between monitoring, evidence review, and supervisory command. The left screen was developed as the fleet monitoring and spatial situation awareness screen. The right screen was developed as the command workspace for system-initiated exceptions and operator decisions. The vessel monitoring screens were kept as supporting monitoring displays rather than command screens.

C. Appendix 3 - Design Process and Iterations



C. Appendix 3 - Design Process and Iterations

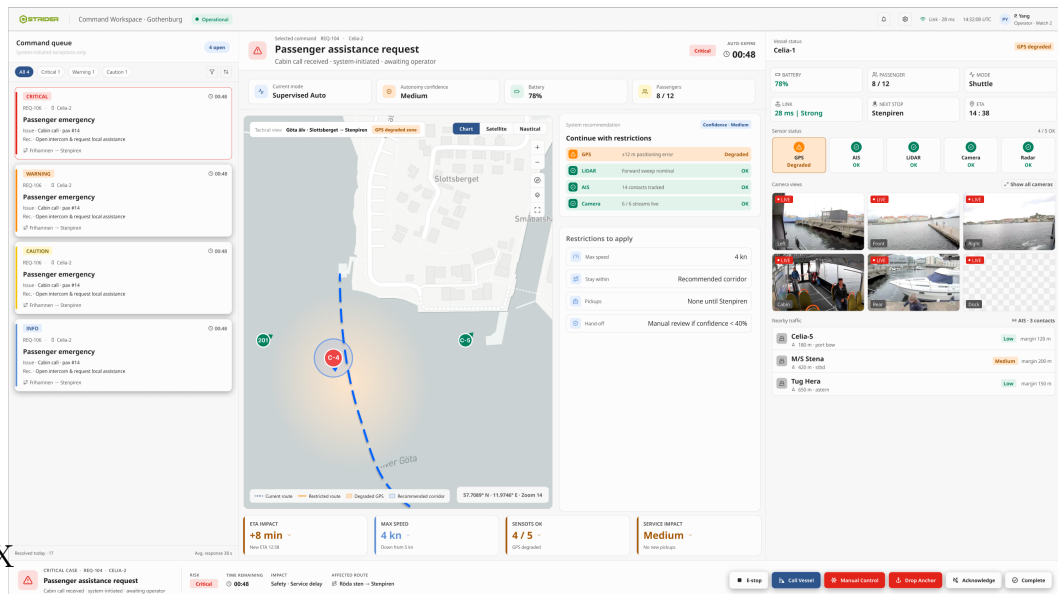
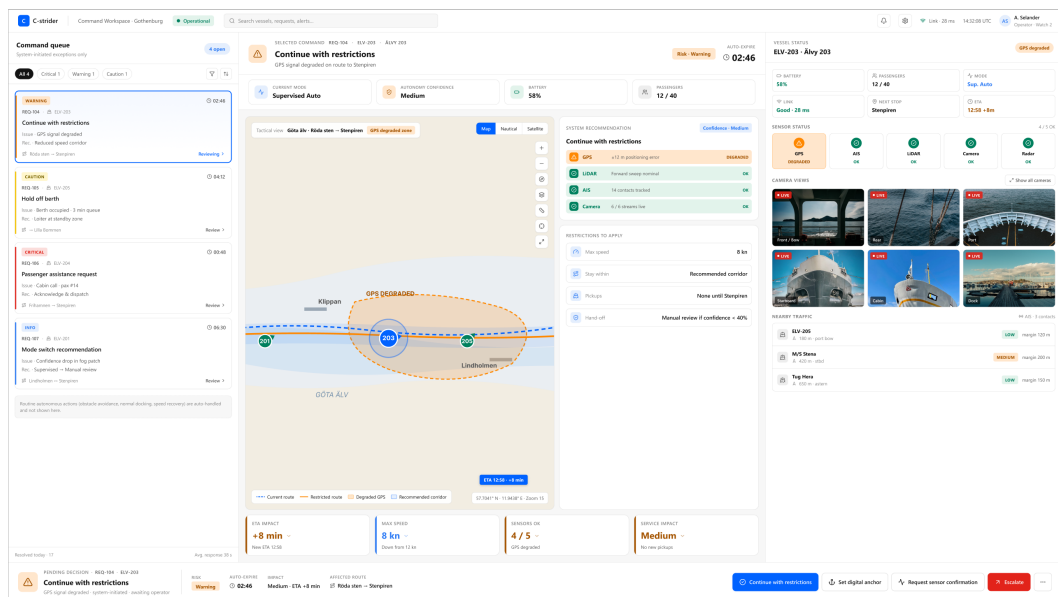
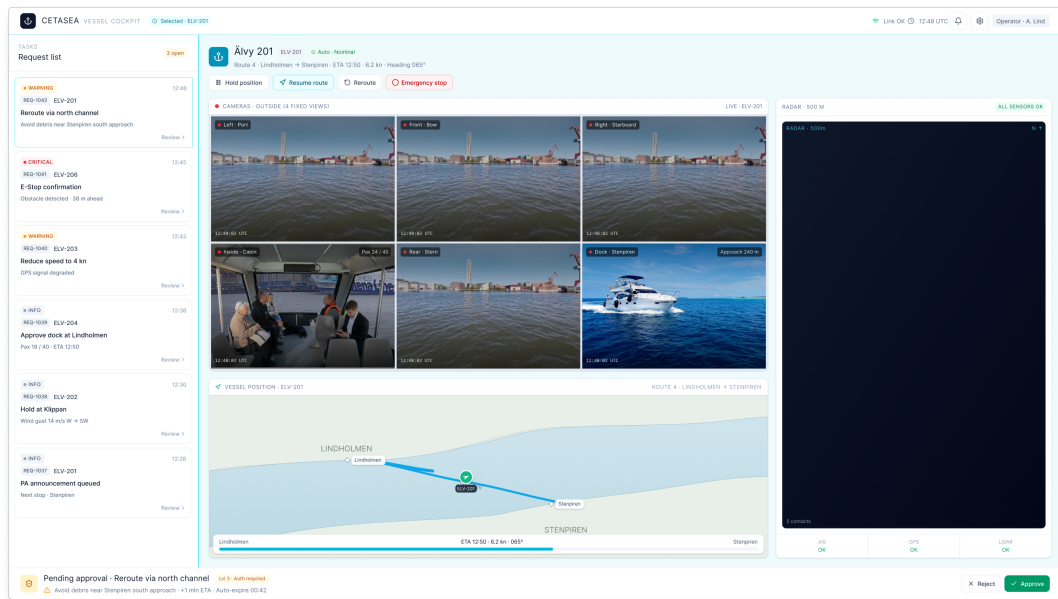


Figure C.7: Right screen iteration

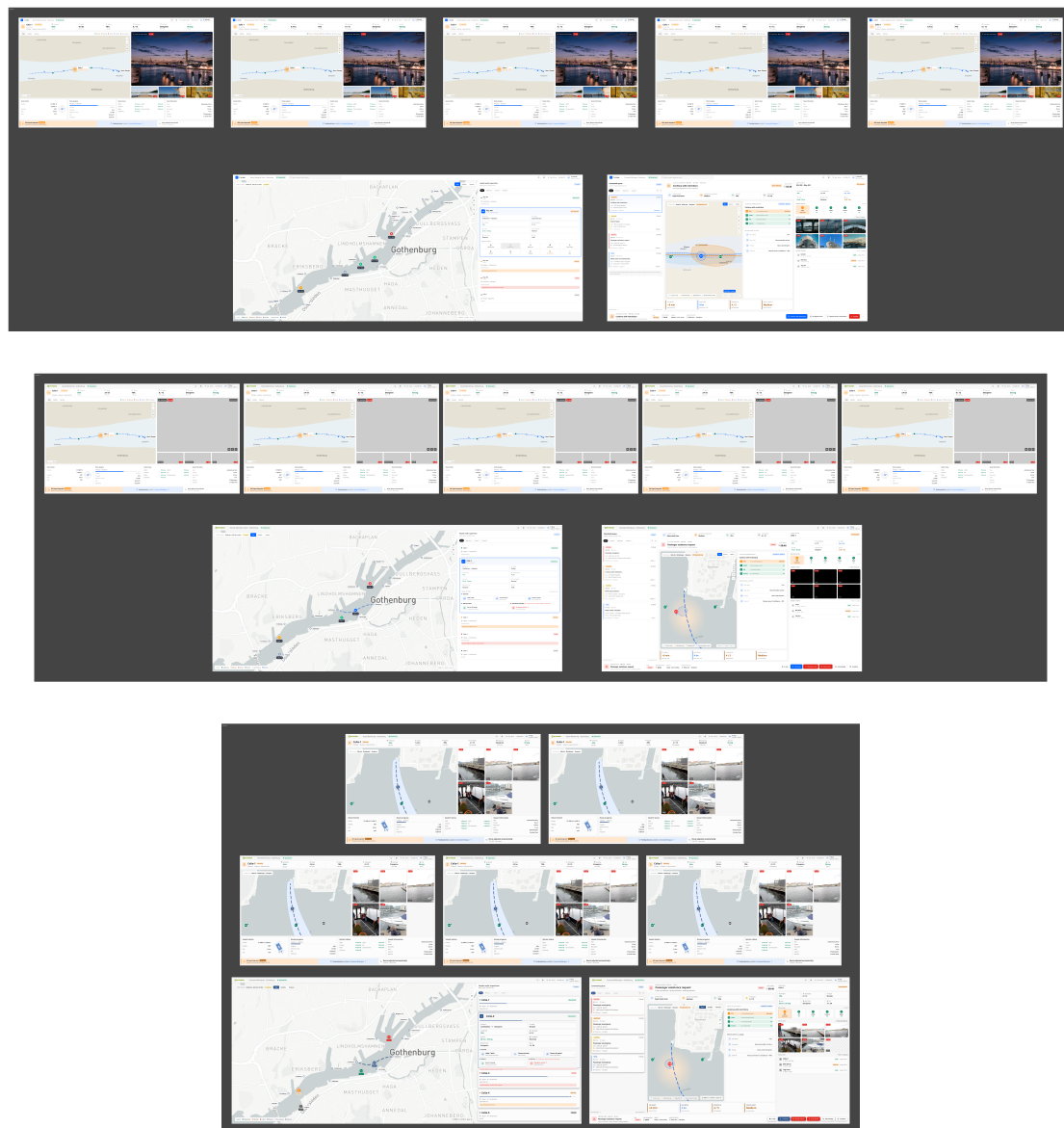


Figure C.8: Screen layout iteration

D

Appendix 4 - Information Architecture and Command Logic

This appendix presents the information architecture and supervisory command logic used in the final ROC interface concept. The purpose is to clarify how information, screens, alert levels, event types, and operator commands are structured in the prototype. This appendix supports the design rationale in Chapter 6 by making the boundary between routine automation and human supervisory command explicit.

The final interface does not treat the ROC operator as a continuous remote driver. Instead, the interface is structured around fleet monitoring, situation understanding, exception handling, communication, and high-level supervisory command.

D.1 Information Architecture Overview

The information architecture was organised around three main operational layers: fleet-level supervision, command-level decision-making, and single-vessel monitoring. This separation was used to avoid placing all information and actions into one general dashboard.

Table D.1: Information architecture of the final ROC interface prototype.

IA layer	Where it appears in prototype	Main information	Operator question answered
Fleet overview	Left screen	Fleet map, vessel positions, route lines, wharfs, service mode, vessel list, and selected vessel card.	Which vessel or area needs attention?
Selected vessel context	Left screen, right screen, and vessel monitoring screen	Vessel mode, route, ETA, passenger count, battery, connection, autonomy state, and active alert.	What is the current state of this vessel?
Command queue	Right screen	Pending cases, alert level, affected vessel, short reason, and time sensitivity.	What requires my decision now?
Command workspace	Right screen	Selected case, system recommendation, rationale, risk, impact, and expected outcome.	What happened, why does it matter, and what should I do?
Evidence layer	Right screen and vessel monitoring screen	Camera views, sensor status, map context, traffic context, weather or water condition, and confidence indicators.	What evidence supports the system's recommendation?
Response layer	Right screen bottom bar	Available commands, acknowledge, hold, continue with restrictions, contact, escalate, and emergency action.	What supervisory action can I take?
Single-vessel monitoring	24-inch vessel monitoring screens	Local route progress, vessel state, camera feeds, passenger status, system health, and recent events.	How is this specific vessel operating?
Communication layer	Right screen and selected vessel panels	Intercom, PA, crew contact, passenger assistance, emergency contact, and communication status.	Who needs to be contacted or informed?
Event and audit trail	Logs and event panels	Automatic events, acknowledged alerts, completed commands, and operator actions.	What has happened and what has been done?

D.2 Command Scope and Logic

D.2.1 Command Scope

The final prototype defines the ROC operator's work as a combination of monitoring, communication, and supervisory command. The operator is not treated as a continuous remote driver. Instead, the interface supports the operator in understanding the system state, communicating with relevant actors, and issuing high-level commands when routine autonomy cannot close the loop.

In this thesis, command refers to supervisory action rather than direct vessel control. Supervisory commands include actions such as holding departure, continuing with restrictions, requesting confirmation, contacting the vessel, escalating a case, or triggering a protected safety action. Direct manual control, such as continuous steering, throttle control, or joystick-based docking, is treated as outside the main prototype scope and is left for future work.

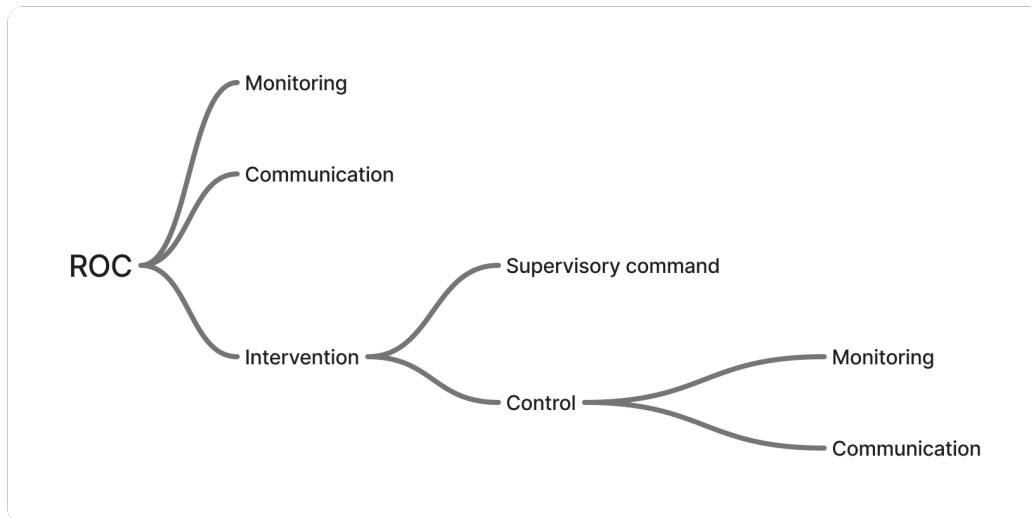


Figure D.1: Command scope in the ROC interface

As shown in Figure D.1, supervisory command is positioned as one part of the broader ROC action scope. This distinction was important for the prototype because it prevented the interface from becoming a full teleoperation cockpit. Routine autonomous actions remain system-handled, while the operator intervenes when a case requires human judgement, communication, responsibility transfer, or safety escalation.

D.2.2 Command Level Logic

The command level logic defines how system events are translated into operator attention and response. The levels are not only severity labels. They also define the expected operator role, the type of response button, and whether the event should be treated as a command case.

In this prototype, not every system event becomes a command. Information-level events are recorded as status updates. Caution-level events are mainly monitored. Warning-level events require operator judgement. Critical-level events require immediate safety response and may involve protected commands.

Table D.2: Command level logic

Level	Core meaning	Operator role	Typical event types	Response logic
Critical	Act now	Immediate safety response	Passenger emergency, imminent collision, loss of control, communication critical, localization lost near risk area, fire, smoke, flooding, or corridor emergency.	Use protected actions such as emergency stop, contact vessel, manual control escalation, drop anchor, hold all vessels, or complete the case after safe state is confirmed.
Warning	Decide soon	Confirm strategy or choose alternative	GPS degraded, exceptional route change, sensor conflict, speed restriction, hold-off berth, mode review, low battery affecting service, take-vessel-out-of-service recommendation, or exceptional fleet reassignment.	Select a supervisory command, such as continue with restrictions, hold standby, request sensor confirmation, approve or reject route change, request alternative, acknowledge, or complete.
Caution	Watch	Monitor and lightly adjust	Berth queue, minor delay, auto-loiter scheduled, mild weather or wake condition, sensor quality advisory, charging advisory, low-impact mode schedule, temporary caution zone nearby, or passenger count notice without mismatch.	System continues automatically. The operator may keep the auto plan, monitor, request status update, notify, acknowledge, or close.

Continued on next page

Level	Core meaning	Operator role	Typical event types	Response logic
Info	Know	Awareness only	Departure request received, departure completed, docking initiated, docking completed, route adjusted automatically, route resumed automatically, mode switch scheduled or completed, charging scheduled, diagnostic completed, passenger count updated, or sensor restored.	No command decision is required. The operator may track event, acknowledge, or close.

The four levels also define the boundary between routine automation and supervisory command. Info and caution events are normally handled by the system and shown as status or advisory information. Warning events become command cases when the system requires human judgement, confirmation, or an alternative strategy. Critical events become protected command cases because they involve immediate safety risk or emergency response.

E

Appendix 5 - Visual Design System Support

This appendix summarises the visual design system decisions used in the final ROC interface prototype. The purpose is not to provide a complete design system specification, but to document the main visual rules that supported consistency, readability, and operational credibility.

The prototype used an OpenBridge 6.1-inspired maritime HMI visual language. The visual design decisions focused on semantic colour use, clear typography, consistent component states, recognisable icon usage, and stable layout patterns across the multi-screen workstation.

E.1 Colour and Status System

Colour was used semantically in the final prototype rather than as decoration. The main purpose of colour was to help the operator distinguish between normal operation, selected states, information events, caution states, warning states, critical cases, and unavailable actions.

The palette was organised into base interface colours and operational status colours. Base colours were used for panels, buttons, strokes, text, disabled states, and active elements. Status colours were used to separate different levels of operational urgency, including information, caution, warning, critical, and running states.



Figure E.1: Colour palette used in the final prototype, including base interface colours and semantic status colours for information, caution, warning, critical, and running states.

Table E.1: Colour and status system used in the final prototype.

Colour group	Main tokens	Application in prototype
Base interface colours	Brand #8BAC2D; button #FCFCFC; background #F7F7F7; stroke #DDDDDD; neutral #535353; active #1F1F1F; disabled #BEBEBE.	Used for the general interface structure, including panels, button surfaces, borders, text hierarchy, active states, and unavailable actions.
Critical	Background-critical #FBDDDB; element-critical #E2231A.	Used for safety-critical cases such as passenger emergency, collision risk, loss of control, communication loss, fire, or other urgent emergency states.
Warning	Background-warning #FFE6CC; notification-warning #FF8800; element-warning #A55400.	Used for cases that require operator review and a supervisory decision, such as sensor conflict, degraded localisation, route change, or service limitation.
Caution	Background-caution #FFF6CC; notification-caution #F5C400; element-caution #7A6500.	Used for low-impact advisory states that should be monitored, such as berth queue, minor delay, mild weather, or sensor quality reduction.
Running / normal operation	Background-running #E2F3EB; element-running #00805E.	Used to indicate active operation, healthy running state, or a vessel/system state that is proceeding within expected limits.
Information	Background-info #E4EEFD; notification-info #5D8FD5; element-info #2D548B.	Used for routine system events, completed actions, scheduled changes, status updates, and other information that does not require operator decision.

E.2 Typography and Information Hierarchy

Typography was used to create a clear hierarchy between screen-level information, object-level information, supporting details, labels, and actions. This was important because the ROC interface contains dense operational information across several screens. A consistent typography system helped the operator scan the interface, identify the current focus, and distinguish primary information from secondary metadata.

The prototype used a limited set of text styles, including title, subtitle, overline, body, button, active body, label, and active label. These styles were applied consistently across vessel cards, command queue items, evidence panels, status labels, and response buttons.

Title		Overline	Active body	Active label
Subtitle		Button	Body	Label
Text style		Text attributes		
Aa	Title	Weight: \$text-weight-l Size: \$text-size-150 Case: none		
Aa	Subtitle	Weight: \$text-weight-s Size: \$text-size-150 Case: none		
Aa	Overline	Weight: \$text-weight-m Size: \$text-size-075 Space: 0.5 Case: capitalize		
Aa	Body	Weight: \$text-weight-s Size: \$text-size-100 Case: none		
Aa	Button	Weight: \$text-weight-m Size: \$text-size-100 Case: none		
Aa Bb	Button-two-line	Weight: \$text-weight-m Size: \$text-size-100 Case: none		
Aa	Body-active	Weight: \$text-weight-l Size: \$text-size-100 Case: none		
Aa	Label	Weight: \$text-weight-s Size: \$text-size-075 Case: none		
Aa	Label-active	Weight: \$text-weight-l Size: \$text-size-075 Case: none		

Figure E.2: Typography styles used in the final prototype, including title, subtitle, overline, body, button, active body, label, and active label text styles.

Table E.2: Typography hierarchy and application in the final prototype.

Text style	Application in prototype	Purpose
Title	Screen title, main panel title, selected case title, and major section heading.	Identifies the main area or object currently being inspected.
Subtitle	Secondary screen title, vessel name, route group, and supporting panel title.	Provides a lower-level heading without competing with the main title.
Overline	Small category labels, section markers, status group labels, and metadata headings.	Supports quick scanning and separates groups of information.
Body	Event descriptions, system explanations, evidence text, and recommendation details.	Communicates the meaning of a situation or system recommendation.
Active body	Highlighted body text, active values, selected information, and important data fields.	Draws attention to information that is currently active or decision-relevant.
Button	Response buttons, command actions, acknowledgement buttons, and navigation controls.	Makes available actions clear and readable.
Button-two-line	Longer response buttons or commands that require two-line wording.	Allows complex supervisory actions to remain readable without reducing button clarity.
Label	Status labels, metadata, time, mode, ETA, confidence, battery, passenger count, and sensor state.	Presents compact supporting information without overwhelming the main content.
Label-active	Selected status labels, active tabs, focused metadata, and highlighted state indicators.	Distinguishes active or selected labels from passive supporting information.

E.3 Components and Interaction States

The prototype reused a set of component patterns across the multi-screen interface. These components were used to keep vessel information, command cases, evidence panels, status labels, and response actions visually consistent. This was important because the ROC operator needs to move between fleet monitoring, command review, and single-vessel monitoring without relearning the interface logic on each screen.

The component set was not developed as a complete standalone design system. Instead, it worked as a local component library for the final prototype. The main focus was to keep status presentation, case handling, evidence review, and command response consistent across different operational situations.

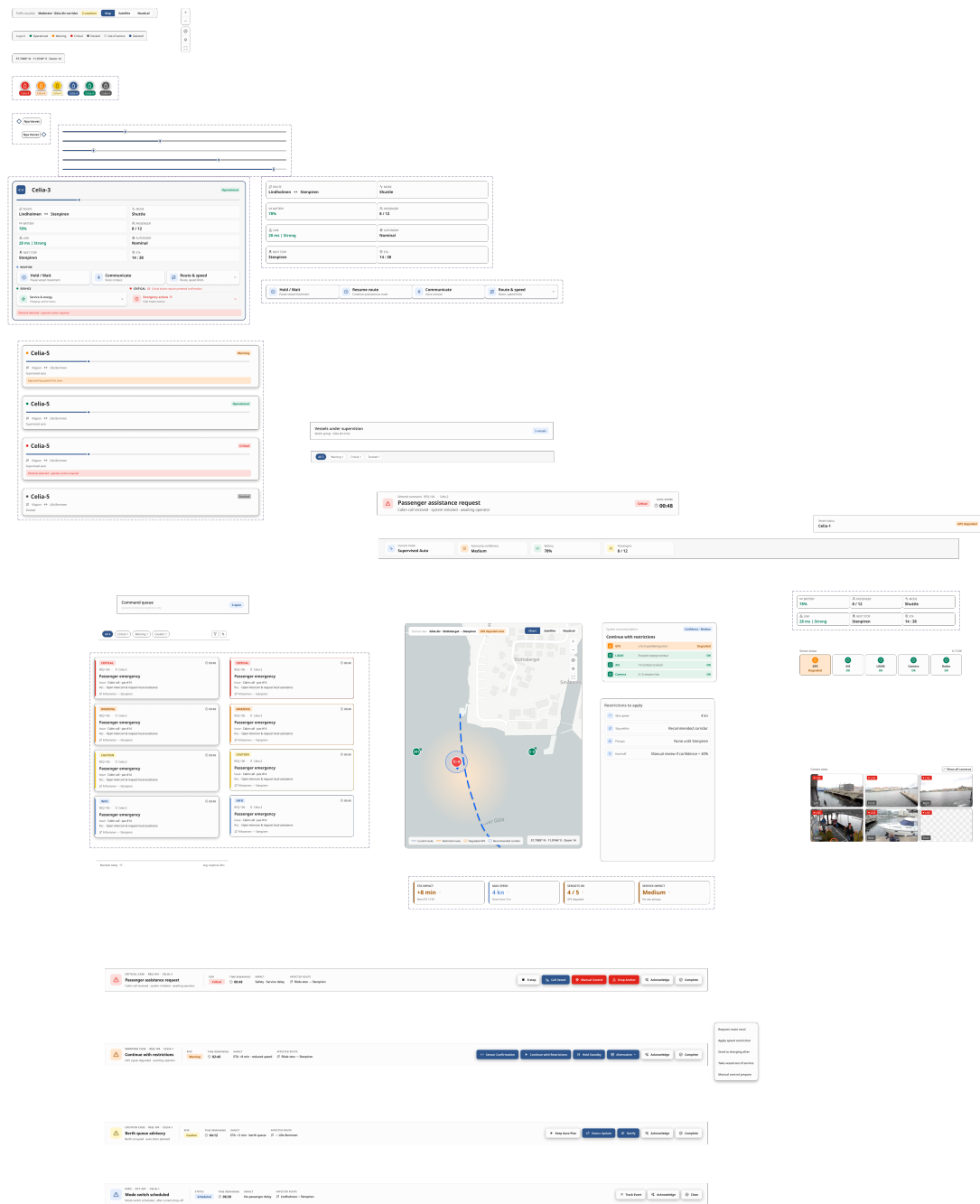


Figure E.3: Selected component examples used in the final prototype, including status labels, vessel cards, command queue items, evidence panels, camera panels, information cards, and response bars.

E.4 Icon Usage

Icons were used to support recognition and reduce visual scanning effort in dense operational screens. The icon set included vessel-related icons, route and navigation

icons, system-state icons, communication icons, and command-related icons. In the final prototype, icons were mainly used together with text labels, especially for safety-related actions and supervisory commands. This was done to reduce ambiguity and avoid relying on icon meaning alone.

 Cstrider-own-ship	 time	 route
 anchor watch	 stop	 radar
 communication-radio	 manual drive	 silence
 passed	 up	 down
 layer	 target select	 bidirectional-hor
 battery-hor	 link	 autonomy
 next stop	 passenger	 mode
 ETA	 speed	 ethernet-google
 media-play	 media-pause	 conning-iec
 chart-update-review	 com-pa	

Figure E.4: Icon set used in the final prototype, including vessel, route, communication, sensor, status, and command-related icons.

F

Appendix 6 - AI Use Documentation

This appendix documents how generative AI was used during the thesis process. It focuses on the types of prompts used in the project and how the outputs were controlled, revised, and verified by the author. The purpose is not to provide a full transcript of AI conversations, but to make the role of AI transparent.

AI tools were used as process support for writing, translation, design reflection, structure checking, prototype critique, and visual communication. They were not used as independent evidence, research participants, or automatic decision-makers. All final report content, design decisions, prototype changes, and evaluation interpretations remained the responsibility of the author.

Table F.1: Summary of prompt use during the thesis process, Part 1.

Prompt use area	Typical prompt purpose	Human control and verification
Report structure and argument development	Prompts were used to review chapter structure, reorganise sections, clarify the thesis argument, reduce repetition, and check whether the report moved logically from background to findings, requirements, design, evaluation, and discussion.	The author selected the final structure, removed repeated content, and checked that each section matched the actual thesis scope and prototype.
Academic writing and language refinement	Prompts were used to draft, rewrite, translate, and simplify report text. They were also used to make English paragraphs clearer, more direct, and more suitable for an academic thesis.	The author reviewed and edited the final wording. AI-generated text was not accepted directly when it did not match the project scope, design outcome, or intended academic tone.
Literature synthesis and design implication support	Prompts were used to summarise literature, compare concepts, and translate human factors topics into design implications. This included situation awareness, remote supervision, ship sense, workload, automation transparency, and supervisory command.	Literature-based claims were checked against actual sources. AI-generated claims were not treated as references unless they could be connected to real literature or project material.
Interview and analysis support	Prompts were used to refine interview questions, scenario reconstruction material, coding categories, coding matrices, and transitions from qualitative findings to design-oriented themes.	The author used the outputs as structuring support. The interpretation of interview material, coding decisions, and final findings remained author-led.
Information architecture and command logic	Prompts were used to define screen responsibilities, information layers, command scope, alert levels, event types, and the boundary between routine automation and supervisory command.	The author checked whether the proposed logic matched the actual prototype and the thesis scope. Full manual teleoperation and unsupported functions were removed or treated as future work.

Table F.2: Summary of prompt use during the thesis process, Part 2.

Prompt use area	Typical prompt purpose	Human control and verification
Prototype critique and screen layout refinement	Prompts were used to critique left-screen, right-screen, and vessel monitoring screen layouts. They were also used to discuss map density, command workspace structure, camera use, response bars, and multi-screen responsibility.	The author made the final design decisions in Figma and checked whether suggestions were feasible, clear, and consistent with the developed prototype.
Visual design system support	Prompts were used to describe colour use, typography hierarchy, component states, icon usage, and OpenBridge-inspired maritime HMI styling.	The author selected the final colour palette, typography styles, components, and icons. AI was used to describe and organise these decisions, not to define a complete design system automatically.
Evaluation writing and limitation framing	Prompts were used to formulate the evaluation aim, evaluation method, SAGAT-inspired question structure, feedback summary, design revision table, and limitation statements.	The author ensured that the evaluation was described as a formative design review, not as formal validation. Claims about safety, workload, situation awareness, and operational performance were kept limited.
AI-assisted visual exploration	Image-generation prompts were used to explore early visual concepts, workstation atmosphere, cover images, and presentation material.	AI-generated images were used only as visual exploration or presentation support. They were not treated as user research data, evaluation evidence, or final interface screens.

F.1 Prompting Approach

Most prompts were written as task-specific design or writing requests rather than open-ended questions. The prompts usually included the current design context, the intended thesis scope, and the expected output format. For example, prompts were used to ask for a clearer section structure, a more direct academic paragraph, a table format for appendix material, or a critique of whether a screen layout matched the role of monitoring, communication, and supervisory command.

The prompting process was iterative. When an AI response was too broad, too generic, or inconsistent with the prototype, the prompt was refined with more specific constraints. Typical constraints included: keeping the scope limited to monitor-

ing, communication, and supervisory command; avoiding full manual teleoperation; matching the final prototype; using simple academic English; and avoiding claims that were not supported by the evaluation.

F.2 Use of Prompt Outputs

The outputs from AI prompts were used in four main ways. First, they were used as drafting material for report sections, tables, captions, and appendix text. Second, they were used as critique material to identify unclear logic, repeated content, or weak design arguments. Third, they were used as translation and language support between Chinese working notes and English thesis writing. Fourth, they were used as ideation support for visual concept exploration and presentation images.

The author did not use AI outputs as final evidence. AI-generated suggestions were checked against the actual prototype, project discussions, uploaded literature, evaluation feedback, and the thesis scope. When an AI output introduced unsupported functions, overclaimed evaluation results, or described design features that were not present in the prototype, that content was removed or rewritten.

DEPARTMENT OF INDUSTRIAL AND MATERIALS SCIENCE
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY