CHALMERS
UNIVERSITY OF TECHNOLOGY

Designing a Web-Based Graphical User Interface for Managing Digital Twin Simulations in Healthcare

Master's Thesis in Biomedical Engineering

Alva Cider
Norea Ronnby

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's Thesis 2025

Designing a Web-Based Graphical User Interface for Managing Digital Twin Simulations in Healthcare

Alva Cider
Norea Ronnby

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2025

Designing a Web-Based Graphical User Interface for Managing Digital Twin Simulations in Healthcare

Alva Cider
Norea Ronnby

Supervisor: Mattias Seth, Electrical Engineering
Examiner: Stefan Candefjord, Electrical Engineering

Master's Thesis 2025
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Visualization of a Graphical User Interface for Clinical and Technical Users for Managing Digital Twin Simulations in Healthcare.

Typeset in LaTeX
Gothenburg, Sweden 2025

Designing a Web-Based Graphical User Interface for Managing Digital Twin Simulations in Healthcare
Alva Cider
Norea Ronnby
Department of Electrical Engineering
Chalmers University of Technology

Abstract

With an aging population, a growing shortage of healthcare professionals, and increasing demand for care at home, the need for digitalization in healthcare is becoming more urgent. Integrating multiple medical devices into large-scale systems adds complexity, making thorough testing essential prior to real-world deployment. At Chalmers University of Technology, there is an ongoing research initiative developing a digital sandbox environment for managing digital twin simulations in healthcare. This thesis aimed to design a web-based graphical user interface (GUI) for the sandbox environment. Using a human-centered design approach, the GUI was developed through an iterative process involving prototyping and user testing, grounded in a realistic fall detection simulation case. The resulting interface features two integrated but distinct views: a patient-centered view for clinical users and a system-oriented view for technical users, enabling effective cross-disciplinary collaboration. Although still an early prototype, the GUI provides a solid foundation for further development and serves as a foundation for future development and integration of more complex scenarios.

Keywords: Sandbox, Digital Health, Digital Test Environment, Graphical User Interface, Healthcare Digitalization, GUI Design, Digital Twin.

Acknowledgements

We would like thank all the test participants who contributed, making this project possible. We are especially grateful to Robert Höglund, Peter Kelly, and Ulrika Björner for sharing their insights into the healthcare and regional care sectors during the interview process. Our warmest thanks go to our supervisor, Mattias Seth, and assistant supervisor, Fanny Apelgren, for their continuous support, valuable input, and guidance throughout the project. We also appreciate the contributions of Stefan Candefjord, whose thoughtful feedback during key phases of the project helped shape its direction.
Thank you all.

Alva Cider, Norea Ronnby, Gothenburg, May 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

API	Application Programming Interface
CDS	Clinical Decision Support
DT	Digital Twin
EHR	Electronic Health Record
FHIR	Fast Healthcare Interoperability Resources
GUI	Graphical User Interface
HCD	Human-Centered Design
ISO	International Organization for Standardization
IoMT	Internet of Medical Things
IoT	The Internet of things
MDR	Medical Device Regulation
UI	User Interface
UX	User Experience
VGR	Västra Götalands Region

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

With an aging population worldwide, an increased demand for care solutions that allow patients to remain in the comfort of their homes, and a shortage of healthcare professionals, there is a growing need for digitalization within the healthcare sector [1]. The shift towards more digitalization is essential to ensure good healthcare for all individuals. As the digitalization transforms the healthcare sector, the need for more medical technology creates new complexities regarding data handling, regulatory aspects and interoperability. The integration of numerous medical devices into large-scale systems introduces significant complexity. Past implementation challenges highlight the importance of thorough testing [2]. This has led to a growing interest in sandboxes, i.e. controlled environments that facilitate testing and development of new technologies [3]. Sandbox environments offers the ability to create digital twins (DT), i.e virtual models of patients, devices or entire systems, and simulate complex scenarios [3], [4]. Helping identifying bottlenecks, potential risks, and mismatches saving both time and cost [4]. Such capabilities support smoother development by enabling testing throughout the entire development process, before testing in real world setting [2]. This is particularly beneficial in healthcare, where regulatory procedures can be time-consuming, and early testing can make great difference [3]. Additionally, sandbox environments can simplify deployment of updates without causing complications or operational disruptions for patients, as many rely on their digital devices and health applications.

At Chalmers University of Technology, there is an ongoing research project called ASAP Autumn leaves [5]. ASAP stands for Acute Support Assessment and Prioritization and is a collective name for several projects within pre-hospital digital health. Autumn Leaves is a project aimed to improve pre-hospital emergency care response for fall related accidents among the elderly. By enabling automatic fall detection, streamlining the alarm chain, and digitalizing information sharing, assessment, prioritization, and management, the project supports a more efficient and effective fall injury care process. All projects within the ASAP initiative are based on the same technical framework, with a digital sandbox being developed to enable DT simulations of real-world scenarios. This sandbox offers the ability to set up complex systems and explore their potential in a controlled environment. The sandbox has four main building blocks, which are fundamental to the sandbox environment and are listed and described in Table 1.1. These building blocks are intended to serve as a foundation, enabling tailored simulations to be performed with any desired

ASAP Building Block	**Functionality**
Providers	Devices or information providers responsible for initiating a simulation by sending data to ASAP Services for further processing.
Services	Software modules responsible for processing information coming from ASAP Providers.
Consumers	GUI (Graphical User Interface) responsible for displaying information from ASAP Services to end users.
Scenario Engine	Orchestrator that enables the user to determine what to simulate and how it should be conducted, ie., decide on the simulation scenario, what components to be used and how they should fit together.

Table 1.1: Main building blocks of the ASAP sandbox, from [2].

combination of components [2].

To maximize the ASAP sandbox's usability and enable interdisciplinary collaboration, it is being developed to support both technical and non-technical users. Users with technical expertise will be able to add components and data into the sandbox, making them accessible to all sandbox users for setting up simulations. Data and components can also be generated using a synthetic data generator, enabling the evaluation of fictitious cases [2]. This facilitates collaboration between professional roles, regardless of technical background, in the development of future healthcare solutions.

The sandbox environment is being developed based on the Fast Healthcare Interoperability Resources (FHIR) framework, a standard for the structured and secure exchange of healthcare information electronically [6]. When combined with standardized clinical terminologies such as LOINC and SNOMED CT, semantic interoperability can be achieved, creating a common understanding of data between systems [7]. This enables that all components added can be connected and data can be transferred seamlessly. In other words, users with less technical experience, such as clinicians, can utilize the sandbox to explore the potential of digitalization within the healthcare sector, without needing to manage the underlying technical complexities. All integrations occur behind the scenes and are fully hidden from the user.

However, if components added to the sandbox do not adhere to these standardized terminology frameworks, semantic interoperability is not achieved. While data may still be transmitted, it will not be properly understood without additional translation efforts. This underscores the importance of testing through simulations, as real-world implementations often expose issues such as semantic mismatches, inconsistent data interpretation, and integration failures. These issues can be mitigated by adding terminology frameworks.

For the sandbox to function as intended and allow users to create simulations effectively, a user-friendly interface is essential. Especially for healthcare professionals, it is crucial for enabling interaction with DTs without extensive training [4]. In this project a graphical user interface (GUI) for the sandbox was developed, focusing fall detection systems.

Another important aspect for the sandbox to reach its full potential and serve as a powerful tool in healthcare development is the availability of a simulation result protocol [2]. The protocol is accessible after each simulation and allows users to identify issues, understand outcomes, and use the results as a foundation for implementing new medical technologies. Therefore, this project also investigated the design of the simulation result protocol for visualization of simulation results.

1.2 Purpose

The purpose of this thesis was to create a web-based user-friendly GUI for the ASAP sandbox environment that facilitates the control and management of DT simulations. These simulations aim to simplify the testing of digital health solutions in real-world conditions. The primary focus was on the ASAP: Home/Autumn Leaves project, which aims to optimize early pre-hospital care in emergency settings within home environments, focusing on fall accidents among the elderly. The GUI aims to provide an intuitive setup process and clear guidance throughout a simulation workflow, enabling efficient and seamless testing of various scenarios prior to clinical testing or real-world implementation for both healthcare professionals and researchers.

To accomplish this, following research question was formulated:

- How can a sandbox GUI be designed, using a human-centered design approach, to effectively address the needs, workflows, and usability expectations of both technical and clinical users in the context of healthcare simulations?

  – Do digital sandbox environments exist within the healthcare domain, and if so, how are they designed, to what extent are they accessible, and what purposes do they serve?

  – Which types of fall detection devices and stakeholders are of relevance when simulating fall-related incidents in a healthcare context, and what commercially available devices could be used in such simulations?

  – How should a simulation result protocol be designed to ensure that simulated outcomes are interpretable, repeatable, and valuable to different user groups?

1.3 Goals

The project goals were defined as follows to achieve the overall purpose of the project:

1. Define user requirements:
   - Identify and analyze the needs of two primary user groups:
     - Technical users (e.g., engineers and researchers): Require access to advanced technical features and control of simulations
     - Clinical users (e.g., healthcare professionals or staff involved in operational development): Require an interface tailored for users with less technical knowledge, focusing on simplicity and clarity.
   - Document specific user requirements for each group to guide the design process.

2. Design a user-friendly GUI:
   - Create an intuitive interface for each user group through an iterative design process, ensuring:
     - Comprehensive navigation: Easy and logical access to all features.
     - Clear workflow: Streamline processes for setting up and running simulations.
     - Usability: Tailored experiences for both technical and non-technical users.
   - Ensure that the GUI supports setting up a simulation representing a real world scenario.

3. Develop a simulation result protocol:
   - Design a protocol that provides insights into simulated system performance.
     - Include functionality to:
       * Analyze interactions between systems.
       * Highlight problematic areas and provide alerts for potential issues
     - Create an intuitive protocol structure that ensures:
       * Readability: Easy to read and follow.
       * Clarity: Easy to understand performed simulation.
       * Repeatability of simulation: All elements and settings presented.

1.4 Limitations

1.4.1 User Requirements

Due to the project's time constraints, the participant recruitment process was limited. As a result, the number of interviewees was kept low to ensure that only relevant individuals were included. To compensate for the limited participants, a broader group with relevant technical experience was recruited, rather than targeting those with the most precise background. This approach was intended to mitigate the risk of biased test results.

1.4.2 GUI

The developed GUI primarily focuses on fall accidents within the ASAP project: Home/Autumn Leaves, based on a representative case scenario. As a result, it does not encompass all functionalities that might be required in the final version of the sandbox GUI. The current design is limited to the simulation setup view and does not include the interfaces for running or completed simulations. However, the potential design of these additional views is discussed in the discussion section.

The aim of this project was to design a GUI that could serve as a foundation for a functioning web application. However, the GUI in this project was never intended to be fully operational or a complete web application. The primary focus was on structure, functionality, and clarity of information presentation, with less emphasis placed on visual design elements such as color schemes and typography. Furthermore, the scope was restricted to a single interface and layout per user group, and aspects such as responsiveness were not addressed in this work.

Another limitation was that no functional simulations were performed. Only the interactive design of the GUI was developed. The prototypes were not translated into executable code or integrated into a working website.

1.4.3 Simulation Result Protocol

The simulation result protocol was designed with broad guidelines due to the early phase of the sandbox development. The design of the protocol focused on how information should be presented, rather than specifying exact details, since such details are likely to evolve over time as the sandbox is developed further.

2

Theory

This chapter presents theoretical and regulatory concepts relevant to sandbox environments in healthcare, providing important background to understand the complexity of health innovation. While these concepts are not always directly revisited in the development or discussion, they inform the design of a flexible and realistic GUI. The theoretical foundation also supports the selection of components and the structuring of user interactions to reflect realistic workflows and user needs in healthcare environments. The chapter also provides insights into Human-Centered Design (HCD), design methodologies, and GUI elements that inform the GUI design and its realization in the prototype.

2.1 Simulations

Simulations are models of real-world systems or processes designed to closely imitate their key characteristics and behaviors [8]. They provide a controlled environment where different scenarios can be tested without risking harm or disruption to actual systems in the real world.

In a simulation environment, Digital twins (DT) can be used to understand functioning and effects of physical objects [4]. The DT technology is mapping the real world into the digital world and enables functionalities as simulation, prediction and optimization [4]. By realizing the interactions between the real and digital world, DTs enables insight in real time outcomes of different scenarios. A DT is a representation or virtual copy or model of a physical object in a simulation environment allowing for understanding the objects functioning, effects and specifications [9]. The technology is characterized by data fusion, life cycle coexistence and the ability to optimize, and control physical entities [4]. DTs has its origin in the aerospace industry of the US military but has evolved to other sectors across industries, including healthcare. In healthcare, DTs are commonly used to model patient outcomes by creating digital representations of patients, medical devices and systems [4]. Modeling patients can assist in predicting the effects of various drugs and understanding the best treatment choices [9]. DTs allows repeated simulations, tests and improvements of physical objects until the model performs as desired [10]. Once that occurs, the model is ready for manufacturing or improving in the real world. By using DTs, errors and risks can be identified and mitigated in advance [9]. This enables predictive analytics, personalized medicine, and operational efficiency improvements in a virtual environment.

It is important to systematically document results from models and simulations to ensure transparency, reproducibility and traceability [11]. This can be done through a document, including the simulation setup and outcomes.

2.2 Sandboxes

The sandbox approach is a mechanism facilitating innovation in multiple sectors [12]. It is not a well-established definition, and today there is a broad use of the term "sandbox". According to [12] a sandbox is referred to as a "safe space" for development, testing and evaluation. Mainly, there are two types of sandboxes: Regulatory and the sandbox as a testing environment [3].

2.2.1 Regulatory Sandboxes

The first sandboxes were the regulatory sandboxes, used to promote innovation in highly regulated sectors, initially within financial technology [3]. Since then, sandboxes has expanded to several other fields.

Regulatory sandboxes provides a controlled environment giving the innovators the opportunity to test new products and services and introduce an innovation to the market at a lower cost and in a safer way since testing has been more rigorous [13]. They can be used as tools for developing evidence about how new products, technology or business models work and what outcomes it will produce when implemented in regulated sectors.

There is a growing interest in regulatory sandboxes generally and the number of countries using them in digital health is increasing [13]. In healthcare, sandboxes can be useful for multiple stakeholders. Innovators and entrepreneurs can receive regulatory guidance and learn how their products and services can integrate with existing infrastructure and other systems. Medical technology is subject to extensive regulation, with laws and requirements differing across countries. Section 2.2.2.1 provides an overview of the key regulations that must be complied with.

Patients and population can benefit from reviving early access to new innovative products that are in conformity with existing regulations, minimizing risks at a lower cost [13]. Government and regulators can benefit from promotion of development of products and services following regulations and patient rights as well as the evaluating the effectiveness of existing and new regulations. Further, claims from innovators can be tested in a sandbox environment.

2.2.2 Sandbox as a Testing Environment

The concept of a sandbox as a testing environment is described in the literature using a variety of terms. Some commonly terms used for describing these sandbox are:

- Simulation Sandbox [14]
- Digital Sandbox [15]
- Technical Sandbox [16]
- Synthetic Data Sandbox [17]

The sandbox as a testing environment are one of many approaches to testing throughout the Innovation Life-cycle [3]. An illustrated view of how different testing approaches relate to each other, according to [3], based on the Innovation Life-cycle Stage and environment type can be seen in Figure 2.1. According to this illustration, sandboxes are positioned between Living Labs and Real World Testbeds. Living Labs, can be either in real world or simulated environments and for early testing [3]. Sandboxes are safe spaces and are further into the development stage. Real World Testbeds are testing in the real world environment the product/service is meant to operate.

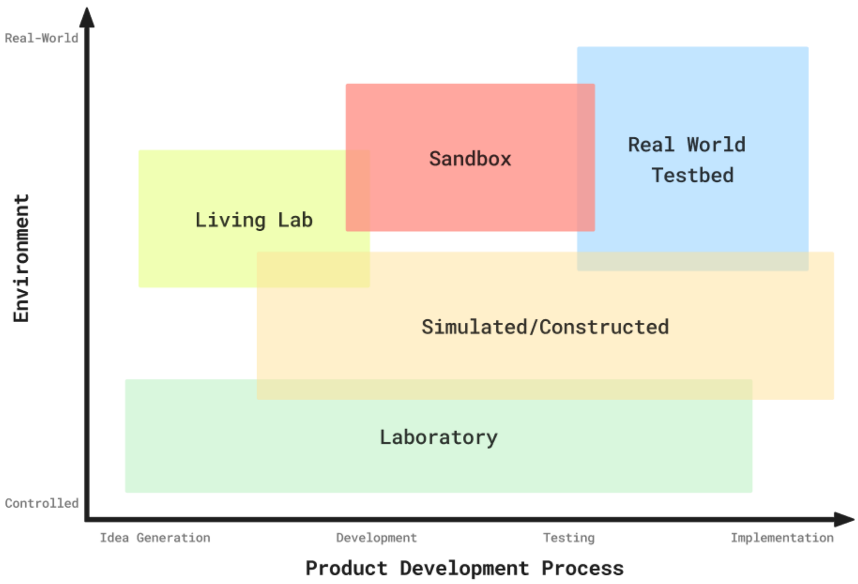


Figure 2.1: An illustration of how [3] defines sandbox environments in relation to other testing methods.

2.2.2.1 Regulations and Standards in Healthcare

Regulations that can either be tested for compliance within regulatory sandboxes, or temporarily bypassed during early-stage testing conducted in a sandbox used as test environment, include:

- **Medical Device Regulation (MDR):** Regulation governing the development, verification, labeling, production, and more of medical devices in Europe [18]. In Sweden, Läkemedelsverket (the Swedish Medical Products Agency) provides guidance and designates third-party bodies responsible for assessing compliance with the regulation.
- **Food and Drug Administration (FDA):** Ensures that medical devices adhere to corresponding national regulations in the United States [19].
- **General Data Protection Regulation (GDPR):** Governing how personal data must be processed and protected within EU [20]. GDPR is fundamentally about safeguarding individuals' rights and freedoms.
- **Health Insurance Portability and Accountability Act (HIPAA):** Regulate privacy concerns within the U.S [21].

- **Patient Data Act (PDL):** Regulates the management of patient data, ensuring patient safety and maintaining high standards of care in Sweden [22].
- **Artificial Intelligence Act (AI Act):** Regulate Artificial intelligence within EU [23]. As a part of the implementation, regulatory sandboxes is to be available by 2026 [24].

In addition to legal regulations, various standards also apply during the development and commercialization of medical devices. International Organization for Standardization (ISO) and International Electromechanical Commission (IEC) standards are among the most widely used frameworks [18]. For example, ISO 13485 specifies requirements for a quality management system in the development of medical devices. However, not all standards are harmonized with MDR, meaning that MDR may impose additional requirements beyond those outlined in ISO standards. Therefore, it is essential to ensure full compliance with MDR, not only the applicable ISO or IEC standards. To demonstrate compliance with EU laws and regulations, products shall have certain markings. CE marking is one such indicator, signifying that a product meets "high safety, health, and environmental protection requirements" [25]. CE marking is mandatory for products sold within the European Economic Area (EEA).

2.2.3 Synthetic Data in Healthcare

Data is a fundamental component in research, innovation, and development across various fields. In healthcare, data access is strictly regulated due to privacy concerns, making it challenging to obtain the necessary information for the efficient development and implementation of new products, services, research or systems [17]. Today, anonymization and pseudonymization are commonly used solutions for enhancing privacy of health related data [26]. A more innovative approach to overcome this challenge is using synthetic data, allowing organizations to share datasets more broadly while maintaining privacy [17]. Synthetic data can be generated using various statistical methods, such as multiple data imputation and Bayesian bootstrap, as well as AI-driven techniques. Specific to electronic health records (EHR), tools like Synthea generate clinically realistic synthetic patient records using public health statistics and clinical guidelines. This way of generating data ensures it maintains the statistical properties of real-world data without exposing sensitive patient information.

The growing interest in synthetic data is largely driven by its potential to enhance analysis and accelerate technological development by providing more accessible and representative healthcare datasets [17]. A common concern is how closely synthetic data resembles real patient data. While it is not identical, exact replication is not necessarily the goal. Instead, synthetic data is created with privacy protection and bias mitigation in mind [27], which shifts the focus toward evaluating the utility of the data rather than its perfect resemblance to reality.

A major advantage of synthetic data is that it allows for broader access to sensitive healthcare information without compromising patient privacy [27]. By safeguarding privacy, enabling faster and wider data sharing, and offering larger and more flexible datasets than are typically available from real-world sources, synthetic data addresses three critical challenges in making healthcare data accessible [17]. It facilitates testing, validation, and innovation while supporting compliance with regulations such as GDPR, PDL, and HIPAA.

2.3 System Architecture and Communications

Modern digital systems rely on efficient system architecture for enabling functionality and scalability [28]. The client-server model is a basic architecture, supporting data exchange and processing across distributed networks. Communication within and between systems is established through protocols enabling data, features and functionality exchange [29]. To enable communication between software applications standardized communication protocols like APIs define consistent methods for transmitting information [29].

2.3.1 Client-Server Model

The client-server model is based on the distribution of functions between two types of autonomous processes: Clients and Servers [30].

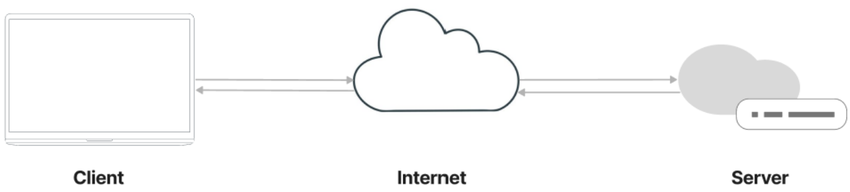


Figure 2.2: Simple Client-server architecture.

A client is any system, application, or process that requests services from a server, while a server is responsible for fulfilling those requests [30]. This paradigm has been implemented because it allows information servers to centrally manage shared computing resources while also defining rules for their administration. Client-server architecture forms the foundation of cloud computing, where a client (typically a computer or other device) requests resources or services from a server over a network connection. The sever processes the request and then responds to the client. This communication is facilitated by APIs, such as REST APIs, which operate over established protocols like HTTP, as described in the section below [31]. The GUI is the front-end to the client and is enabling simple interactions for the end user with the system without deeper network or IT knowledge [32]. Well-designed interfaces simplify complex systems, allowing users to focus on completing their tasks rather than understanding the underlying infrastructure.

2.3.2 Communication in Digital Systems: APIs and Protocols

Application programming interfaces (API) is a code based connection allowing different software systems to communicate [33]. They specify how the client should interact with different software components. The API is a set of protocols giving the rules for enabling the communication, defining how requests and responses should be structured [29]. There are different category of APIs based on how they operate and where they are used. Representational state transfer (REST) APIs, commonly used in web applications, communicate via HTTP [31]. Standard database operations are carried out using HTTP methods such as GET, PUT, POST, and DELETE, which allow for reading, creating, updating, and deleting records. Message queuing telemetry transport (MQTT) protocol is widely used for enabling low-latency communication between devices connected over the internet, enabling real-time data transmission [34]. Frameworks like FHIR is commonly used as a structured framework for standardizing healthcare data, enabling integration with EHRs [35].

2.3.3 Web Applications

A web application is a software application that runs on a remote server and is accessed through a web browser over a network [36]. It follows a client-server architecture, where the web browser serves as the client interface. Unlike traditional websites that mostly deliver static content, web applications provide dynamic, interactive services by responding to user input and communicating with backend servers in real time. They can facilitate tasks such as data entry, real-time updates, and user-specific content rendering, allowing for complex operations to be executed via a user-friendly interface .

2.3.3.1 Monolithic Architecture and Microservices Architecture

There are two software engineering paradigms that are dominating web application development [37]. These are monolithic and microservice-based architecture. Monolithic services are built with a single code base including multiple services that are not independently executable. The services uses different interfaces as HTTP(s)/HTML, REST API and Web services for communication with external systems and end users. Microservices on the other hand brakes down the services into small independently deployable services. Making microsevice-based applications easier to scale in specific parts.

2.3.4 Internet of Things and Internet of Medical Things

The internet of things (IoT) is a widely used technology enabling interconnected devices to communicate via the internet, allowing automated data exchange, remote monitoring and decision-making [38]. The increasing adoption of wireless communication has led to the rapid integration of IoT applications across various domains, including healthcare.

IoT is defined by the International Telecommunication Union (ITU) as "A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies" [39]. IoT connects physical objects to digital platforms, allowing real-time data and analysis from any location [40].

Relying on structured network architecture, IoT systems manage device communication, data transmission and processing [38]. The most common model follows the client-server model, where clients (devices) send and receive messages through a central server. Typically, an IoT system consists of multiple clients, which must establish a connection with a server before exchanging data using IoT-specific communication protocols. The architecture in common for IoT is the centralized network that can be seen in figure 2.3.

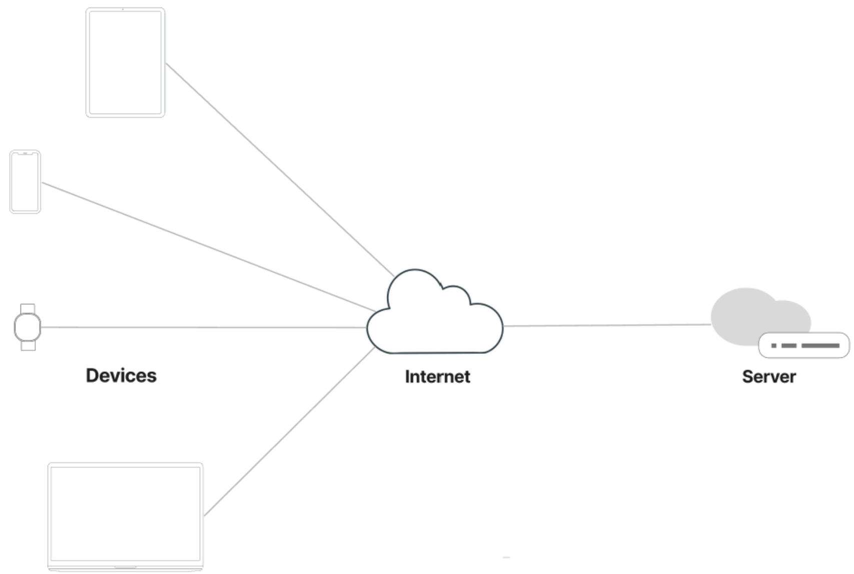


Figure 2.3: Internet of Things (IoT) network architecture.

For IoT devices to function seamlessly, standardized protocols ensure efficient, secure and scalable data exchange and communication between clients and services.

In healthcare, IoT plays a crucial role for integrating medical devices, wearable sensors, and cloud-based analytics [41]. When applied specifically for medical purposes, IoT is often referred as the Internet of Medical Things (IoMT). It comprises a network of devices, sensors, and software systems that enables real-time data exchange and healthcare data management. IoMT enables smart healthcare including remote monitoring of patients and AI-assisted diagnostics and predictive analysis [40]. By establishing interconnected systems, IoMT have the potential to enhance healthcare workflows, improve patient outcomes and contribute to personalized and efficient medical solutions.

Interoperability is a critical challenge in IoT systems due to a large number of different technologies, architectures, security mechanisms and communication protocols being used [42]. Using a flexible middleware between has been suggested as a solution due to difficulties of solving the problem by standardization. For IoMT platforms, interoperability is often the limiting factor for streamlining health care delivery, hindering communication and risking patient safety [43]. The interoperability of IoMT can be broken down to 6 different levels of interoperability.

- Device - Ensuring that heterogeneous devices can connect.
- Network - enabling information exchange over the internet.
- Syntactic - enabling processing of the dataformats and structures.
- Semantic - enabling common understanding of the message content.
- Cross-Platform - enabling different platforms to work together seamlessly.
- Cross-Domain - enabling communication across organizational and national borders [43].

By combining various technologies and standards from different levels, higher levels of interoperability in IoMT platforms can be achieved [43]. For pre-hospital care, the primary standards enabling interoperable IoMT platforms are REST APIs, Wi-Fi gateways, JavaScript Object Notation (JSON), MQTT, FHIR and webbsockets.

A different approach for addressing interoperability challenges is the use of middleware platforms [42]. Technologies like FIWARE and OpenICE acts as intermediates that aggregate and standardize data formats from heterogeneous sources [44], [45]. Middleware can convert messages exchanged between heterogeneous systems into a standard form that is understood by all parties [46]. Meaning instead of all devices following the same standard, a middleware can translate the data to ensure mutual understanding. FIWARE is an open-source platform that provides tools for collecting, processing and visualizing data form divers IoT systems [44]. OpenICE, on the other hand, supports integration of various medical devices through software adapters [45]. By implementing the Integrated Clinical Environment (ICE) standard, which defines a safe, patient-centric system architecture, OpenICE contributes to improved interoperability and enhanced patient safety.

By serving as a translation layer between systems, middleware solutions such as FIWARE and OpenICE lay the foundation for seamless data exchange. However, if systems already share common terminologies and data models, the reliance on middleware for translation can be reduced. Some commonly used terminologies are outlined below.

2.3.4.1 Terminology Systems

Terminology services facilitate standardized coding and classification to enhance semantic interoperability. For example, terminologies such as the International Statistical Classification of Diseases and Related Health Problems (ICD), which is an international standard for coding diagnoses, health states etc. [47]. Another terminology is SNOMED CT which can cross map to other classifications and terminologies [48].

LOINC is a standardized clinical terminology used for coding health measurements, observations, and documents [49]. It provides a structured way to represent clinical questions, which can be answered using other terminologies such as SNOMED CT, or with numerical values.

2.3.5 FHIR

Fast healthcare interoperability resources (FHIR) is a standards framework developed by Health Level Seven (HL7) [6]. Designed as the next generation standard for safe data exchange in healthcare, it integrates modern web standards with a strong focus on interoperability. The FHIR standard is highly versatile and can be applied in EHR-based data sharing, cloud communications, mobile applications, and server communication between healthcare providers. The framework builds on and adapts RESTful practices, facilitating data exchange via messages, documents or service-based architectures. FHIR addresses syntactic interoperability by standardizing the structure and format of healthcare data, but it does not fully ensure semantic interoperability, which requires a shared understanding of the meaning of data [7]. This is typically achieved through common terminologies or ontologies such as SNOMED CT or LOINC, which FHIR supports but does not enforce.

Healthcare standards struggles to handle the diverse and evolving needs of healthcare systems [6]. As more fields and optional features are added over time, implementation complexity and costs increase. One approach to this is to use custom extensions, but these tend to lead to compatibility issues and fragmentation. FHIR addresses these challenges by providing a structured, yet flexible framework for extending resources through Profiles, allowing applications to define specific constrains while maintaining broad compatibility. Profiling is not specific to FHIR, but it is a key enabler that allows FHIR to maintain high flexibility and support customized implementation. Today, different healthcare systems rely on various, often incompatible, standards, making data sharing and interpretation of patient data across systems difficult [50].

The FHIR standard is based on modular components called resources, which can be combined to build flexible and interoperable healthcare systems [6]. Information can be exchanged either as individual resources or grouped together in bundles, which consist of multiple related resources. Each healthcare entity, such as a patient, diagnosis, or medication, has a corresponding resource that serves as an instance-level representation, simplifying data exchange. RESTful APIs are used to manipulate resources, while JSON, XML, and RDF are supported as data formats.

For development and integration of medical applications in EHR and other clinical systems a technical standard called Substitutable Medical Applications and Reusable Technologies (SMART) on FHIR has been established [51]. In addition to FHIR models and API, the SMART on FHIR standard also encompasses authorization, authentication an UI integration. Following this standard, applications can seamlessly be integrated into healthcare system without requiring vendor specific adjustments. This allows third-party applications to be launched directly within, for example, an EHR system [52].

Another service utilizing FHIR is CDS Hooks, which provides cards containing information, suggestions or links to additional information or SMART on FHIR applications in the user interface [53]. These cards are triggered in real time to support the clinicians workflow. To generate relevant recommendations, the CDS service requires access to specific FHIR resources [54].

2.4 GUI

A graphical user interface (GUI) enables human-computer interaction by providing a visual interface that hides the underlying programming code, allowing users to interact with the system without worrying about the technical infrastructure [32]. Instead of relying on text-based commands, GUIs use graphical elements such as windows, icons, lists, buttons and menus to facilitate communication between the user and the system. User interact with these elements primary through pointing devices, most commonly a mouse. In this section, common design methods, principles and components are described.

2.4.1 Human-Centered Design Process

Human-centered design (HCD) is a design approach that places users, along with their needs and requirements, at the core of the design process [55]. It incorporates principles of human factors and ergonomics, usability and user behavior, with the aim of improving system performance, user satisfaction, and accessibility. HCD serves as a complementary strategy to other system design methods, ensuring that interactive systems are optimized for the people who use them.

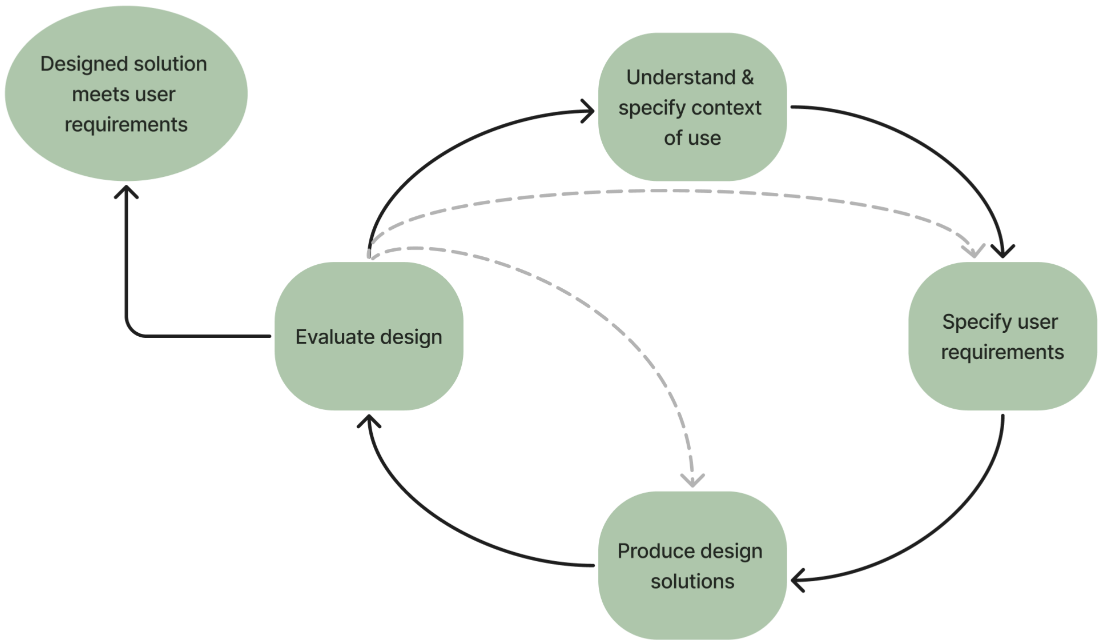


Figure 2.4: The iterative human-centered design process.

The process is inherently iterative, emphasizing continuous refinement based on user feedback throughout development. According to the ISO 9241-210 standard, the process consists of repeated cycles of understanding the context of use, specifying user requirements, creating design solutions, and evaluating those solutions with users, see Figure 2.4. This cyclical approach enables designers to uncover latent user needs, reduce design errors early, and incrementally improve usability. In this project, these principles guided the development process, ensuring that user needs were continuously re-evaluated and integrated into each design iteration. The most important methods applied to support this iterative approach are described below:

2.4.2 Normans Design Principles

A widely recognized set of design principles in Human-Computer Interaction (HCI) is Norman's seven fundamental design principles [56]. These principles provides a framework for designing intuitive and user-friendly interfaces. The principles are:

1. Discoverability
2. Feedback
3. Conceptual model
4. Affordance
5. Signifiers
6. Mapping
7. Constraints [56]

Discoverability refers to the ability of the user to perceive and determine, in the design, the actions that are possible and the current state of the system [56]. This should be clear without trial and error. Feedback ensures the user receives clear, continuous information about what actions has been made and the result of the actions. This prevents confusion as well as enhances the confidence of the user. Conceptual model refers to that the design should give the user understanding about the system. Helping the user to get a mental model of the system and a feeling of control. A well-structured conceptual model will enhance discoverability and evaluation of results of the system. Affordance describes the way design element should suggest their intended use. The appearance of an object or interface component should indicate how it can be interacted with. Signifiers are signs guiding users on how to interact with the interface. While affordance is inherent in the design, signifiers are intentionally placed indicators clarifying interaction possibilities, e.g. labels or icons. Mapping refers to the relationship between controls and their actions. Effective mapping ensures that interface elements align logically with their function, using spatial layout and temporal contiguity to enhance usability. Constraints guide user interactions by limiting available actions based on physical, semantic, cultural and logical constraints. These constraints help prevent errors and make interfaces more intuitive.

2.4.3 Prototyping

There is no single correct way to prototype, and various approaches have been proposed and debated over time. According to [57], the linear prototype design process can be refereed to as the traditional method. This approach consists of five sequential stages; sketching, wireframing, mockups, interactivity, and development, as illustrated in Figure 2.5. Each of these stages contributes to the gradual refinement of an application's interface and functionality. Sketching includes brainstorming to generate the most simple basic prototyping, which can be performed with paper and pencil. Wireframing, often in low fidelity (LF), is a basic sketch to put the architecture in place of an applications interface. It is often used for evaluating the early concepts of the design. Mockups are more detailed wireframes. This includes visual details as colors and typography. Interactivity includes animation or light coding to the mockup to show functionality to the application. Development is finalizing part where the product is programmed into a full functioning application. While this linear workflow provides a structured approach, modern prototyping has largely shifted toward an incremental and iterative process. In practice, each stage can act as a prototype in its own right, capable of being tested, reviewed, and refined. This evolution aligns better with HCD principles, ensuring that feedback is incorporated early and often to meet user needs more effectively.

Figure 2.5: Prototyping stages within the linear design workflow.

2.4.4 Personas

A persona is a fictional character used to describe the characteristics of a future user [58]. Personas are created to model the target users of the product or service and all main user groups should be presented by a separate persona. The number of personas should not exceed five to six, as higher number makes it more difficult to distinguish between them and remember their characteristics [59]. Key components to include when creating a persona are: name, age, occupation, hobbies, needs and frustration [60]. Personas should ideally be created based on high-quality collected data [61]. However, when limited data or resources are available, assumptions can be used instead. This approach is known as provisional personas or ad hoc personas. These personas are based on developers or designers experience, expectations and intuition about the user [59].

Lene Nielsen [59] proposes a 10-step approach, divided into four parts, to create a persona. The four parts and corresponding steps can be seen in Figure 2.6. The red main part in Figure 2.6, *Collection and Analysis of Data*, includes gathering and analyzing user data to build a general understanding of the product's users. The yellow main part, *Persona Description*, focuses on determining how many personas are needed and defining their key characteristics. The blue main part, *Idea and Scenario Development*, involves creating situations illustrating a reason for a persona using the product. Lastly, the green main part, *Acceptance from organization*, ensures that the personas should be accepted or possibly rejected of people within the organization or designers.

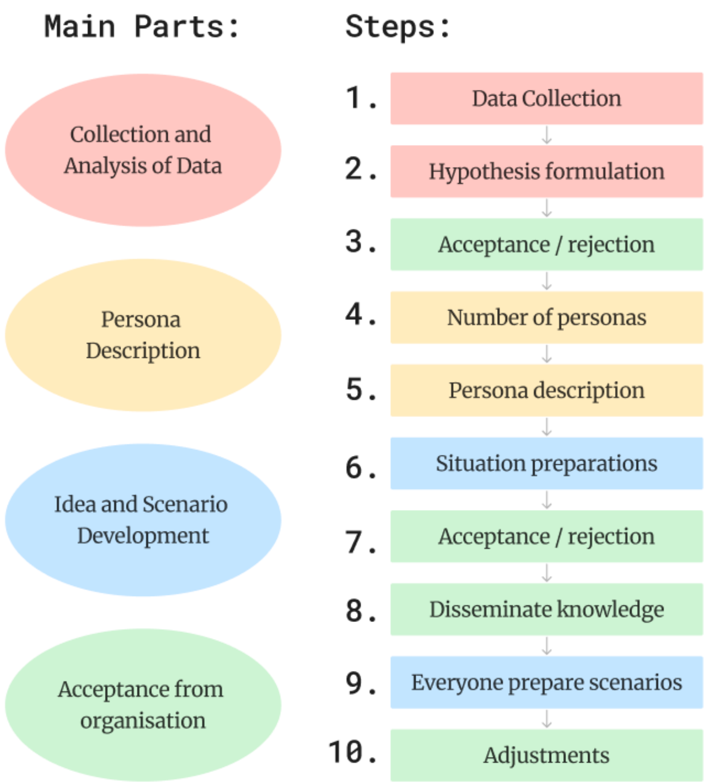


Figure 2.6: A 10-step approach divided into four main parts for creating a persona.

2.4.5 UI and UX

In the development of a GUI, both user interface (UI) and user experience (UX) principles plays a crucial role [62]. UI refers to the interaction between the user and system [63]. These interactions can include commands, content, appearance and graphical elements. The UI serves as the software medium that enables interaction and covers the visual design and layout of an application [64]. UI design focuses on providing an intuitive and accessible interface, allowing the user to effectively interact with the system. UX, on the other hand, encompasses the overall feel and experience when the user interacts with an application or any type of content [63]. UX is a broader concept that includes usability, usefulness, emotional impact and meaningfulness. As defined by [64], UX is " The totality of the effects felt by the users before, during, and after interaction with a product or system in an ecology". UX design focuses on how users interact with the system and weather the interface effectively supports the user's workflow.

2.4.6 User Friendliness and Usability

"User-friendly" is a commonly used term that is frequently referenced in various contexts, particularly in discussions of HCD and Usability. Usability is a broader concept then user-friendliness, according to ISO 9241-11 [65] usability defined as: "Usability: extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use".

Effectiveness represent that the user should be able to use the perform the desired task or what they want to accomplish and also that it should behave as expected from the user [66]. Efficiency means that the user should be able to reach their goal or complete the desired task within a certain time frame. Satisfaction is about the users perception from testing a product. Their feelings and opinions on how well the product met their specific needs are linked to their satisfaction.

2.4.7 Testing

There are multiple ways to evaluate whether a GUI functions as intended. Different testing methods can be applied at various stages of the design process. These methods are generally categorized into two main types: user testing and usability testing [57]. The following sections describe these methods in detail, including recommendations on the optimal number of participants for testing.

2.4.7.1 Usability and User Testing

User testing is typically conducted before or in the early stage of the design phase to gather users' preferences and expectations. This is used to validate user requirements and inform design decisions. In contrast, usability testing is commonly used later to assess the end user experience in terms of usability [67]. The four main insights gained from performing usability testing are usefulness, effectiveness, learnability, and user satisfaction. Usefulness aims to determine if the interface works for its intended use. Effectiveness examines whether the user can perform the desired task with ease. Learnability focuses on how easily the user can interact with and use the application, or whether practice is needed. Lastly, user satisfaction is about the user's experience and their attitude toward using the webpage.

Some common methods for usability testing include:

- **Task analysis:** Users are asked to perform specific tasks, revealing potential misunderstandings and providing insight into how they approach different actions [57].
- **Eye tracking:** By recording the user's eye movements, it is possible can see where they look and focus their attention [57]. This allows evaluation of which objects or areas receive the most attention and interest.
- **A/B testing:** A method used to compare different versions of a product against each other to determine which one users found work better for the outcome [57]. This test is often evaluated with the help of questionnaires. The users are allowed to try the GUI first and then answer the questions. This kind of comparison test can be utilized in all stages of the design process [66].

Some common user testing methods include:

- **Card sorting:** Participants organize cards (or post-it notes) with labels according to instructions. This reveals user expectations regarding layout, categorization, or navigation [57].
- **Interviews:** Participants answers questions about their preferences and suggestions for the product [57].

- **A/B testing:** As with usability testing, users interact with different alternatives to assess engagement and preferences [57].

2.4.7.2 Test Procedure and Participant Guidelines

To conduct effective testing, it is essential to create a test plan addressing the core questions: when, how, where, who, why, and what [66]. These elements are crucial for understanding the necessary steps and enables clear communication, so all participants involved are aligned on the test's purpose and requirements. The most important plan to include in the test plan is the research questions. What does this testing want to get answer to? The questions should be measurable or observable.

When a participant is to take a test, they should be given a script explaining what will happen [67]. The script should be brief (ideally under two pages), simple, and easy to understand. It should answer common concerns such as: how long the test will take, what type of questions they will receive, and what the test entails. If the prototype design is not fully interactive, it is important to inform that before the test i started. The developer can explain functions that are present but not yet functional, for example, to save time during development. Most importantly, it must be made clear that the system is being tested, not the user. Participants may feel pressure to perform correctly, but the goal is to understand how the interface works for real users, not to test user intelligence or knowledge.

The participant should also receive tasks or questions [67]. It is important to keep down number of task and question so the user keep interest in the assignment and prevent frustration.

2.4.7.3 Optimal Number of Participants for Testing

To determine whether a GUI provides a good user experience and work as intended for the user group, it is important to decide on the number of people who will test the interface between development stages. There is no statistically determined number of test participants needed to ensure all problems within the user interface will be discovered [64]. Instead, the ideal number depends on the specific context and goals of the testing. Nielsen and Molich [68] suggested in 1990 that testing with 4–5 users is sufficient to identify 80% of the usability issues. This is now considered a rule of thumb in user testing [64]. The reasoning is that major issues are often identified by the first few users, and when the number of test participants increases, fewer new issues tend to emerge [68]. However, this number has both been accepted and confirmed as well as questioned [64]. The appropriate number of participants should ultimately be determined based on the nature of the product, the complexity of the tasks, and the diversity of the user base.

This "rule of five" is based on the probability formula shown in Equation 2.1, where: P = discovery likelihood, p = detection rate, n = number of participants used [64].

$$P = 1 - (1 - p)^n \tag{2.1}$$

A key limitation of this formula is determining the participants' actual detection rate [64]. This rate is not constant and can vary depending on circumstances, timing, and other factors.

2.4.8 GUI Components

Most of GUIs are built using similar components, regardless of the application's purpose or design aesthetics [32]. Users have various ways to interact with an application to take action or use a command [58]. These actions enable the user to complete tasks, and a key consideration is that users often bring expectations based on their prior experience with other applications, such as where buttons or controls are located. Adhering to established standards and recognizing that a lack of originality can be beneficial is important. The goal in designing an interface should be to make actions available, easy to find, and supportive of the user's workflow. Another benefit of using common components is the availability of libraries that provide pre-built elements. Bulma is an example of a CSS framework offering such components [69]. Other widely used libraries include Bootstrap and Tailwind CSS [70], [71].

Below are some common GUI components:

- **Windows:** The window divides the screen into separate areas and is used to group and isolate functionality in GUI-based software [61]. In a window a user can display different programs, directories or data visualizations for example.
- **Button:** Buttons are typically rectangular with rounded edges, making them easily identifiable as an interactive element [61]. Buttons have the advantage of being easy to use even for the most inexperienced user , but the disadvantage of taking up a lot of space on the interface [58]. See Figure 2.7 for a visualization.

Figure 2.7: Representation of a button appearance.

- **Radio Button:**
  Radio buttons are characterized by their round shape and the fact that they appear in groups [61]. Selecting one button automatically deselects the others. See Figure 2.8 for a visualization.

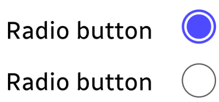


Figure 2.8: Representation of a radio button appearance.

- **State-switching buttons**
  State-switching buttons typically have two stages: on and off. When the switch is pressed, it moves to the corresponding side and indicates the other

mode [61]. They are often used in mobile applications but less frequently in desktop and web applications. See Figure 2.9 for a visualization.

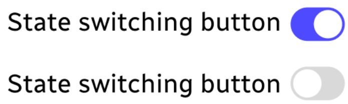


Figure 2.9: Representation of a state-switching button appearance.

- **Toggle buttons:**
  Toggle buttons are buttons remaining in pressed-down state once clicked, often suited for toolbars [61]. This is indicating the button operation continues until clicked again and not being being momentary. Toggle buttons can be small and rely on visual recognition rather than text. They can behave like radio buttons, deselecting others once activated. See Figure 2.10 for a visualization.

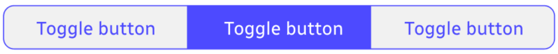


Figure 2.10: Representation of a toggle button appearance.

- **Check boxes:**
  Check boxes are used for selecting one or multiple options from a list [61]. They should be square in shape to avoid confusion with radio buttons. A check mark or an X appears, when clicked, and clicking it again removes the mark. Check boxes are intuitive but also forces the user to slow down since the corresponding text requires reading. See Figure 2.11 for a visualization.

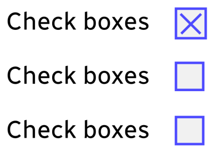


Figure 2.11: Representation of a check boxes appearance.

- **Spinners:**
  Spinner controls are mostly used for numeric input that needs to increase or decrease [61]. They include two small buttons on the right, typically half the size of the spinner control, often represented by small arrows pointing in corresponding directions. Users can change the numeric value either by clicking the arrows or by manually entering a number. See Figure 2.12 for a visualization.

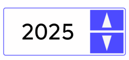


Figure 2.12: Representation of a spinner appearance.

- **Icon**
  Icons are small images that convey information to the user visually, instead of using text [72]. An representational icon is meant to represent an actual physical object and inherit property of it, making it faster for recognition and categorization than text. Icons makes learning easier for novices because they provide a set of familiar objects helping the user with inferences of the interface. The disadvantages of icons are difficulties to design the icons to represent the desired meaning. See Figure 2.13 for a visualization.

Figure 2.13: Representation of a icon appearance.

- **Links**
  Links are buttons without borders, characterized by the underline and usually blue color [58]. The underline can also be shown when hovering the link. Can be used when there is no need to drag a lot of attention to the link. See Figure 2.14 for a visualization of a link can appear.

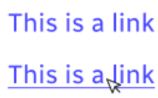


Figure 2.14: Representation of a link appearance.

- **Text Controls**
  Text controls is a display of static textual information, such as instructions, labels or outputs generated by the system [61]. And is one of the simplest components. The primary function is to provide guidance and clarity within an interface, ensuring users understand the purpose of other elements in the GUI.
- **Menus:**
  Menus are lists of options that users can select within the GUI [32]. They include:
  - **Menu Bars:** Menu bars typically control operations that affect the entire application but can also include actions for individually selected elements [58]. Generally they show all possible actions that can be performed within the GUI.
  - **Drop-Down menu:**
    A drop-down menu is activated by clicking on a menu title, causing a list of selectable options to appear in a small window right below it [61]. The menu temporarily overlays part of the screen and remains visible until the user makes a selection or clicks outside of it, causing it to collapse. Once the menu is expanded, the user can select an option, typically by clicking on the desired list item. See Figure 2.15 for a visualization.

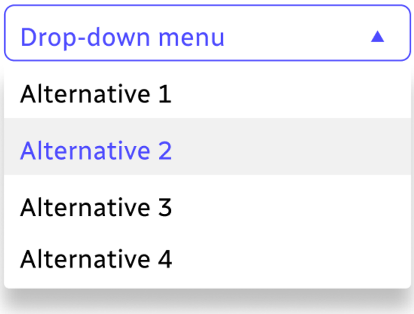


Figure 2.15: Representation of a drop-down menu appearance.

– **Pop-Up menu:**
Pop-Up menus can appear when right-clicking on a selected item or panel [58]. They can also appear when pressing an icon. They should be kept short and list the most common actions a user would want to perform. See Figure 2.16 for a visualization.

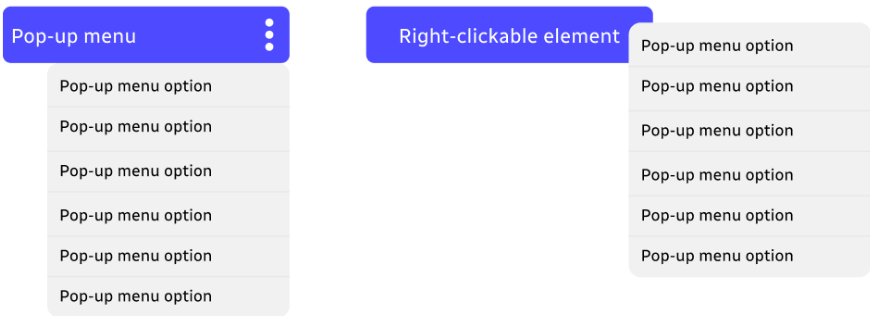


Figure 2.16: Two different representations of a drop-down menu appearance.

2.4.9 User Interactions

In GUI design, enabling user interaction is essential to make the interface functional. On desktop systems, interaction is most typically facilitated through a pointing device, with the mouse being the most common device [61]. The following outlines the most common interactions the user perform in GUI systems:

- **Pointing:** Objects can respond to being pointed at (hovered) [61]. When this occurs, the object may change visually, such as by highlighting or changing appearance, to indicate that it is interactive.
- **Clicking:** In general, this triggers some kind of action on the pointed object [61]. Depending on which button is used for clicking, different actions can be executed.
- **Drag and Drop:**
The drag-and-drop interaction involves clicking and holding the mouse button while moving an object to a different location on the screen [61]. Upon releasing the mouse button, the object is placed in the new location. Most often used to move an object or indicate an action with the selected object [58].

- **Accelerators:** Also known as keyboard shortcuts, which are key combinations that allow users to quickly access functions and perform tasks more efficiently [61]. These shortcuts, such as Ctrl+C for copy or Ctrl+S for save, are designed to streamline user interactions, reducing the need for mouse-based navigation.

2.4.10 User Feedback Mechanisms

To enhance the user experience in applications, providing timely and clear user feedback is essentials [61]. Feedback mechanisms help users understand the outcomes of their actions, reduce confusion, and guide them toward completing their tasks efficiently.

One common user feedback mechanism is dialogs. Dialogs are pop-up windows that provide information to the user and often require confirmation or input [61]. A typical example might be pressing "OK" to acknowledge a message. Dialogs come in various forms:

- **Alerts:** Mainly used to inform the user about events or outcomes, such as successful actions or warnings, keeping them updated on the system's state [61].
- **Error messages:** These should appear when the user performs an incorrect action. Effective error messages clearly explain the problem and guide the user toward a solution [58].
- **Success messages:** Providing positive feedback when user complete actions successfully can improve learnability and confidence among users [61].

Another user feedback mechanism is rich visual modeless feedback (RVMF), which passively informs about objects status or other attributes as a process indication [61]. They are displayed without any action. Some examples are:

- **Process indicator:** These visuals, such as progress bars or spinning icons, communicate that a process is underway [61]. This is especially important for processes that take longer than a few seconds, helping users understand that they need to wait. Ideally, it should also provide an estimate of the remaining time until completion.
- **Disable states:** Grayed-out elements or icons indicate that certain functions are currently unavailable [61].

Beyond dialogs and RVMF, other visual feedback play an important role in guiding user interactions:

- **Hover effects:** When holding the pointer over an object it can appear differently indicating that the component is hovered [61].
- **Hints:** Could for example be a pop-up text with a tip to perform the desired task [61]. Can be shown when the pointer is hovering a object for a while without performing an action.

3

Methods

3.1 Literature Study

A literature study was conducted to gain insight into the current availability of sandboxes within healthcare, including how they are designed and used. In addition, various fall detection solutions were examined to gain a better understanding of the available options and to support the creation of realistic fall detection simulation setups. The sources used were the search engines Google for an initial overview, followed by a focus on scientific search engines, IEEE Xplore Digital Library, PubMed and Elsevier ScienceDirect. Some keywords used for the search can be seen in Table 3.1. In some cases, search limits such as publication year, language were applied to narrow down the results and ensure relevance. Especially when the initial search returned a high number of records. Other searches, especially exploratory once where fewer results were expected or where the terminology was less standardized the search was kept broader without restrictions. Additionally, in several searches, quotation marks where used around multi-word phrases like "fall detection" to ensure exact phrase matching. This was important to avoid retrieving unrelated articles including the individual word "fall" and "detection" separately. However, not all searches included quotation marks, which may have led to broader but potentially noisier results. This was a deliberate trade-off during the exploratory phase of the literature study.

Database	Keywords	Limits (filter, limits, refine)	Number of records
IEEE Xplore	Sandbox	None	873
IEEE Xplore	"Fall detection" AND Wearable	year 2020-2025	527
IEEE Xplore	Sandbox AND "Test Environment"	None	8
IEEE Xplore	"Regulatory Sandbox"	None	7
IEEE Xplore	Digital test environment AND "fall detection"	None	21
IEEE Xplore	Sandbox AND "Digital Health"	None	1
PubMed	digital health "test environment"	English, year 2020-2025, Humans	5
PubMed	Sandbox	English	338
PubMed	Digital Health AND Sandbox	None	9
PubMed	Smartwatch "fall detection"	None	24
PubMed	"Fall detection" AND devices	None	139

Table 3.1: Keywords used in the literature study.

3.2 Simulation Case

A simulation case was created to represent a realistic and user-relevant scenario for potential simulation. The case was developed in consultation with Ulrika Björner, Business Developer in digitalization at the Elderly Care and Welfare Administration in Gothenburg, the project supervisor, and insights gained from the literature study on fall detection systems.

Once the case was defined, a flowchart illustrating the included components as well as the data transfer paths was created using Figma, a widely used tool for GUI design.

3.3 Design Process

The simulation case formed the basis of the design process, anchoring the design in a realistic scenario to better address the system context and user requirements. Based on the case, user requirements were identified, and GUI prototypes along with a simulation protocol were developed using an iterative HCD approach. This ensured a flexible, user-focused development process with continuous refinement throughout the project.

The workflow of the iterative process consisted of a planning period where tasks and goals was defined followed by development phase where tasks where implemented. Further the Testing and Feedback phase was entered and the interface was tested and feedback was collected. Lastly the Evaluation phase was entered evaluating if the solution met the user requirements or if further refinements are needed. The process is presented in Figure 3.1

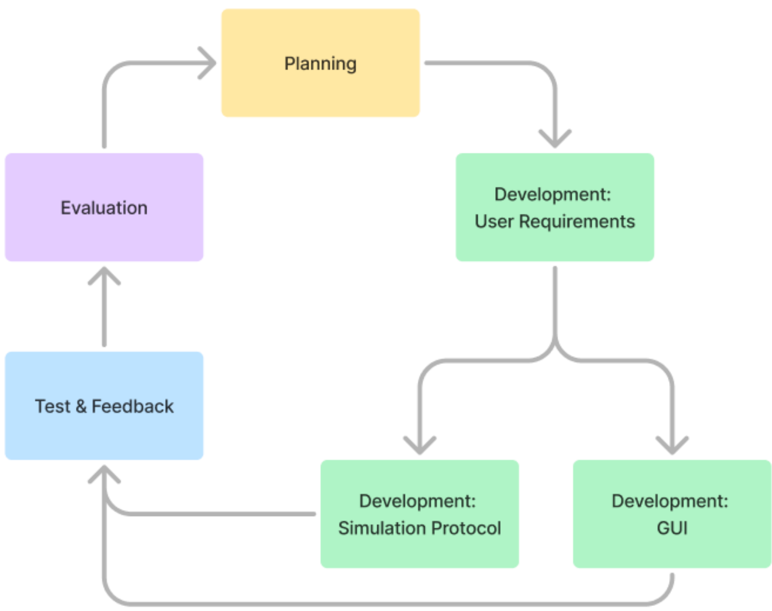


Figure 3.1: Workflow visualization for this project.

3.3.1 User Requirements

To generate user requirement specifications, the intended users were identified and characterized using personas. The personas were developed using the 10-step approach outlined in Figure 2.6 (Section 2.4.4).

Interviews severed as a foundation to refine and characterize the personas. For the technical user, three interviews was conducted with researchers from the Chalmers Digital Health group. The interviewees were: Mattias Seth, the supervisor of this project and a PhD student in signal processing and medical technology at the department of electrical eengineering. Anna Bakido and Fanny Apelgren, both also PhD students in the same field and department.

For the clinical user, two interviews were conducted. One with Peter Kelly, a physician and researcher in the field of digitalization in healthcare. The second with Robert Höglind, central operations developer at Sahlgrenska University Hospital in Gothenburg.

Based on the personas and insights from the interviews, a requirement specification was developed with measurable objectives. This specification was updated iteratively after each development iteration, incorporating feedback and evaluations to address gaps in user requirements.

3.3.2 GUI

To create a user-friendly GUI design for each user group, two different interfaces were developed in parallel. This approach aimed to meet the different user requirements and enhance usability.

To facilitate the prototyping of the GUI, a linear design process approach was utilized, as described in section 2.4.3. Throughout the design process, Norman's design principles were taken into consideration to ensure that the design is intuitive and easily comprehensible for the user. The prototyping was performed using Figma.

The linear design steps were integrated into the iterative HCD process to ensure that the best prototype from each stage was carried forward. The decision of which prototype that was considered the best was made trough usability testing. The performed test and deliverable after each project iteration is visualized in Table 3.2.

Iteration:	**Step**	**GUI**
Iteration 1	Delivery Test	Sketching & Wireframe Own Evaluation
Iteration 2	Delivery Test	Mockups A/B Testing
Iteration 3	Delivery Test	Interactivity Task Oriented Testing
Iteration 4	Delivery	Final Interactivity

Table 3.2: Overview of each iterations deliverable and test for the GUI.

As shown in Table 3.2, the result from sketching & wireframes were not tested on user groups, primarily due to the challenge associated with evaluating low fidelity prototypes and the limited number of test participants. Therefore, testing was postponed until the mockups were developed, ensuring that the evaluations would result in more valuable insights. Based on their perceived potential from own evaluations, three sketches along with their wireframes were chosen to be refined in the next stage. The mockups were assessed with help of A/B testing. Interactivity was assessed through a task-oriented test, where factors such as incorrect clicks and task duration were notated. All test used are described in more detail in section 2.4.7.1.

The number of participants for each test was determined using the rule of thumb outlined in section 2.4.7.3, resulting in five participants with relevant backgrounds for each test. For tests that applied to both user groups, five participants with each corresponding backgrounds were included to ensure a comprehensive evaluation. For example, the clinical interface where tested by ten users, since it could be of relevance for first time technical users. The participants on the test was chosen depending on their technical experience. Their professionals roles where engineers, technicians and clinicians. However, due to challenges in recruiting clinical users, some tests included only three or four participants with clinical backgrounds. Similarly for recruiting technical users it was challenging finding developer and technicians working with digital healthcare integrations. In these cases, other users with similar technical experience were used as test participants. This approach was deemed acceptable, as the rule of thumb was still met.

3.3.3 Simulation Result Protocol

From the user requirement specification the simulation result protocol was designed iteratively. First important components to include were identified. To get a better understanding of the potential users and where they would prefer information to be presented within the document a card sorting test was performed. Ten participants with varying backgrounds were included to gain broad and diverse insights. With gained insights four different sketches of the document structure was made and tested once again using an A/B test. In Table 3.3 the delivery from each iteration and corresponding test is presented.

Iteration:	**Step**	**Simulation Result Protocol**
Iteration 1	Delivery Test	Component identification Card Sorting Test
Iteration 2	Delivery Test	Sketch A/B Testing

Table 3.3: Overview of each iterations deliverable and test for simulation result protocol.

3.3.4 Project Iterations

An overview of the project iterations corresponding deliverables and tests of the project can be seen in table 3.4.

Iteration:	Step	User requirement	GUI	Simulation result Protocol
Iteration 1	Delivery	First draft	Sketching & Wireframe	Component identification
	Test	-	Own Evaluation	Card Sorting Test
Iteration 2	Delivery	Updated draft	Mockups	Sketch
	Test	-	A/B Testing	A/B Testing
Iteration 3	Delivery	Final	Interactivity	-
	Test	-	Task Oriented Testing	-
Iteration 4	Delivery	-	Final Interactivity	-

Table 3.4: Overview of iterations and deliverables.

3.3.5 Use of AI

In this project, OpenAI's ChatGPT (3.5 and 4o) has been utilized in two ways. First, AI-generated images were used in the GUIs to avoid using images of individuals' faces for privacy reasons and to prevent the promotion of specific brands. In figures where AI-generated images are used, the prompts used to generate them are provided in the figure caption. We have the right to publish these images and hold the copyright to them. Second, ChatGPT was employed to assist with language editing and grammatical corrections throughout the report, enhancing clarity and improving the overall reading experience. To ensure that the meaning of the text was not unintentionally altered, all AI-generated suggestions were carefully reviewed. In most cases, the suggestions were either modified afterwards or used as inspiration to improve the original text. Some prompts used is presented in the Table 3.5 below:

Prompts	Output
Create an image of a smartwatch. It should not resemble any typical brand.	Image
Create some images of elderly people that can be used as profile pictures.	Image
How can I improve this sentence to make it easier to interpret: X	Text answer
How can I make this sentence more scientific? X	Text answer

Table 3.5: Examples of prompts send to ChatGTP. (X stands for text that is to be improved)

4
Results

4.1 Literature Study

In this section the result of the literature study of digital health related sandboxes and fall detection devices is presented.

4.1.1 Existing Digital Health Sandboxes

There is limited extent of publicly accessible digital health-related sandboxes. Additionally, few sandboxes exists within the same domain as this project aimed to develop a GUI for. However several sandboxes do exist for digital health related purpose, such as EmSim. EmSim is a sandbox used for educational purpose within informatics which replicates common Electronic medical records (EMR) [73]. This sandbox is developed exclusively for educational purpose, emphasizing that sandboxes can have different purposes and applications. Another notable example is Epic on FHIR, a free resource that provides a sandbox for testing the implementation of standards critical for interoperability [74]. Epic on FHIR primarily targets developers and organizations working on EHR systems and aims to optimize these for better patient care. However, it is evident that this sandbox is designed for users with technical expertise, such as developers who are familiar with REST APIs and data formats like JSON, posing accessibility challenges for non-technical healthcare professionals.

In the UK, the Care Quality Commission have established several sandboxes [13]. One focusing on care from home, machine learning in diagnostic and screen services and digital triage in health services. UK Information Commissioner's Office offers a regulatory sandbox to support all organizations, including those in healthcare, in developing products and services handling personal safety. The UK Medicines and Healthcare products Regulatory Agency has their own regulatory sandbox, "AI-Airlock", providing developers in generating evidence for AI technologies used in healthcare.

Further, countries like the United States, India, and Singapore offer various regulatory and testing sandboxes aimed at digital health solutions [13]. For example, the Massachusetts eHealth Institute runs a sandbox challenge program to support and identify digital health companies helping them expand their user base and their products. Similarly, the National Digital Health Mission in India provides a sandbox for organizations aiming to integrate their health innovations into the national digital health system. In Singapore, the Center for Healthcare Innovation has launched a sandbox called the Center for Healthcare Start-up Enterprise Link (CHISEL) [75].

This initiative aim to help companies and startups to try their health-tech solutions but also reach out to caregivers. They have different focuses for different years, and for example, in 2024, the theme was: "Strengthening Care for a Healthier and Resilient Ageing Society". This project primarily focuses on matching companies with promising ideas to relevant healthcare institutions, where simulations and/or clinical test-bedding will be carried out.

An emerging trend in digital health sandboxes appears to be the focus on interoperability and AI-regulation. The Global Patient co-Owned Cloud (GPOC) provides technical sandboxes focused on integrating standardized medical classification systems (e.g., SNOMED and ICD-11) while ensuring cross-border compliance with healthcare regulations [16].

Additionaly, Stanford Biodesign Digital Health has developed an open-source framework called Spezi for assisting the development of interoperable applications within digital health [76]. It is built on the FHIR standard. The platform is not called a sandbox but gives researchers and developers an opportunity to develop and experiment with healthcare applications in a controlled but flexible environment. Spezi provides a modular, pug-and-play ecosystem enabling seamless integration with EHRs, AI-driven analytics and wearable devices. EU is also set to provide regulatory sandboxes for AI testing by 2026, aiming to ensure compliance with AI regulations, promote innovation, and contribute to informed regulatory decision-making [24].

Moreover, some sandboxes offer open access, such as CDS Hooks sandbox available online [53]. This is easy to access, and there is possible to change things directly online see how potential CDS cards could appear. It also provides a view showing the code, emphasizing the need to understand coding. However, there is no need for coding to test out how it looks when, for example, you receive a card with clinical decision support when prescribing medication.

A recurring theme in the reviewed sandboxes is that they are mainly designed for technical users, particularly developers and IT professionals. This presents a significant barrier for clinical users who are not familiar with programming hindering their participation in the development process. The need for sandboxes that are more inclusive, suited for non-technical users, is critical for future developments in the digital health field.

Some key insights from this literature review include that most available sandboxes typically require some programming skills. For example, users need to download Git, a tool used to track code changes and collaborate on projects, and understand how to manage it [77]. The Epic on FHIR sandbox requires users to build their own applications and have knowledge of REST APIs and data formats such as JSON. Of the sandboxes that were accessible, most were aimed at individuals with technical skills, rather than clinical staff. This highlights a clear gap in sandboxes that are better suited to clinical users without programming knowledge.

To summarize, there are several different types of sandboxes within the field of digital healthcare and the healthcare sector, but few are publicly available, and detailed information about them is limited. As a result, it is difficult to gain in-depth knowledge and fully understand the current state of the landscape of sandboxes. Furthermore, no existing sandbox closely resembles the ASAP sandbox, designed to test full level systems in healthcare. Additionally, the lack of an official definition for the term sandbox, further complicates the identification of relevant projects and research for this topic. Many projects may not explicitly use the term sandbox, further complicating the search for appropriate examples.

4.1.2 Fall Detection Systems

There are various systems available to support elderly individuals at risk of falling. One such system is the personal emergency response (PERS), where the patient manually triggers an alarm by pressing a button to contact an alarm center [78]. However, studies have shown that these alarms have drawbacks as they are useless if the individual is unconscious after a fall or unable to reach the button. One study showed that 80 % of the users did not call for help using the alarm after experiencing a fall while home alone and unable to get up by themselves [79]. Despite this, PERS devices are currently used in the Västra Götalands Region (VGR) as fall alarms among elderly [80]. When the alarm is triggered, Trygghetsjouren, the alarm center for elderly in Gothenburg, is contacted. They call the individual and, if needed, dispatch home care services to assist them.

In contrast, automated fall detection systems can detect falls and alert caregiver or alarm center without requiring any user interaction [81]. These are of interest to implement in the elder care in VGR due to promising benefits for patients [80]. In 2022, VGR ran a project to test the implementation of a smartwatch with fall detection capabilities for elderly that would alert Trygghetsjouren automatically once a fall was detected.

There are two main categories of fall detection system approaches: wearable and non-wearable [81]. The wearable systems use embedded sensors, commonly detecting motion and location of the users body and processing data from one or multiple sensors for detecting a fall [82]. The most common sensors of wearable devices are accelerometers, gyroscopes and barometers [83]. Wearables can be positioned or placed on many different places of the users body [78]. Forearm, wrist, neck, waist, ankle, inside a shoe to mention a few. The typical use of the sensor data is that it is fed into an algorithm deciding wether a fall has occurred.

Wearables can easily detect a fall on the acceleration or impact but has a drawback as they rely on the user remembering and choosing tho wear it [78]. Something that can cause elderly people not using the devices as intended.

The non-wearable systems have embedded sensing systems, commonly camera, radar or other motion detectors [78]. These are typically installed in one or more rooms where the user frequently moves or is at higher risk of falling.

The primary benefit of non-wearable systems is that they do not rely on the user remembering to wear the device. However, they raise privacy concerns instead, especially those that use cameras, which are often viewed as intrusive [78]. Radar-based systems are generally less intrusive, though users may still have concerns about surveillance.

Some examples of wearable systems that exist on the market and in research today are:

- **SafeTRX:**
  Developed by 8 West, SafeTrx Care combines Sony's mSafety wearable with Infonomy's Stumbelometer leg sensor to detect falls in real time [84]. SafeTrx have a branch called SafeTrx Care, which focuses on healthcare monitoring for elderly individuals. The Stumbelometer detects a fall and sends a signal to the mSafety wearable, which then alerts caregivers and emergency contacts. The mSaftey wearable and the Stumbelometer has been tested within the VGR in an evaluation project for fall detection system in 2022 [80]. The mSafety wearable itself has four in-device sensors: accelerometer, GPS, gyroscope and photoplethysmogram. Additionally it has algorithms for fall detection, heart rate variability and positioning among others [85]. The device uses LTE Cat-M1 to connect with the system and transmits data using REST APIs directly from the device. In the setup used in the VGR project, a fall triggered an alarm including time stamp and body position [86]. If the user would fall outside the GPS position was included in the alarm. Body position and geographic position was continuously updated as they change.
- **SOS Micro:** Developed by Bay Alarm Medical, this compact device offers both automatic fall detection and manual emergency alerts [87]. It includes GPS tracking and can be worn on the wrist, around the neck, or clipped to a belt. When a fall is detected, it contacts the company's emergency center, where an agent can communicate with the user. If needed, the agent can alert family members or emergency services.
- **EPA Pendant:** Essence Smartcare has developed multiple sensors to their Care@Home platform for monitoring both indoor and outdoor [88]. They offer an automatic fall detection weareable, EPA Pendant with fall detection using an accelerometer and a gyroscope sensor.

Some non-wearable systems that exist on the market and within research today are:

- **MDsense:**
  Essence SmartCare offers a fall detection device called MDsense. It utilizes radar technology for detecting falls and body position of monitored individual [89]. The MDsense can be installed in different environments including independent living residents or in senior care facilities.
- **Kami Fall Detect:**
  Developed by Kami Vision, this AI-powered system uses cameras to detect falls in real time in the homes of elderly individuals [90]. It sends instant SMS alerts to relatives and allows two-way communication with family members to check if everything is okay. If no response is received from family members, the system can automatically contact emergency services via RapidSOS.

Integrating fall detection systems into IoMT systems presents a promising path for advancing healthcare and personal safety significantly [91]. IoMT integration enables monitoring and assistant of vulnerable individuals on a new level. Making it possible for fall detection devices to communicate with other systems.

The literature study indicates that while there are several fall detection solutions on the market, no autonomous detection system is currently implemented in VGR. However, evaluation and implementation studies have been conducted on some devices, such as the SafeTRX wearable to be integrated in their existing system for elderly care infrastructure. Given the limitations of the systems, there are suggestions of using multiple sensors to account for weaknesses in the different devices [78]. One of the main challenges remains the integration of these technologies into existing healthcare frameworks. As such, simulation tools and test environments play a critical role in evaluating feasibility, effectiveness, and user acceptance before large-scale real-world deployment.

4.2 Case

The selected case for developing the GUI involved all the main components of the ASAP sandbox: providers, services, consumers, and the scenario engine. In this setup, three providers, one service, and one consumer were used. The scenario engine orchestrates the interactions between these components, managing their connections and data exchange. The included components are:

- Providers:
  - Smartwatch
  - Radar Camera
  - EHR
- Services:
  - Cloud Service - including data processing algorithm
- Consumers:
  - Alarm Center Interface

The case is presented in Figure 4.1, where providers are represented by blue boxes and the data types used from the corresponding device in green. The service is represented in yellow and it is embedded algorithm shown in gray. Lastly the consumer is visualized in pink. Direct signals sent between the different components are represented with red arrows. When a data request is initiated, it is shown with yellow arrows, while the corresponding response is depicted with blue dashed lines.

The simulation scenario begins with a fall alert sent from the smartwatch. This alert includes a time stamp, body position and GPS coordinates and is transmitted to the cloud service. Upon receiving the alert, the service requests additional data from both the smartwatch, EHR and radar camera. The specific data requested can be seen in Figure 4.1. Once all relevant data is collected, the algorithm within the cloud service processes the information to decide wether to alert the alarm center or classify the event as a false alarm.

This case enables testing of interoperability between multiple systems, evaluation of algorithm performance under defined conditions, and assessment of patient outcomes.

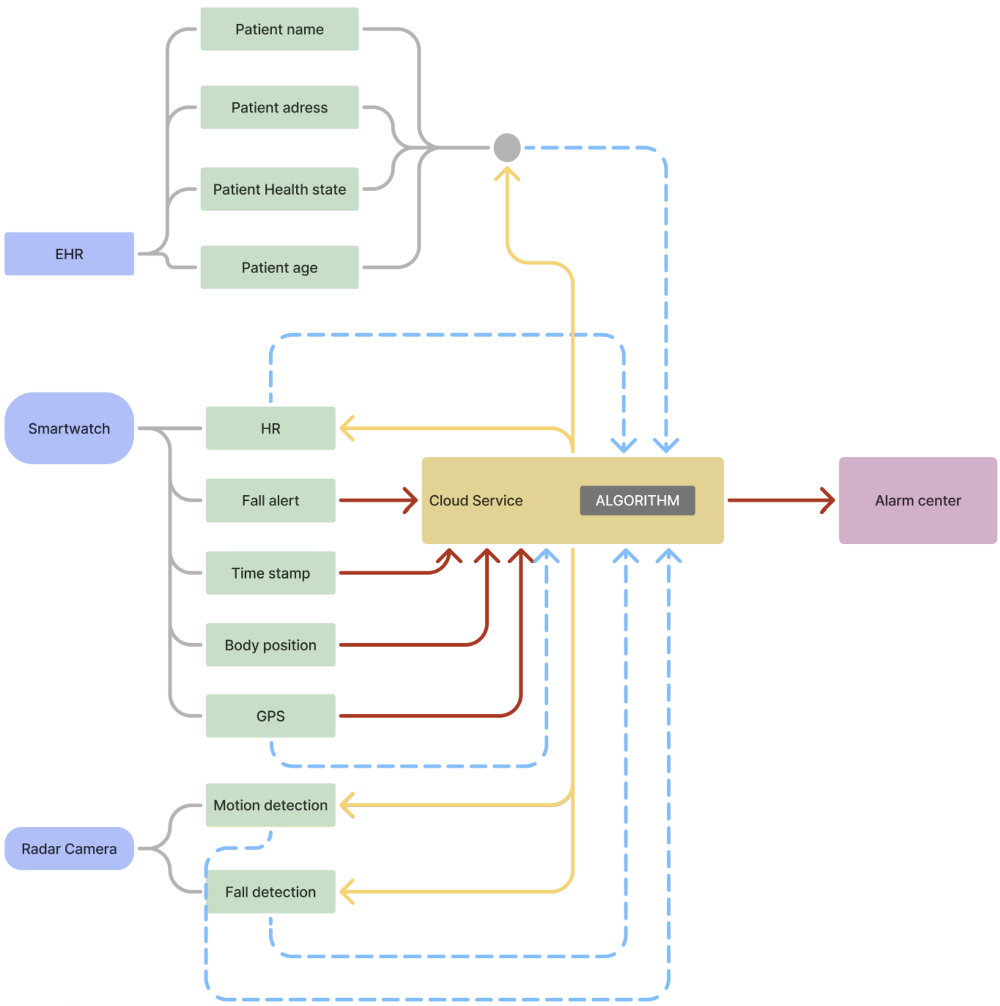


Figure 4.1: The case representing the design simulations setup.

A simplified version of the case including the fall event can be seen in figure. 4.2.

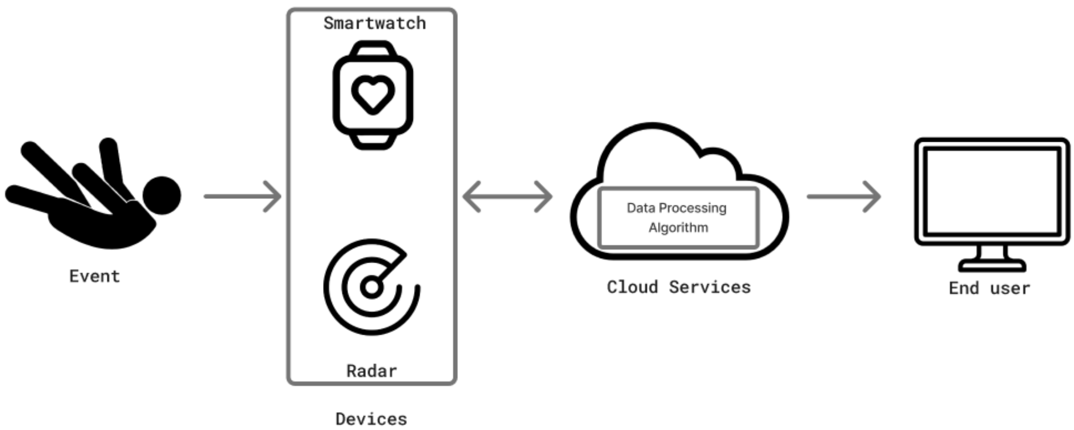


Figure 4.2: Case representing the design simulations setup.

4.3 User Requirement

The interface requirements for the sandbox simulation setup were defined for both technical and clinical user groups and are summarized in Table 4.1. While some requirements were shared across both groups, others were specific to either the technical or clinical users. Each requirement is marked to indicate its relevance for the respective user group: ’T’ for technical users and ’C’ for clinical users. The table indicates applicability to each user group using an ’x’, while a dash (’–’) denotes non-applicability. Additionally, the most critical requirements, those considered essential for functionality are marked in the column ’Must Haves"

The categories are structured around key functionalities: general simulation setup, component management, communication configuration, execution, collaboration, and user assistance. This structured approach ensures that the system can accommodate a broad range of use cases, from in-depth algorithm testing and system interoperability analysis to clinically relevant patient-centered simulations.

Category	Requirements	T	C	Must Haves
General setup requirements	• Configure a sequential simulation flow	x	x	-
	• Create or select a simulation case	x	x	x
	• Repeat simulations with small variations (e.g., modifying one component or parameter)	x	x	-
	• Create a simulation from a product/component-centered perspective	x	-	-
	• Create a simulation from a patient-centered perspective	-	x	-
Component Management	• Add components	x	x	x
	• Select multiple components	x	x	x
	• Remove components	x	x	x
	• Duplicate components	x	x	-
	• Change a component while maintaining existing communication paths	x	x	-
	• Display detailed information about the component	x	x	x
	• Change component settings	x	x	x
	• Visually illustrate all added components	-	x	x
	For Provider Components:			
	• Select source of data: synthetic or real-world.	x	x	x

Table 4.1: Technical and clinical requirements for simulation setup and component management, where ’T’ represents technical and ’C’ represents clinical.

	• View available and select relevant data elements	x	x	x
	• View available data formats	x	-	x
	• Open data files for inspection	x	-	-
	• View the data transfer format	x	-	-
	For Service Components:			
	• Select how to process input data	x	x	x
	• Select a data processing algorithm, if applicable	x	x	x
	• Select data pre-processing methods	x	-	-
	• View details of the processing methods	x	-	-
	• View input data requirements for algorithm (format, time constraints, etc.)	x	-	x
	• View data mapping issues, including both format and semantic interoperability problems	x	-	x
	• Receive suggestions of data mapping	x	-	-
	For Consumer Components:			
	• Select data to present and how it is shown in the consumer interface	x	x	x
	• View input data requirements (format, time constraints, etc.)	x	-	-
	• Select method for visualizing data in consumer interface	x	-	-
	• View and modify the underlying code for data input configuration for consumer interface	x	-	-
Communication Setup				
	• Add and remove communication paths between components	x	x	x
	• Only allow the creation of valid communication paths	x	x	-
	• View and select what data is transferred	x	x	x
	• Set the time interval for data transfer	x	x	x
	• Select and view the type of communication (e.g., push, request, or response) between components	x	-	x
	• Display the timeline of communication	x	-	x
Save & Collaboration				
	• Save simulations for future modifications	x	x	-
	• Share simulations with other users	x	x	-
	• Save component-specific settings	x	x	-

Table 4.1: Technical and clinical requirements for simulation setup and component management, where 'T' represents technical and 'C' represents clinical.

Simulation Execution				
	• Be able to start a simulation	x	x	x
	• Prevent simulation launch on critical errors	x	x	-
	• Receive a clear indication that the simulation is in progress	x	x	x
	• View time remaining until completion.	x	x	-
	• Access and export the simulation result protocol	x	x	x
User Assistance				
	• Offer non-intrusive and context-aware user assistance	x	x	-
	• Ensure that hints are easily accessible when needed	x	x	-
	• Allow users to enable or disable hints and help as desired	x	x	-
Patient Management				
	• Select a pre-defined patient profile from an existing library	-	x	-
	• Create a custom patient persona, by specifying or generating desired information	-	x	x

Table 4.1: Technical and clinical requirements for simulation setup and component management, where 'T' represents technical and 'C' represents clinical.

4.3.1 Technical User

Insights into the needs of technical users were gathered through three interviews with researchers from the Chalmers Digital Health group. These interviews revealed a broad range of design considerations needed to accommodate researchers with diverse goals. Depending on the specific use case, different levels of detail are required. One participant, Mattias Seth, highlighted the importance of using the sandbox for evaluating and simulating system interoperability at higher interoperability levels. Another participant, Anna Bakidou, emphasized the importance of using the environment to test and validate algorithmic performance, including identifying and resolving unexpected outcomes. Fanny Apelgren focused on the value of the sandbox for comparing outcomes of different system configurations and user groups as well as generating synthetic patient data based on limited parameter input to support data-driven simulation scenarios. All participants recognized the sandbox environment as a valuable tool for research and development within digital health, offering opportunities to explore interoperability, validate systems, and experiment with various digital setups in a safe and flexible setting.

Based on the interviews, the typical technical user is characterized as a researcher specializing in digital health solutions, possessing advanced technical skills and a comprehensive understanding of interoperability both within and across systems. A representative persona is illustrated in Figure 4.3. The persona represents a user engaged in research and development activities in the field of digital health, with a particular focus on system integration, interoperability and algorithm development. This user sees great potential of using synthetic datasets to simulate realistic patient scenarios, validate system behavior, and evaluate algorithmic decision-making. Due to the complexity of these tasks, the interface designed for this user group must support advanced functionalities, such as detailed data configuration, specification of data transfer parameters, visualization of semantic mismatches, and transparency in how algorithmic decisions are derived.

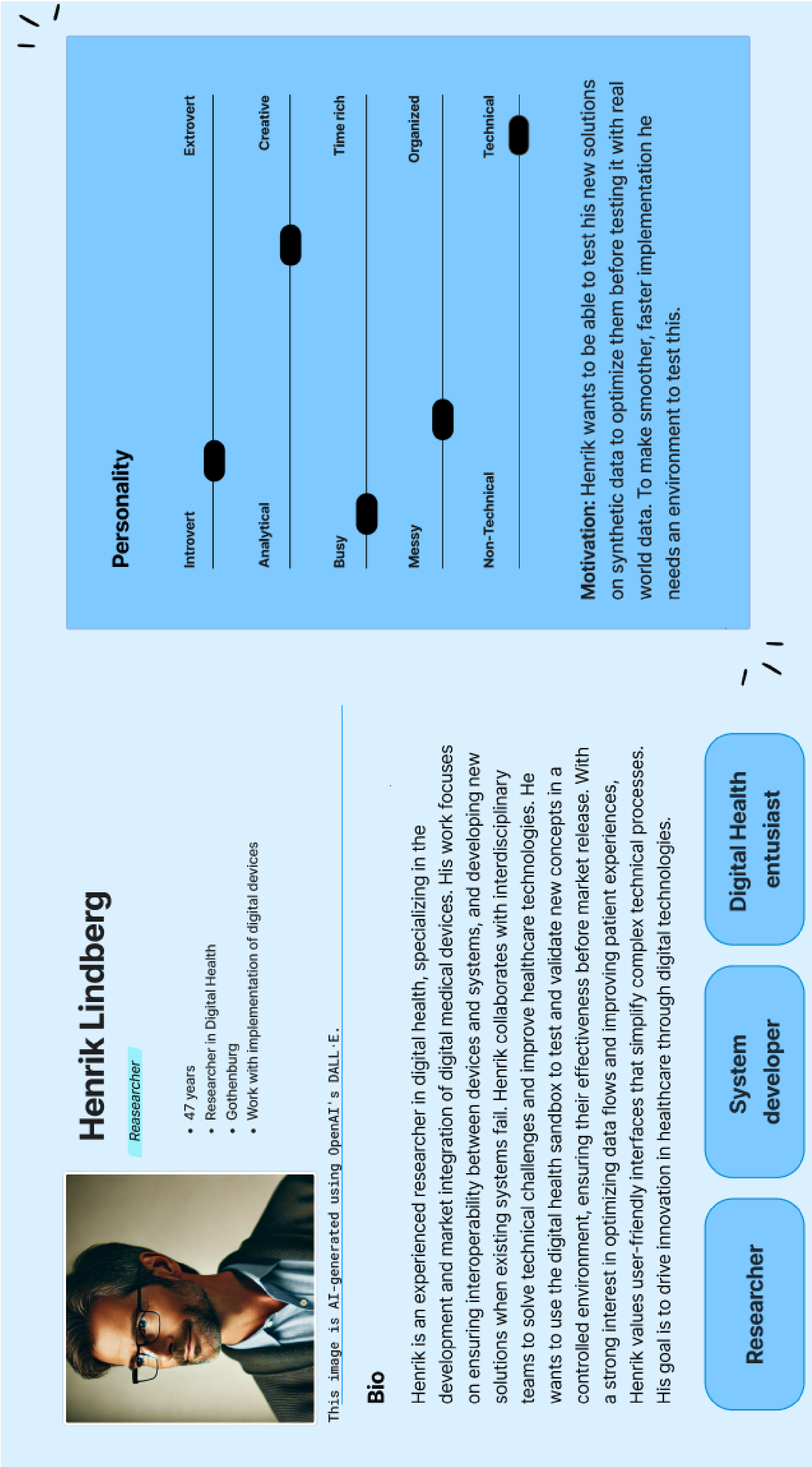


Figure 4.3: A representation of a user with technical background. (The portrait of the persona generated with ChatGPT using the prompt "persona image for engineer", 2025).

4.3.2 Clinical User:

The needs of clinical users were assessed through two interviews with individuals representing this group. These discussions underscored the importance of simplicity and clarity in interface design. One of the most significant findings came from the interview with Robert Höglind, who highlighted the importance of adopting a patient-centered perspective. The patient is the primary focus of healthcare professionals, therefore, it is most relevant to perform simulations from this viewpoint. Höglind also mentioned that other professions in the healthcare production chain, such as managers, may have a product-oriented approach. Healthcare itself has traditionally been production-oriented rather than patient-centered, but a shift is occurring, nowadays there is more emphasis on the patient journey. When considering technical experience, the level of base comfort with technology is relatively high in modern society, and people within the healthcare sector can be considered to have the same ability to use technology as the average person. The most important factor is to have a high degree of familiarity or recognition.

The other interviewee Peter Kelly also pushed for a patient-centered approach to simulations. Since more of the care is moving home to the patient, there will be a greater demand to monitor and evaluate patient data. He wanted to be able to simulate patient outcomes depending on the choices of devices to monitor to see what gave the best result for that patient. Both Kelly and Höglind noted that it is currently unclear who the primary users of the sandbox could be. However, there is potential for a broad user base. Clinicians may use the tool for decision support or to gain a better understanding of medical device functionality, while individuals involved in development or procurement could use it to test the integration of new technologies with existing systems. Kelly mentioned that large procurement processes often involve specialized clinical staff. Although this represents a relatively small user group, it includes both clinical expertise and organizational or technical perspectives, highlighting the diverse roles that may interact with the sandbox in the future.

With this in consideration, the resulting user is described as doctor working with elderly care as well as procurement. The persona are presented in Figure 4.4. This persona represents a user working in the medical field, specializing in elderly care, with dual responsibilities in both clinical practice and medical device procurement. The user is interested in evaluating how well specific devices suit individual patients, as well as determining which technologies should be offered to the elderly population. Due to a demanding workload and moderate technical expertise, the user values an interface that allows for quick and easy simulation of device use in realistic scenarios with a patient centered perspective.

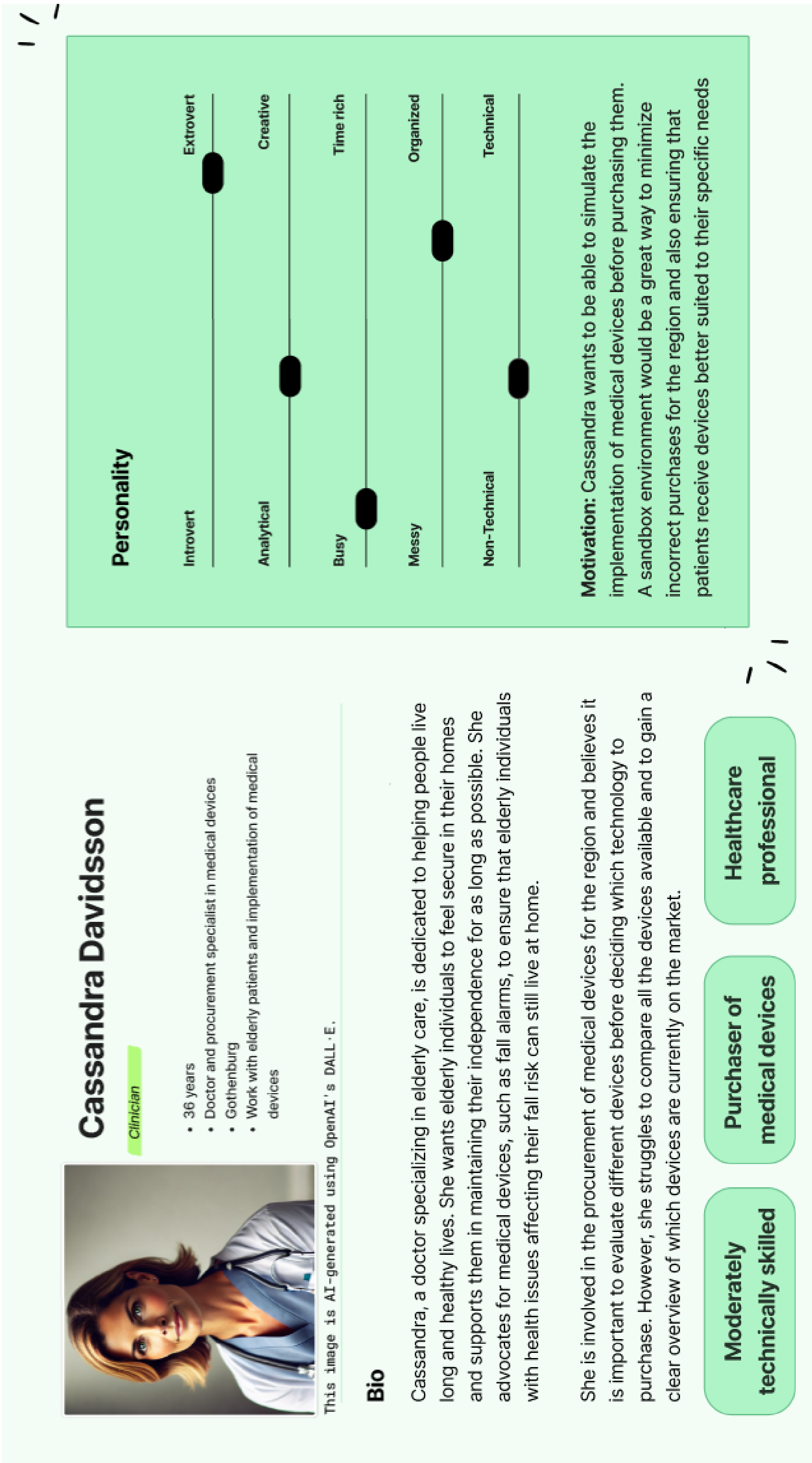


Figure 4.4: A representation of a user with clinical background. (The portrait of the persona generated with ChatGPT using the prompt "persona image for doctor", 2025).

4.3.2.1 Simulation Result Protocol Requirements

For the simulation result protocol, there are requirements for understanding the type of simulation performed, as well as the data types, formats, other technical details used, and if any errors occurred. A simple overview as well as advanced descriptions of the technical issues and potential solutions is required to match the needs of all intended users. These requirements are summarized as follows in Table 4.2.

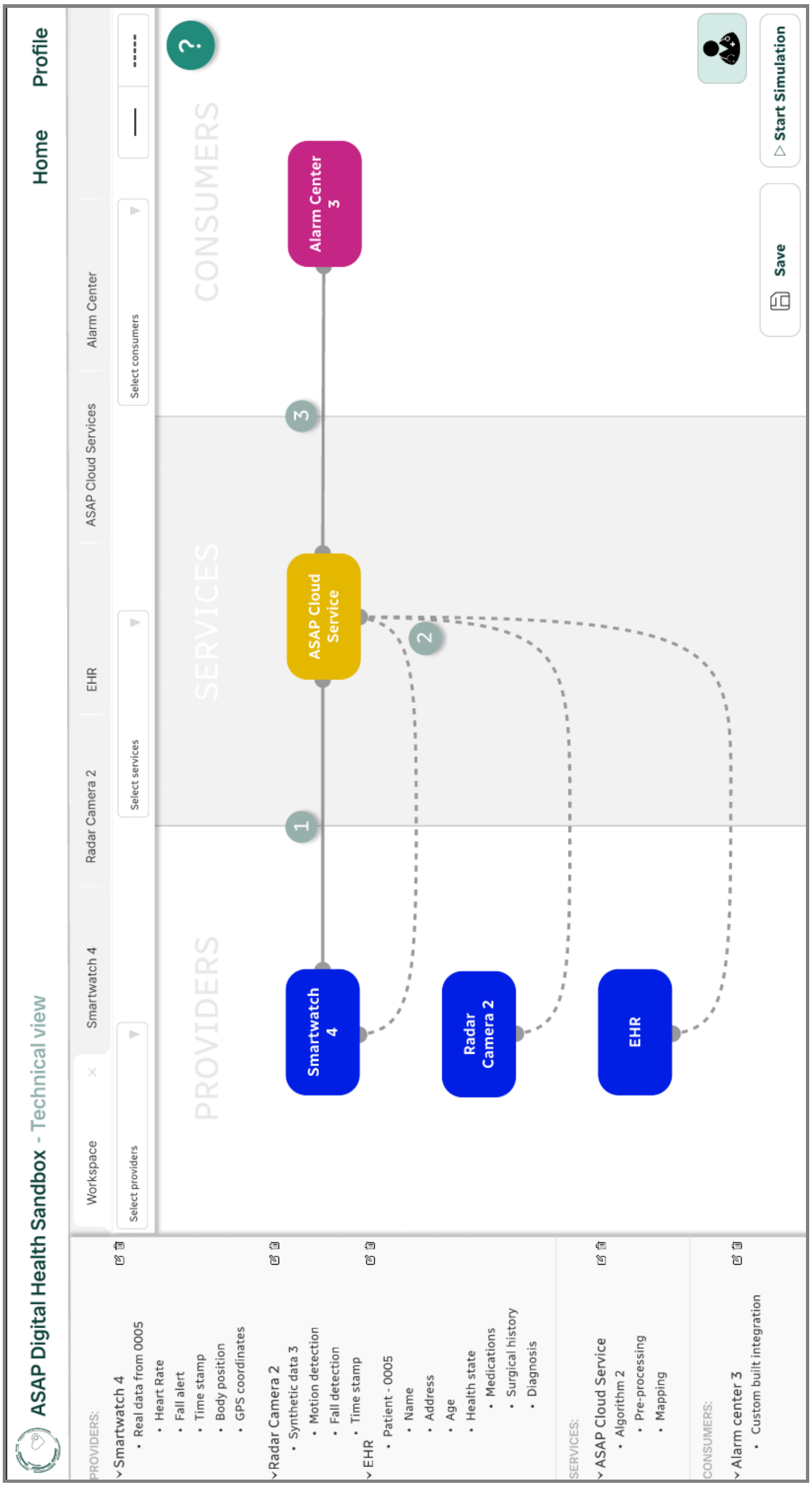
Category	Requirements
Protocol Overview	• Understand the purpose and the type of document • Identify simulation executor • View the date the simulation was performed • Share the result externally without requiring access to the sandbox environment • Download the result as a standalone document • Identify the sandbox system version used for the performed simulation
Simulation details	• Understand the type of simulation performed • Identify components included in simulation • Understand components communication paths • View what data were included and transferred • View format used for data sharing • View data transfer time intervals • Display the timeline of communication
Simulation Results	• Identify problems or errors that occurred during simulation • Receive suggestions for resolving identified problems • Understand the significance of the problems • View algorithmic decisions and understand their justification • Get important information about the set up, that not necessarily is problematic for the specific simulation
Validation and Annotations	• Verify legitimacy of the document • Add comments or annotations to the document

Table 4.2: Simulation result protocol requirements.

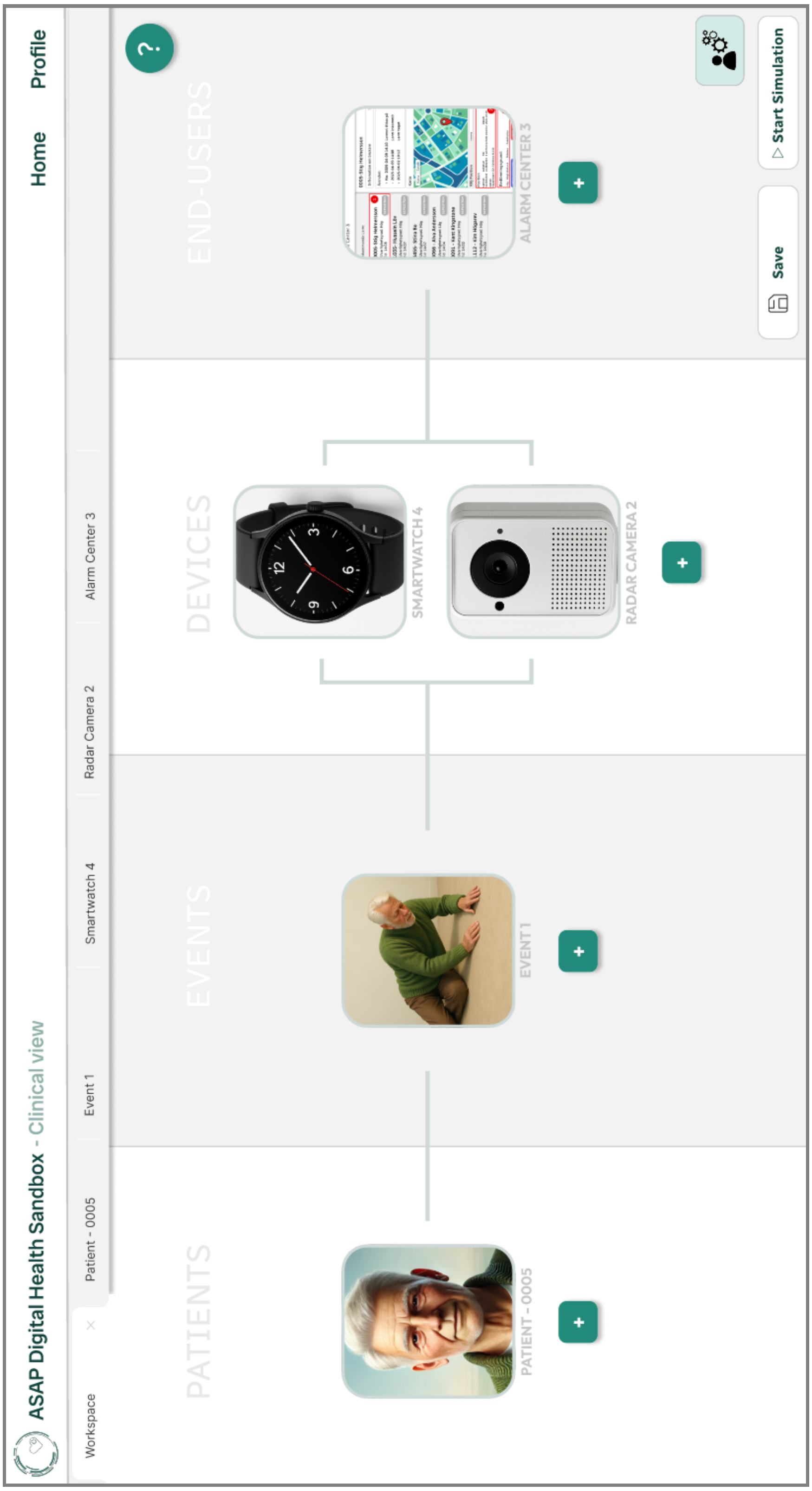
4.4 GUI Design

This section presents an overview of the finalized interface and highlights the key design decisions that shaped its development. A more detailed description of the design process can be found in Appendix A and B.

An overview of the final GUI is presented in Figure 4.5. Since the prototypes were developed in Figma, they are partially functional, allowing interaction with menus, buttons, and other interactive elements of the prototype. Making it possible to set up the simulation case presented in Section 4.2. The interface includes two views that users can switch between, depending on their preferred level of detail. Figure 4.5a shows the technical view of the sandbox, offering more advanced settings and detailed control over the simulation. Figure 4.5b presents the clinical view, designed for a simpler and more intuitive simulation setup. Based on the interviews conducted, it was concluded that the ability to easily access both views was highly desired. Each version is tailored to best match the needs and preferences of its intended user group.



(a) Technical view.



(b) Clinical view (Patient and device images generated with ChatGPT using the prompt "elderly person, profile view", "elderly person falling, neutral background" "smartwatch, neutral background" and "radar camera, neutral background", 2025).

Figure 4.5: Final GUI with completed simulation setup in the technical and clinical view.

The views are designed to maintain a consistent structure wherever possible, ensuring that users can easily recognize the different sections, thereby minimizing cognitive load. Both views include a workspace for setting up the simulation. The workspace refers to the main interactive area within the GUI where users build and configure their simulation setups. It provides an overview of the simulation flow. The main common elements shared between the two views are presented in Table 4.3, including functioning, design rationale, and references to the specific tests from which the results were derived.

GUI Element	**Function**	**Design Rationale**	**Derived from**
Tabs	A navigation bar is located at the top of the page, where tabs appear dynamically as components are added.	Recognizable structure that simplifies navigation between components without requiring a return to the main workspace.	A/B testing: Clinical User (Appendix B)
Workspace with sections	Square-shaped sections for each component type are located in the workspace, with labels indicating the specific component they correspond to.	Makes it easier to identify where each component should appear. It is preferred over an open workspace, as it simplifies understanding of where to focus and facilitates alignment between different views.	A/B testing: Clinical and Technical (Appendix A and B)
Main function buttons	Start Simulation, Save, Switch view - buttons.	Simple to find and interact with. Mainly based on discoverability and feedback, but also other principles from Norman's design concepts.	-
Visual representation of components	Images or blocks representing components in workspace.	Help user to understand that the components have been added and makes them easily recognize understandable.	A/B testing (Appendix A and B)
Clickable components	Clicking on components leads to the the corresponding tab.	Clicking on components is intuitive for users wanting to access them directly. Consistent navigation supports smoother switching between views.	A/B testing (Appendix A and B)

Table 4.3: Common GUI elements for technical and clinical view.

4.4.1 Technical View

The technical view provides a more advanced interface with greater flexibility and a broader range of settings for the components and set up. It is structured into several tabs, each with a distinct layout tailored to its specific characteristics and functionalities. The interface includes different tabs for Provider, Service, and Consumer, as well as a Workspace tab. The workspace allows users to add components in any desired order. A visualization of how the website flow can be utilized for this projects specific case is shown in Figure 4.6. First, the provider components are added and then edited in their respective tab (shown in blue). Next, a service component is added (shown in yellow), configured, and connected to the provider components. Finally, the consumer component is added and edited (shown in purple), and a connection is established to the service component.

The following sections outlines all the tabs available in the Technical View.

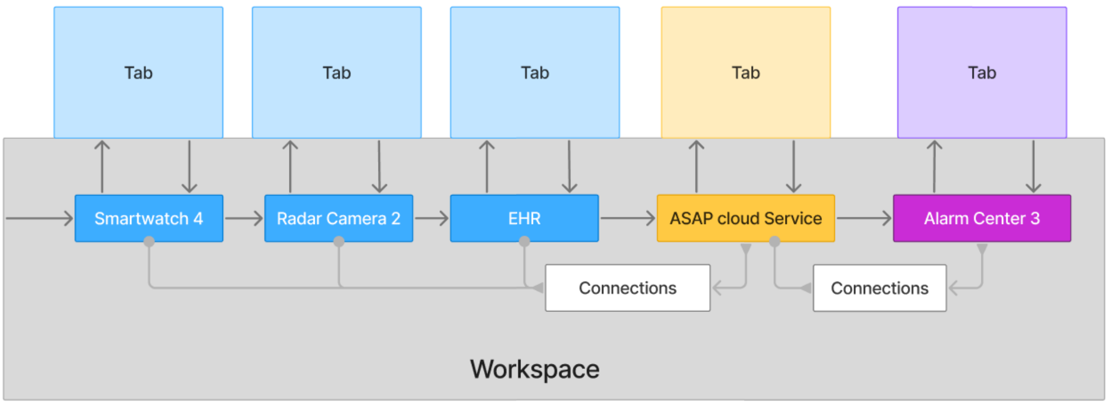
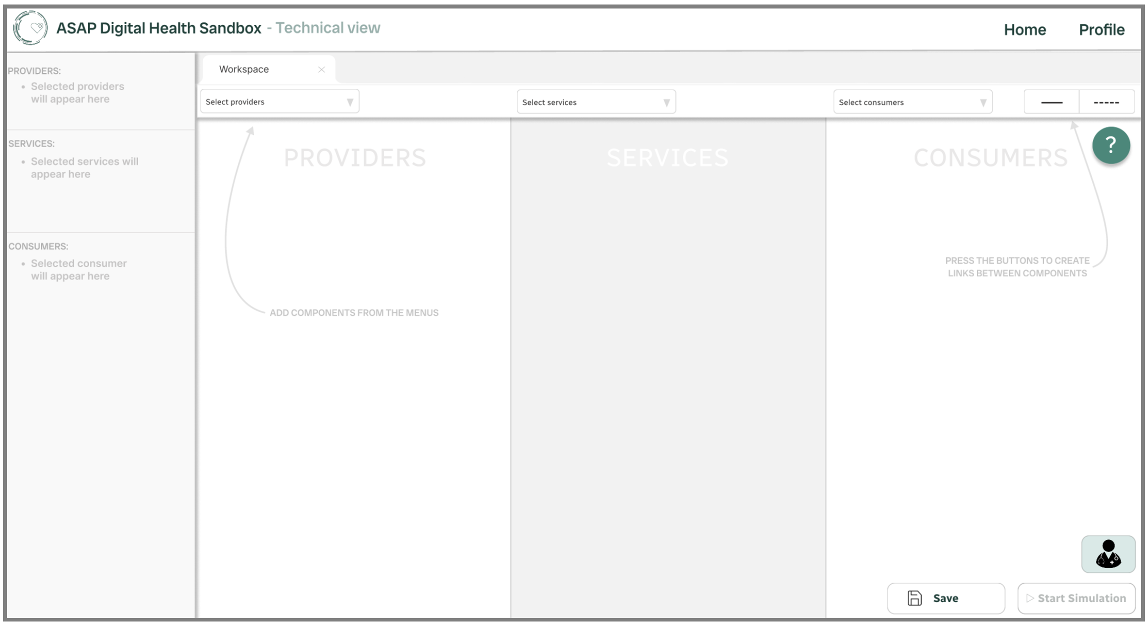


Figure 4.6: Visualization of navigation flow in technical view.

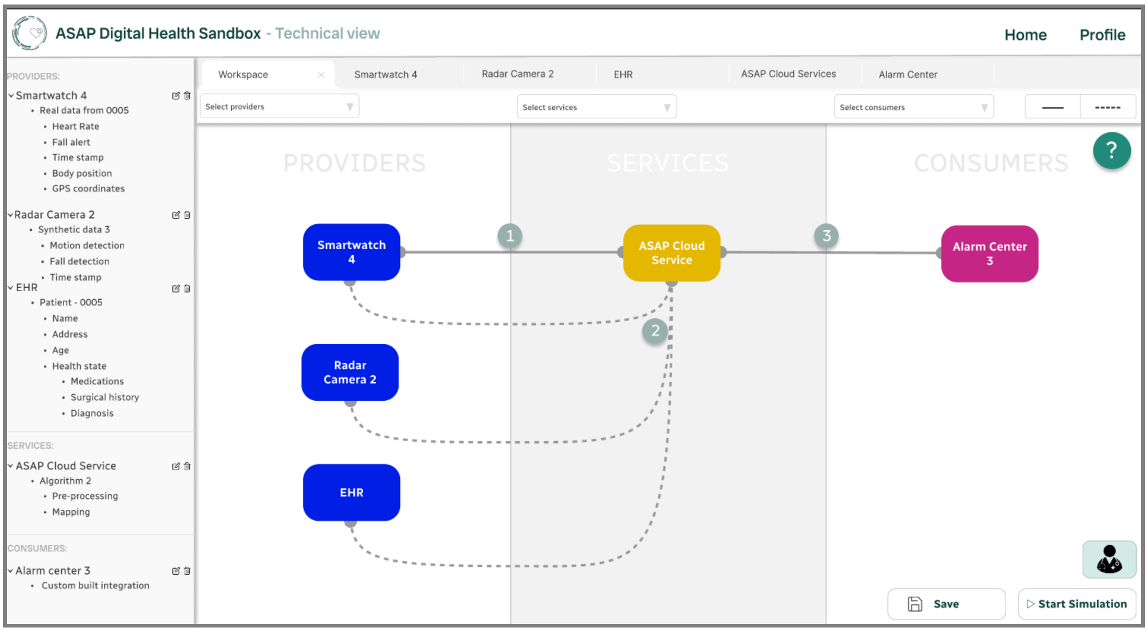
4.4.1.1 Workspace Tab

Figure 4.7 shows the workspace of the technical view, both at start up and after a completed simulation set up. This layout consists of four main elements:

- A workspace divided into three sections.
- Color-coded components.
- A side menu located on the left.
- A top menu bar positioned below the tabs.



(a) Start of technical view with empty workspace.



(b) Completed simulation setup in technical view.

Figure 4.7: Technical view: start and completed simulation setup.

4.4.1.1.1 Workspace and Components The workspace is divided into three sections: Providers, Services, and Consumers, see Figure 4.7a. Selected components appear in the corresponding section as building blocks. Each block is color-coded to indicate its component type. The specific colors for each component are presented in Table 4.4. The colors were selected with to avoid distraction or confusion for the user, while still signaling interactivity, see Figure 4.7b.

Component	Color
Providers	Blue
Services	Yellow
Consumers	Pink

Table 4.4: Component color scheme in the Technical View.

Hovering over a component displays directional arrows, indicating how connections to other components can be made, shown in Figure 4.8. The right and left arrows represent data being pushed or received, while the downward arrow represents a data request/response. Clicking an arrow opens a menu that visualizes the available push or request destinations.

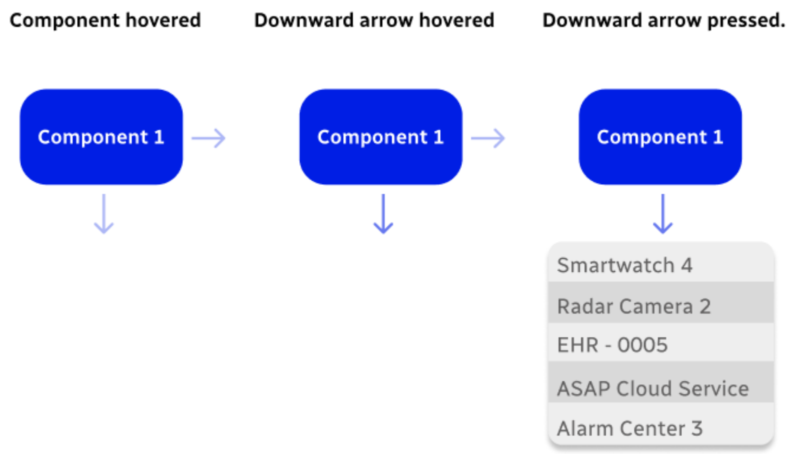


Figure 4.8: Visualization of component functionality. Component and component arrows are hovered to demonstrate how connections can be made.

4.4.1.1.2 Side Menu and Top Menu Bar The side menu contains all added components, organized into the categories: Providers, Services, and Consumers. See the red circle labeled (1) in Figure 4.9 for the empty side menu, and in Figure 4.7b for the filled side menu. The items in the side menu are clickable, redirecting the user to the corresponding configuration tab upon selection. When a component is added, it appears in the side menu, and when edited, the selected data and data types are displayed below. The side menu includes edit and remove icons for each component, enabling more intuitive modifications. Additionally, the side menu can be collapsed, allowing the workspace to expand and occupy the entire page.

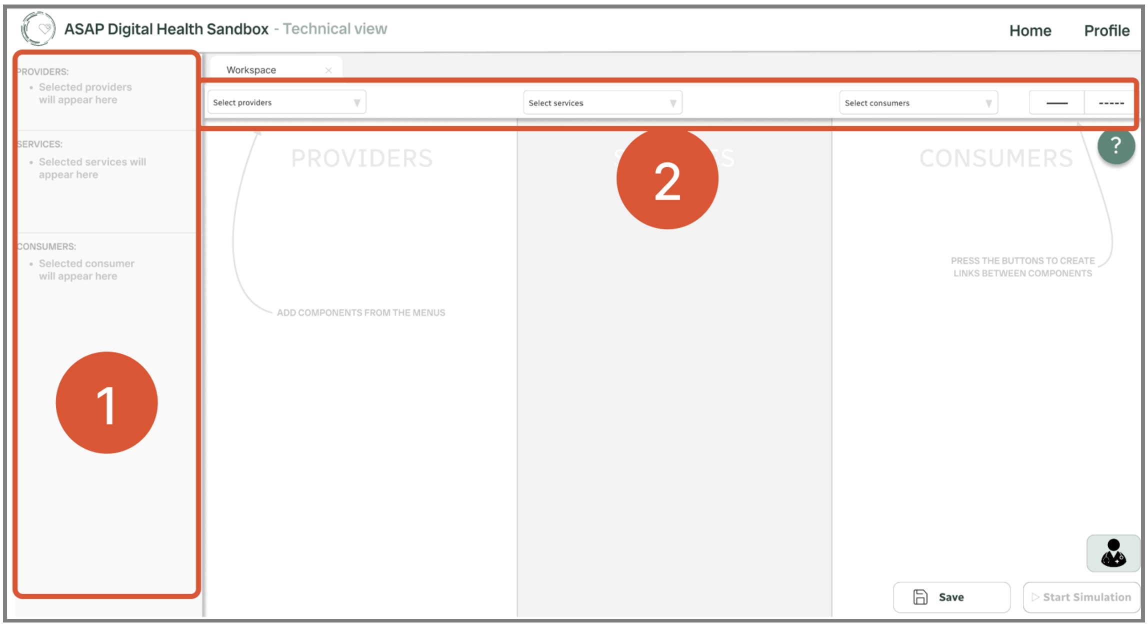


Figure 4.9: Menus in technical start view: (1) Side menu. (2) Top menu with drop down menus for selecting components.

The top menu bar features drop-down menus for adding components, as illustrated in Figure 4.9, where the menu bar is highlighted in red and marked with a red circle (2). There is one drop-down menu for each component type, aligned with the corresponding column in the workspace. Additionally, the menu bar includes toggle buttons for creating connections between component blocks. When hovering over these buttons, informational tooltips appear beneath them, as shown in Figure 4.10. After pressing a toggle button, the user can click on the two components they wish to connect.

Figure 4.10: Toggle button in the menu bar with hover activated information.

4.4.1.2 Provider Tab

The layout of the Provider tab depends on the type of provider, for example, a medical device and an EHR have different layouts due to their differing functionalities. Below, both are described.

4.4.1.2.1 Device Provider Tab The appearance of the Device Provider tab is illustrated in Figure 4.11. On the left, available data types are listed, with their current states indicated by state switch buttons. By default, data types that are typically pushed are pre-selected. Below the state switch buttons, a toggle bar is available for selecting the data source: Real data, Synthetic data, or a Combination of both. The specification of the selected or generated data are shown in a gray box beneath the toggle bar. If more input is needed, e.g. selecting patient data, these options are presented below the toggle bar but before the gray box appears.

On the right, a collapsible code box is provided, displaying the FHIR resources associated with the selected data. This box allows users to view and, if desired, edit the data. The code box can be collapsed to minimize distraction when not in use.

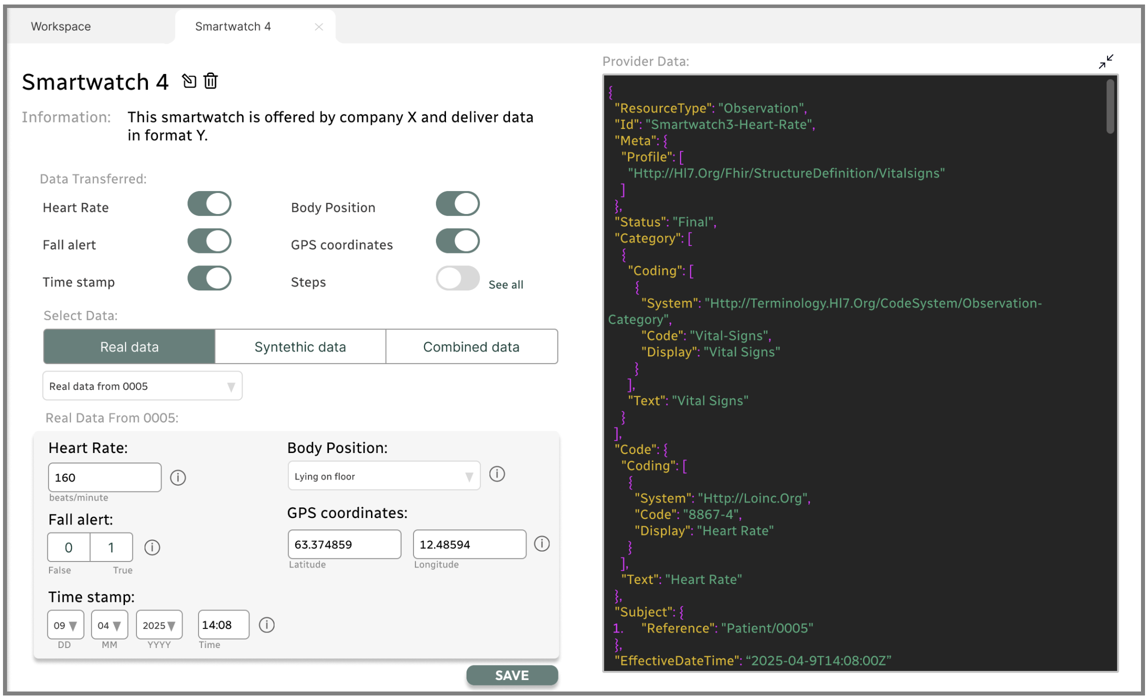
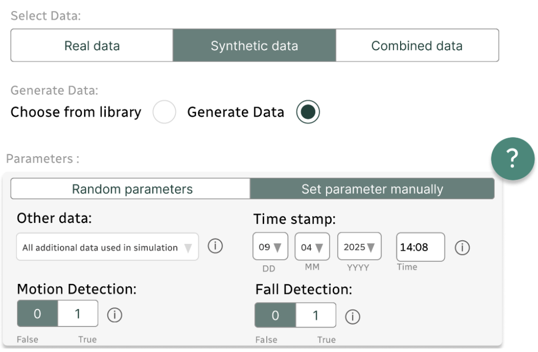
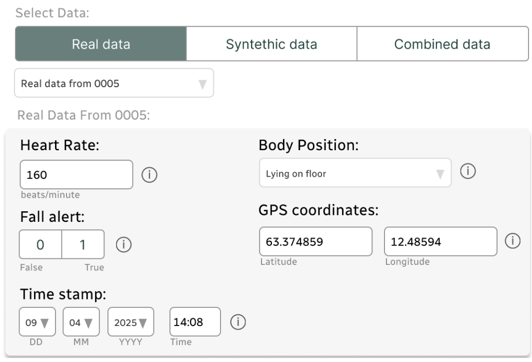


Figure 4.11: Provider tab for a smartwatch with settings for selecting data.

The appearance of the gray data box varies depending on the selected data source, as illustrated in Figure 4.12. The adjustable parameters correspond to the data types selected via the state switches at the top of the page. Next to each data type, an information icon is displayed, indicating that additional details are available. The information includes elements such as normal reference intervals and descriptions of the type of data represented.



(a) Visualizing generating synthetic data based on user-defined parameters for radar camera device.



(b) Visualizing real data selected and parameters that can be changed for smartwatch device.

Figure 4.12: Parts of the provider tab, showing section for generating synthetic data or selecting real data.

4.4.1.2.2 EHR Provider Tab The EHR Provider tab is specifically designed for components that access patient records, such as an EHR. Its layout reflects the complexity and flexibility required to retrieve various types of clinical data according to FHIR standards. At the top right, a toggle bar for selecting the data source is presented first, as illustrated in Figure 4.13. The available data sources are: Patient Library, Create patient and Generate patient. Patient Library includes synthetic patients and previously created patients. Create Patient allows users to manually create a patient and Generate Patient enables the automatic generation of synthetic patient data based on specifications.

On the right, a provider data box is displayed. This box has two views that can be switched between using the toggle bar in the left upper corner of the box: one for visualizing the FHIR resources included in the FHIR bundle, and another for displaying the FHIR bundle code itself. In both views, users can edit the resources to be included in the FHIR bundle. Each FHIR bundle is a link to the corresponding FHIR documentation. Allowing simplified access for the user if desired.

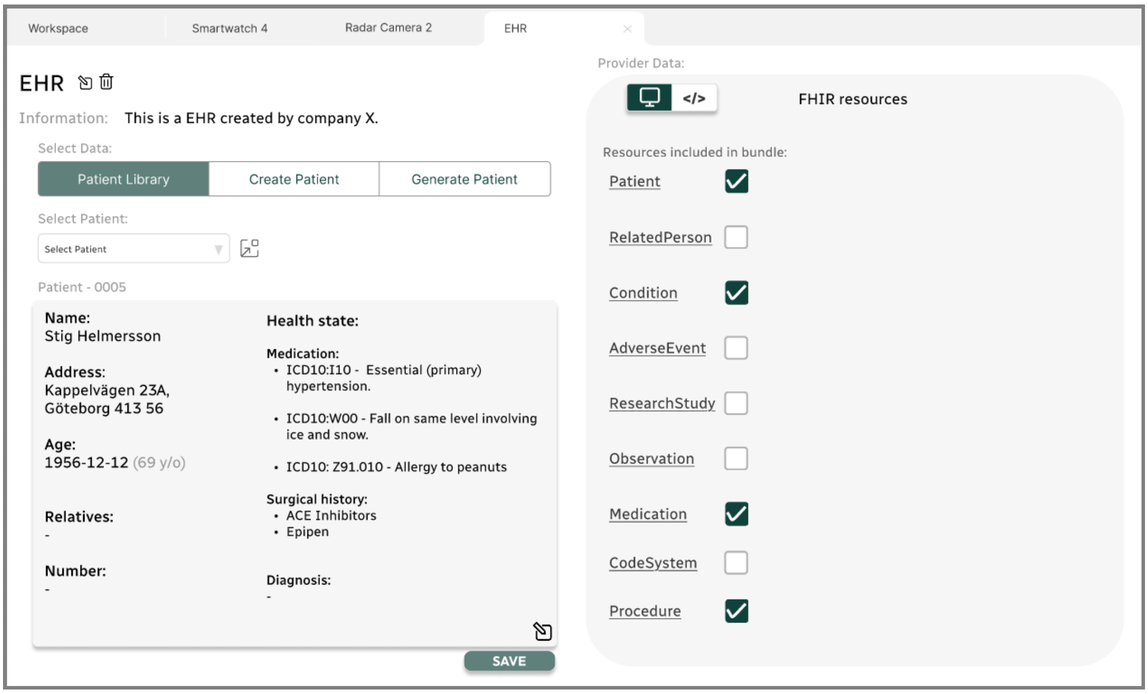


Figure 4.13: Visualization of an EHR Provider tab.

4.4.1.3 Service Tab

The Service tab illustrated in Figure 4.14, enables configuration of how incoming data is processed. On the left side of the page, general information about the selected service is displayed. Below this, data processing algorithm can be selected from a drop-down menu. Once an algorithm is selected, its input and output parameters are automatically visualized to provide a clear overview of its functionality and required data. Options for using pre-processing on incoming data are presented through radio buttons. When pre-processing is enabled, additional drop-down menus are dynamically displayed to allow the user to configure specific methods like normalization or removing outliers.

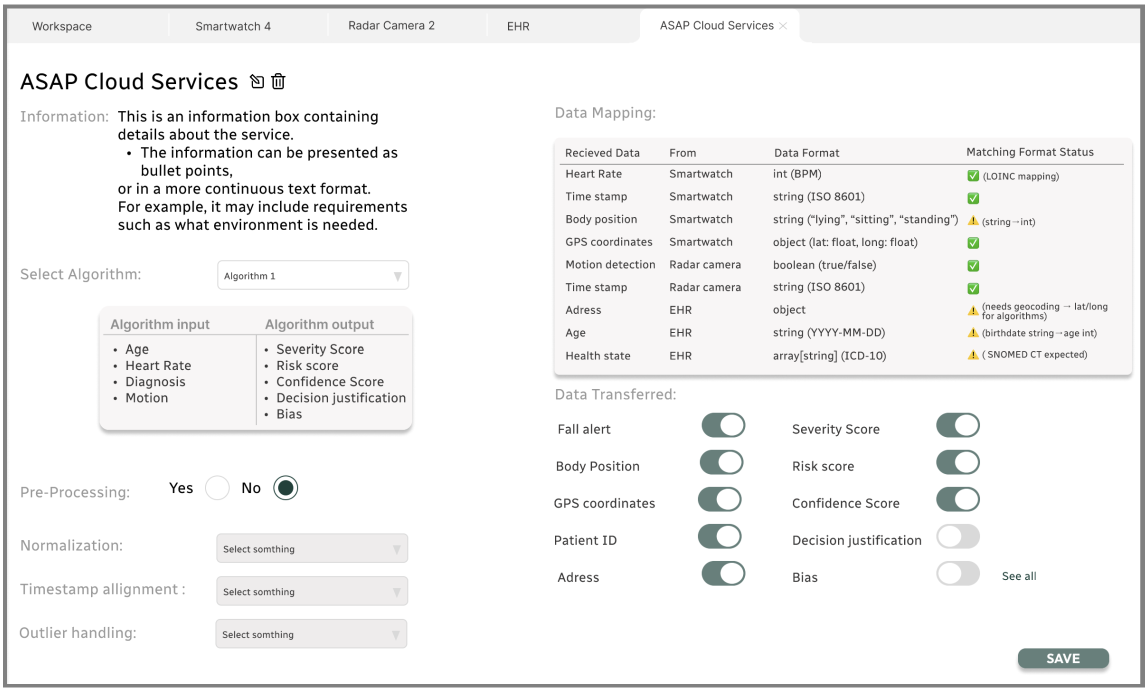


Figure 4.14: Service tab for selecting how to process incoming data.

On the right side of the page, a data mapping box presents an overview overview of the compatibility between incoming data formats and the expected formats of the selected service. If the formats match, a green check mark indicates successful compatibility; if not, a warning icon (exclamation mark) is shown. Clicking on the warning icon opens a pop-up window suggesting appropriate mapping solutions, as visualized in Figure 4.15. Below the data mapping box, switch buttons allow users to specify which data elements should be passed from the service to subsequent components in the system.

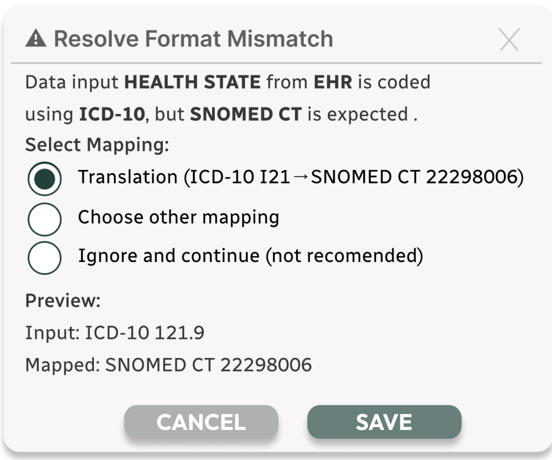


Figure 4.15: Pop-up menu for selecting and configuring data mappings. The window provides information about detected mismatches and offers recommended mapping solutions.

4.4.1.4 Consumer Tab

The Consumer tab resembles the Provider tab, with configuration options presented on the left and a code box located on the right, as illustrated in Figure 4.16. In the code box, users can toggle between a preview of the alarm center interface, the corresponding HTML code and FHIR resource code for the included elements. These views can be accessed via a toggle button located in the top left corner of the code box.

The method for presenting data in the alarm center interface can be selected from the drop-down menu on the left side. Once a method is chosen, red numbered squares appear in the interface preview, highlighting the areas where data can be inserted. Each square corresponds to a pre-defined section with an associated checkbox on the left. Activating a checkbox enables that section, allowing the user to assign specific data to it via an adjacent drop-down menu.

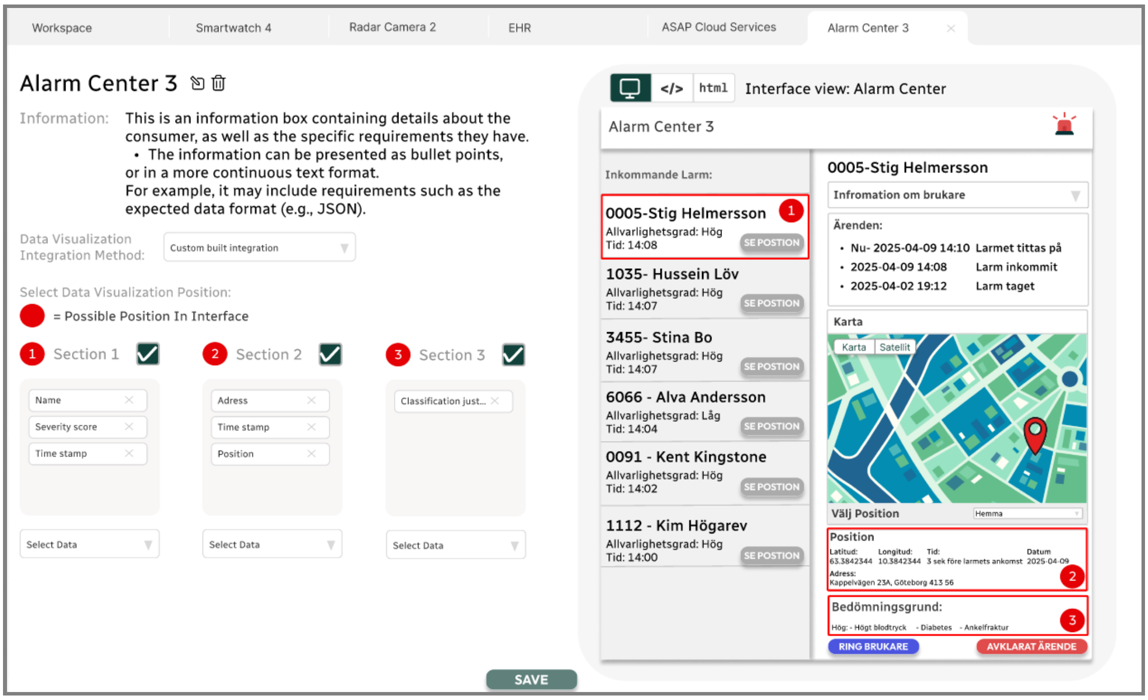


Figure 4.16: Visualization of Consumer tab.

4.4.2 Clinical View

The clinical view of the GUI provides a patient-centered environment for setting up simulations. It includes fewer advanced settings than the technical view to ensure ease of use for clinical users. Similar to the technical view, dedicated tabs are used tailored to support specific functionalities of each part. The clinical interface includes tabs for Patients, Events, Devices, and End users, as well as a Workspace tab. As in the technical view the workspace allows the user to add components in any desired order. A visualization of how the website flow can be utilized for this specific case is shown in Figure 4.17. Following sections outlines all the tabs available in the Clinical View.

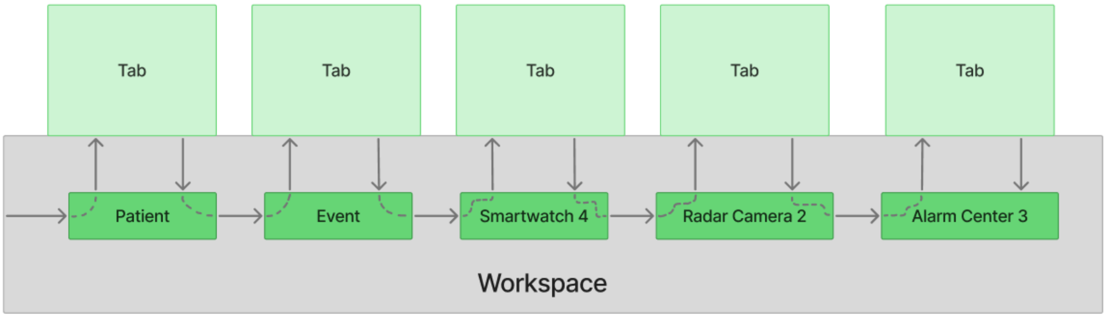
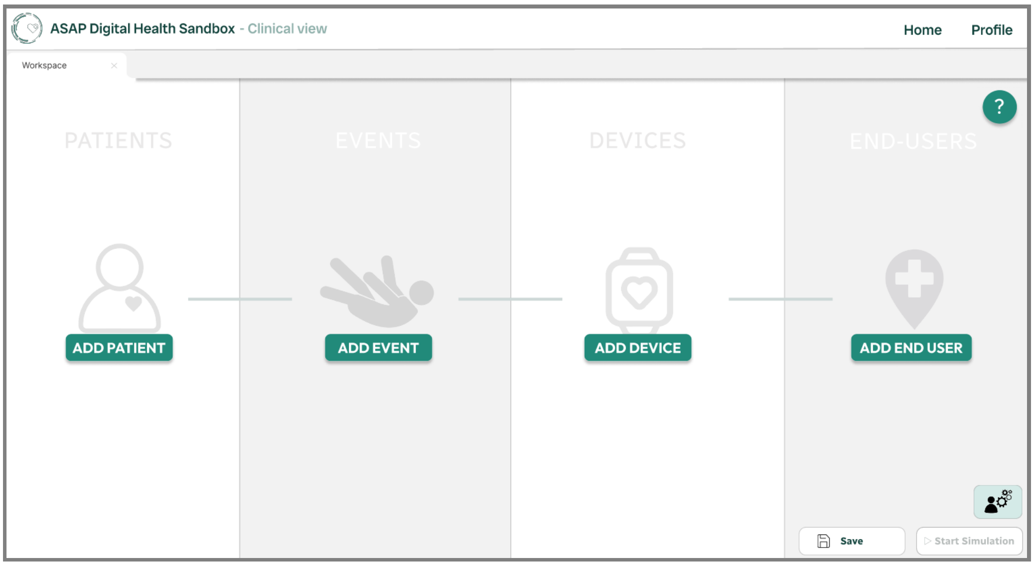


Figure 4.17: Visualization of navigation flow in clinical view.

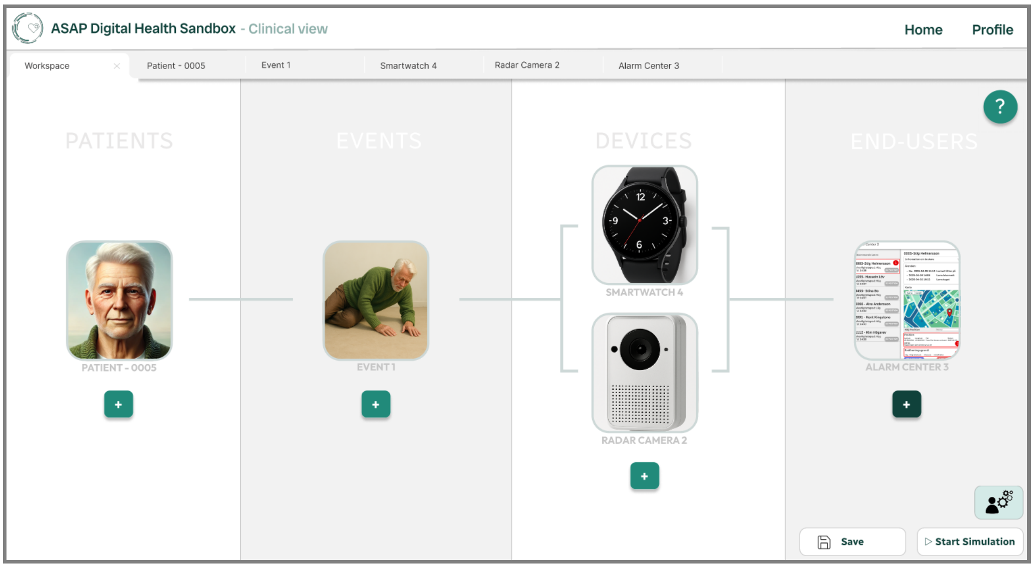
4.4.2.1 Workspace Tab

Figure 4.18 illustrates the Workspace tab for the clinical view. For an empty view of the workspace see Figure 4.18a and for a completed simulation setup, see Figure 4.18b. The workspace tab is divided into four sections: Patient, Event, Devices and End user. These categories are designed to help users easily understand what should be added at each step.

Initially, each section displays placeholder symbols representing the different types of components that can be added, as shown in Figure 4.18a. Below each symbol, a button is available indicating that user can click the symbol or button to add a component. When clicked, the user is redirected to the tab corresponding to the selected component type. Once a component is selected, the placeholder is replaced by a representative image to further support intuitive interaction. Below to the image, an additional '+' button is displayed to indicate that multiple components of the same type can be added, see Figure 4.18b.



(a) Start of clinical view with empty workspace.

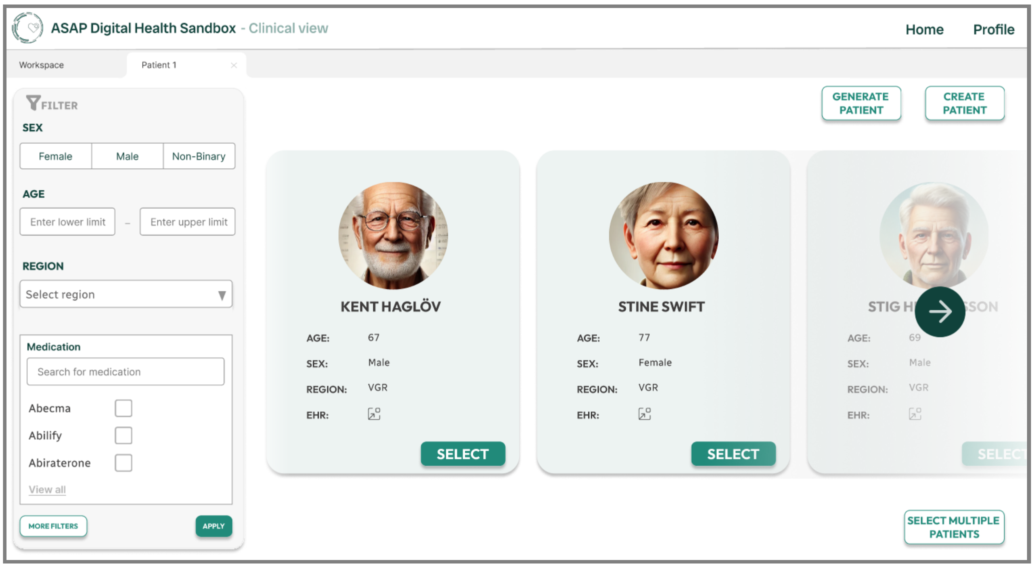


(b) Completed simulation setup in clinical view. (Patient and device images generated with ChatGPT using the prompt "elderly person, profile view", "elderly person falling, neutral background" "smartwatch, neutral background" and "radar camera, neutral background", 2025).

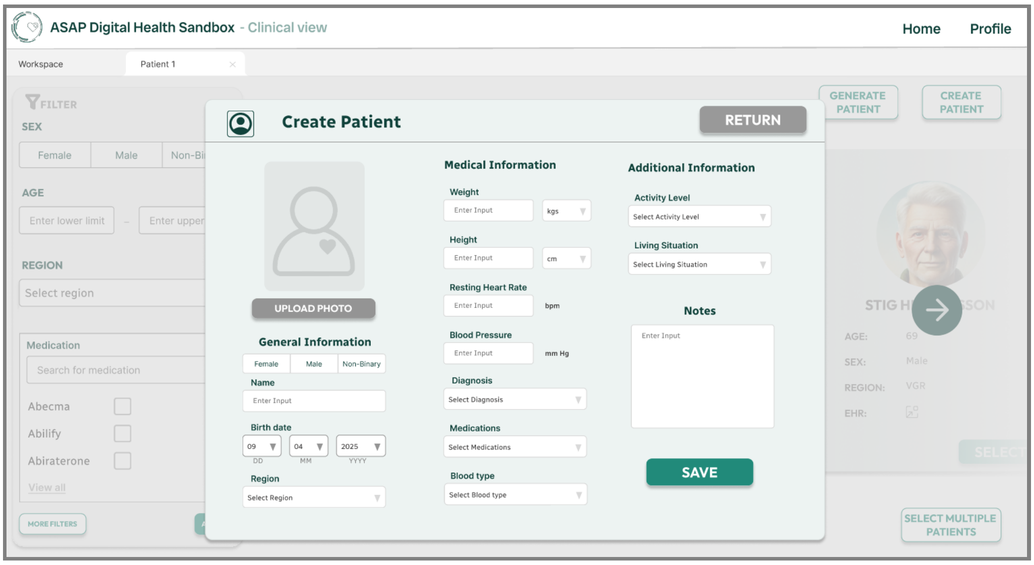
Figure 4.18: Clinical view: start and completed simulation setup.

4.4.2.2 Patient Tab

The Patient tab initially directs the user to the patient library, as shown in Figure 4.19a. In the library, filters can be applied on the left using toggle bars, drop-down menus, and other selection mechanisms. On the right, patient cards are displayed, presenting details such as image, name, age, etc. By clicking the arrow on the left side of the card, users can navigate through the patients in the library. Multiple patients can be selected by clicking the button in the lower right corner. A new patient can be created by clicking the "Create Patient" button in the upper right corner, which opens a pop-up menu, as shown in Figure 4.19b. This menu allows users to define the characteristics of the patient using drop-down menus and text input fields. Additionally, a patient can be generated either randomly or based on specified parameters by pressing the "Generate Patient" button, located in the upper right corner of the Patient Library, as illustrated in Figure 4.19a.



(a) Selecting a patient from the patient library.



(b) Creating an own patient.

Figure 4.19: Selecting a patent for the simulation. (Patient images generated with ChatGPT using the prompt "elderly person, profile view", 2025).

4.4.2.3 Event Tab

The Event tab contains a card with three main buttons that allow the user to specify the type of event to be triggered. Below these buttons, the event name can be set if desired. By default, the name is "Event 1", but it can be changed by the user. Following the name field, users can define the event's location, severity, and a description. The location is selected using radio buttons, while the severity is chosen from a drop-down menu. A description is automatically generated based on the previous decisions, but can be manually edited if desired, as seen in Figure 4.20.

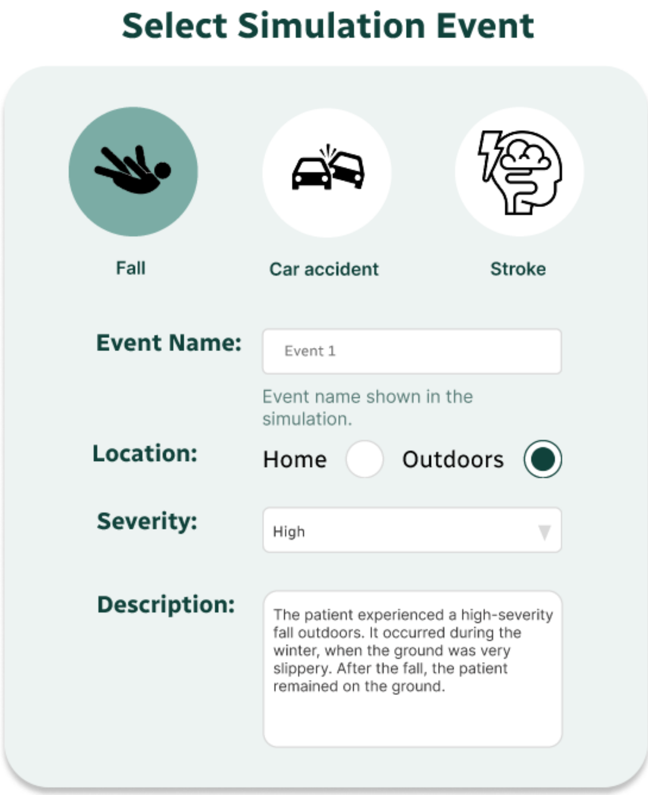
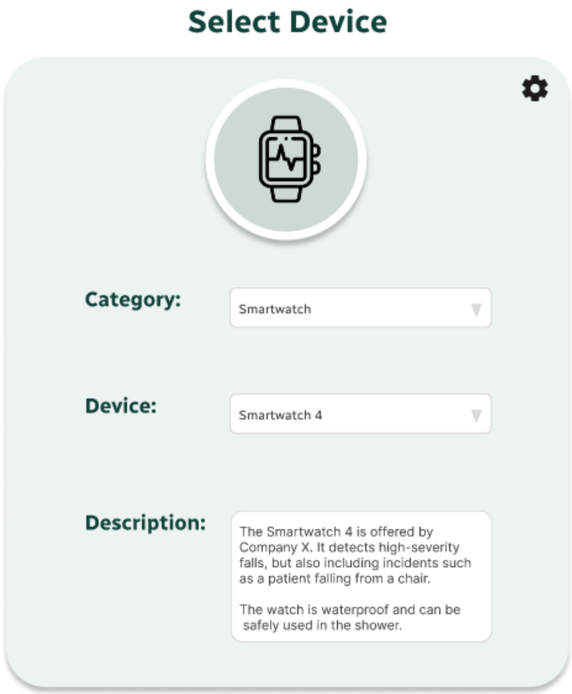


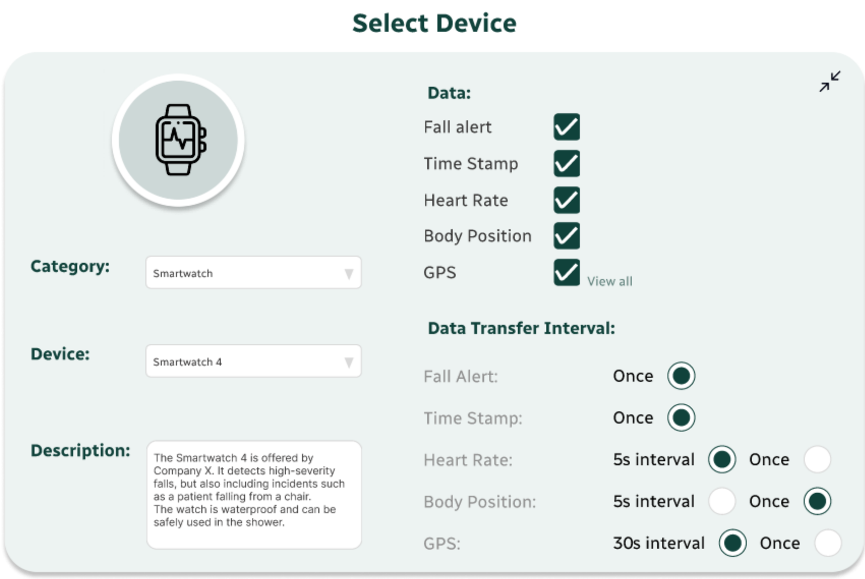
Figure 4.20: Visualization of the event card in the Event tab.

4.4.2.4 Device Tab

The Device tab allows users to configure device-related settings using a card interface, as shown in Figure 4.21a. The card can be expanded by clicking the settings icon in the top right corner, revealing more advanced settings, see Figure 4.21b. In the simplified card view, only the device category and specific device are selected. A description then appears at the bottom, providing brief information about the device. In the expanded advanced view, additional settings are displayed, including data transfer and data transfer intervals. These settings are controlled using checkboxes and radio buttons for ease of interaction.



(a) Selecting a smartwatch device.



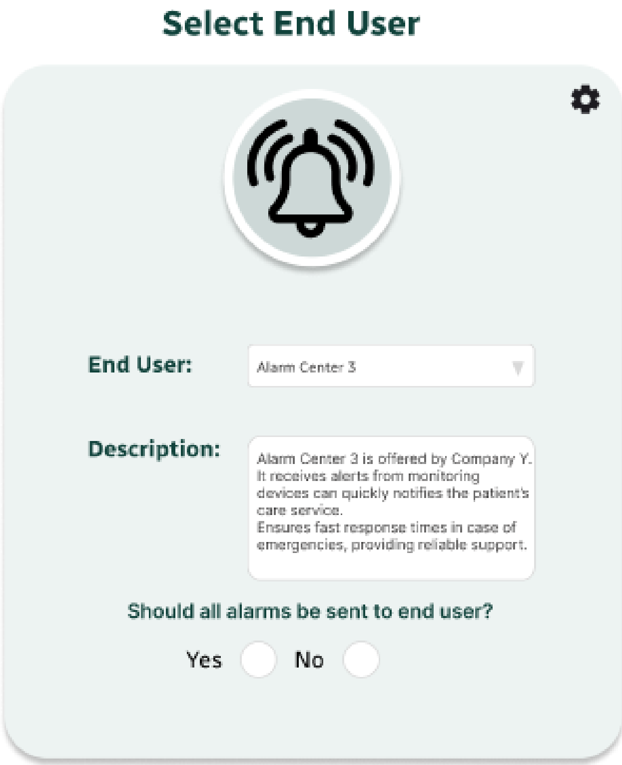
(b) Advanced view for settings of smartwatch device.

Figure 4.21: Visualization of device tab.

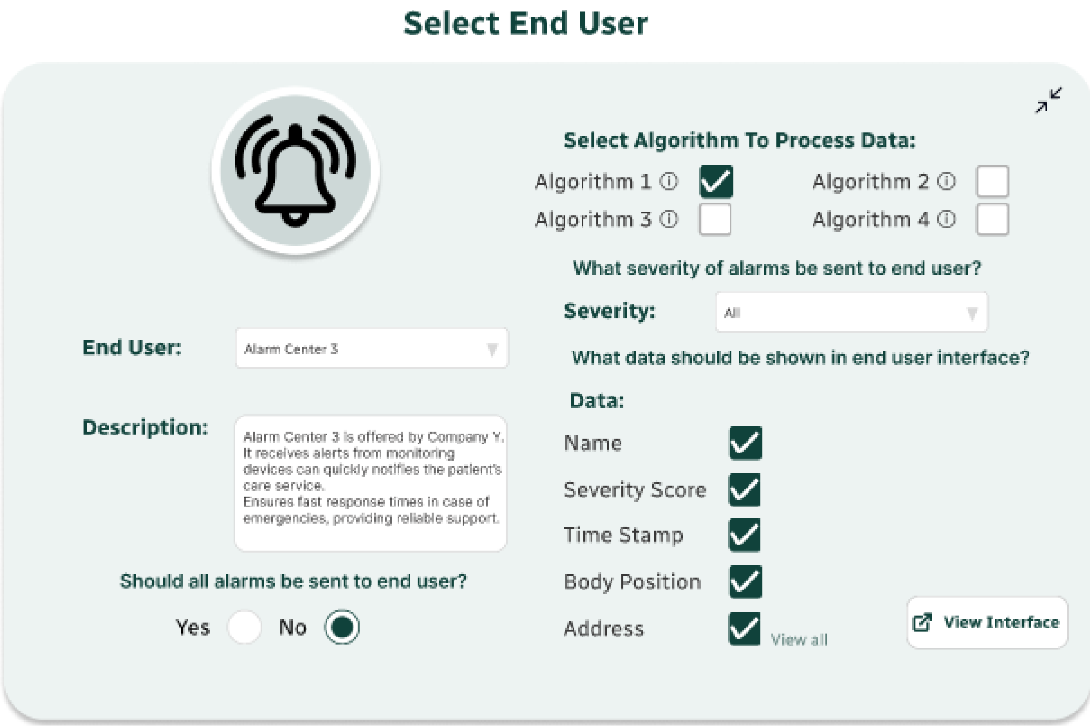
4.4.3 End User Tab

The End user tab is structurally similar to the Device tab, containing an expandable card that initially displays basic settings, see Figure 4.22a and can be expanded to reveal advanced configurations, see Figure 4.22b. Additionally, it includes a preview of the end user interface, see Figure 4.22c, where data can be mapped to its final visual presentation.

In the compact view, the user can select an end user from a drop-down menu and choose whether all alerts should be forwarded to the selected user via radio buttons. If "No" is selected, the card automatically expands to show additional configuration options in Figure 4.22b. In this expanded view, an algorithm can be selected to process incoming data, and thresholds for alert severity can be defined to determine which alerts should be forwarded. A "View Interface" button allows the user to preview how data will be presented in the end user interface. When clicked, the interface layout is displayed, highlighting sections (with red rectangles) where data can be visualized, see Figure 4.22c. The user can then assign specific data elements to each section using the section check boxes and drop down menus.

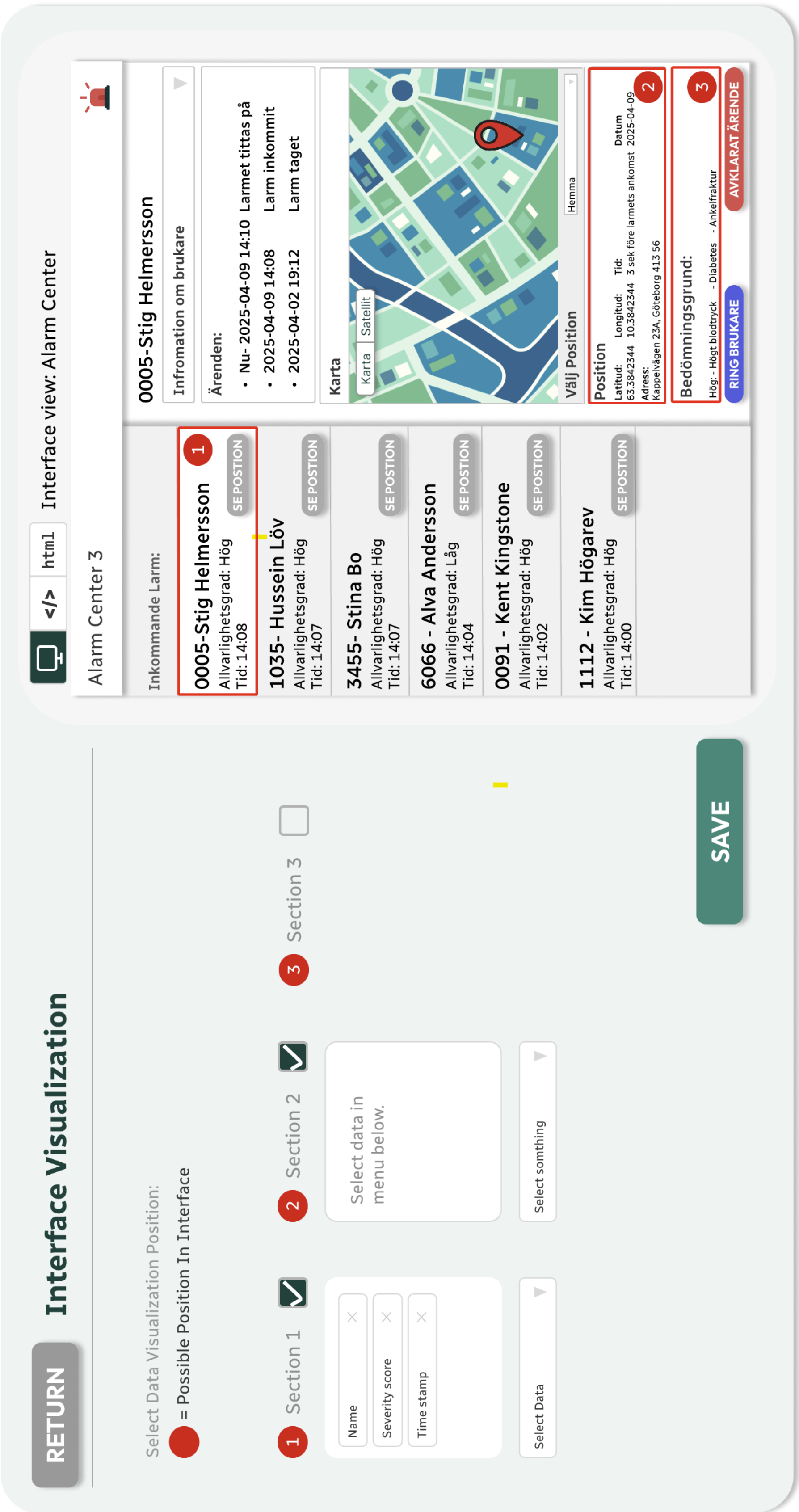


(a) Selecting a smartwatch device.



(b) Advanced view for settings of end user.

Figure 4.22: Visualization of End user tab.



(c) End user interface preview and selection

Figure 4.22: Visualization of End user tab.

4.5 Simulation Result Protocol

Based on user requirements for presenting the simulation results, it was concluded that the purpose of the simulation result protocol is to provide the simulation executor or external stakeholder with relevant information about the simulation. It contains detailed specifications for replicating, improving or implementing the set up. The protocol is delivered in PDF format, making it easy to save, share, and print. This ensures seamless collaboration across teams and allows the protocol to serve as a valuable decision-support tool when deploying or refining simulated setups.

The proposed structure of the simulation result protocol is visualized in figure 4.23. The identified components that is included in the sketch are:

- **Header** - Specifies the type of document, including the title and other relevant details.
- **Date** - The date the simulation was performed.
- **Simulation blueprint** - Describing the simulation so it can be replicated.
- **Watermark** - Identifies that the simulation was executed using the ASAP Sandbox, ensuring traceability and reliability.
- **Simulation executor** - Name or ID of the person setting up and performing the simulation.
- **Notices** - Important information, e.g. GPS position not possible when user is in basement or algorithm outcome and how decision-making was made.
- **Warnings** - Problematic areas with the set up. (e.g. Data transfer with this high interval will cause device running out of battery).
- **Issues** - Parts that will not work in the set up. (e.g. Component A records "heart rate" in bpm while Component B expects "pulse" coded using SNOMED CT, causing a mismatch in data interpretation).
- **Notes** - Space for the simulation executor to leave notes/thoughts about the simulations.
- **Recommendations** - Suggestions for improving the setup (e.g. Add data preprocessing and terminology mapping for Component B to align both format and meaning of incoming data).

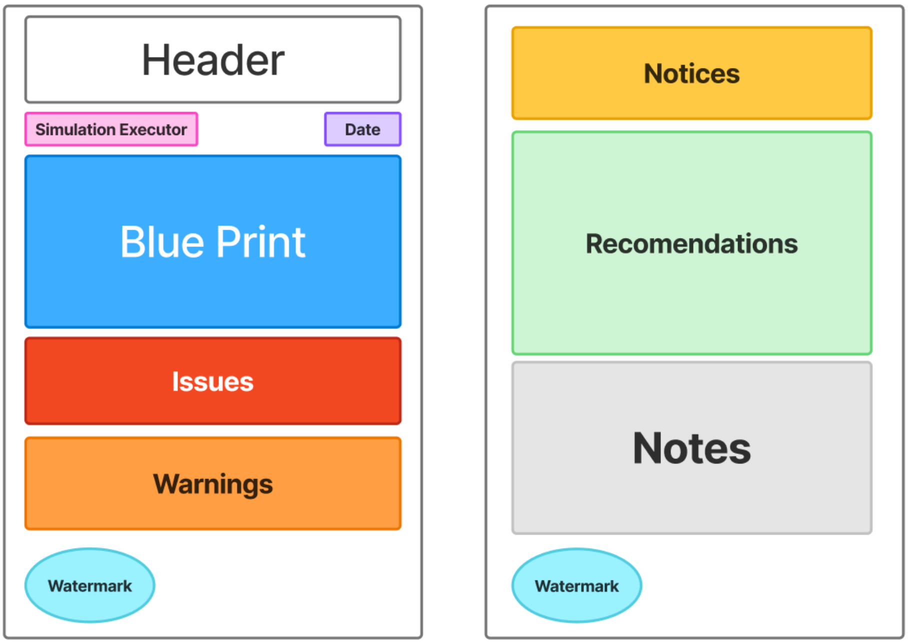


Figure 4.23: Sketch visualizing suggested structure of simulation result protocol.

The design has its foundation in the user requirements and design choices are made based on the input from the two user tests. The initial card-sorting test provided insights into how participants intuitively preferred information to be structured in a document, based on their own perspectives. The revealed a clear preference for placing the header, along with the date and simulation executor at the top of the document. This appears to be an unwritten rule and a common understanding among the participants. Additionally, the simulation blueprint was consistently one of the first elements that the participants wanted to be presented, reinforcing its position early in the protocol. Participants intuitively placed the watermark at the end of the document, following components such as notes. The four core information components: Issues, Warnings, Notices, and Recommendations, were arranged in varying order but were generally grouped together. This results serves as the foundation for the final sketch structure.

More interesting insights given from the initial card sorting test was that the number of pages that participants preferred varied. Some argued that it depended on the amount of information, emphasizing that the order of components was most important. Others preferred the information to fit on a single page if possible. Figures showing test participants results is presented in Appendix C.

In the second test, participants where asked to rank four different document layouts. Important findings from this test include that the simulation protocol should have a simple overview on the first page, making it more suitable for less technically experienced users, while providing in-depth technical details on the following pages. The structure of the initial summary page should follow the same layout as the following pages. Additionally, the watermark should be presented on all pages. For more details insights and participants comments see Appendix C.

5

Discussion

The resulting GUI indicate promising feedback from test participants, suggesting that the GUI has reached a satisfactory level of usability. However, several aspects remain to be explored in order to develop a GUI that functions effectively across a wide range of scenarios and users. As highlighted in the sandbox literature review, sandbox environments show great potential and are beginning to emerge on the market. Nonetheless, we identify a gap in terms of user-friendliness in the available sandboxes, which we believe is one of the most important factors to consider for the further development of the sandbox GUI. The following section presents a systematic discussion of both the promising results and the areas with potential for future development.

5.1 Case

The selected simulation case was designed to reflect a representative and realistic scenario relevant for potential users of the sandbox environment. While the initial ambition was to base the case on existing commercial products identified through the literature review and the interviews, practical limitations emerged. Due to difficulties of finding information detailed enough about the components we decided to use fictive components inspired by products on the market.

The case setup included three data providers, one service and one consumer forming a basic yet illustrative IoMT setup. This structure aligns with one of the sandbox's core purpose: to support simulation and evaluation of interoperability between systems, enable algorithmic decision-making, and assess clinical outcomes based on various configurations.

The smartwatch device was chosen due to increasing interest of implementation in elderly care within VGR and its relevance to VGRs existing projects. To reflect multi-device integration, a non-wearable radar camera was added. This decision not only allowed the GUI to support and visualize multiple simultaneous providers but also responded to real-world use cases in which passive monitoring is valuable, such as when patients forget or are unwilling to wear a device. The radar sensor thus introduced diversity in both the data types and technical interaction patterns simulated in the GUI. The inclusion of an EHR data provider added another layer of complexity and realism, enabling simulations involving patient history, contextual clinical data, and dynamic inputs for algorithm-based decisions. A cloud-based processing service with a rule-based algorithm was introduced to simulate automated decision-making, a critical component in future AI-assisted healthcare scenarios.

The consumer in the simulation, represented by an alarm center interface, reflects a realistic recipient of decision outputs in the chain of care.

Although this case provided a meaningful foundation for GUI design, it was a simplification of the complex reality that the sandbox ultimately aims to support. In practice, simulations may involve a greater number of providers, multiple services, several consumers, and more varied patient cases. Expanding the GUI to support these configurations in a simpler and more comprehensive manner remains an important direction for future development.

The interface structure chosen allowed for focused GUI development and evaluation, ensuring the interface could handle essential workflows and logic before scaling up. This iterative approach aligns with best practices in HCD facilitating continuous refinement based on evolving user needs and technical requirements.

5.2 User Requirements

User requirements were specified from two perspectives: clinical and technical. Given the diverse expectations and visions for the sandbox expressed by the interviewees, it may be valuable to broaden the analysis to include users with a wider range of backgrounds. As a result, additional personas could be developed, complementing the two used in this project. Using only one persona to represent a wide range of users, such as procurement staff and physicians, may oversimplify the user base and overlook important differences in needs and expectations. The objective would not be to create additional interface views, but rather to deepen the understanding of user needs and to identify key characteristics that should shape the sandbox's functionality.

The presented user requirements were not fully met. Some could have been implemented in the GUI with more reflection, although the majority were addressed successfully. The most important requirement from a clinical perspective was the emphasis on patient-centered simulation. This requirement played a key role in shaping the clinical view into what it became. If there is one main conclusion to highlight, it is that a patient-centered perspective is essential for meeting the needs of clinical users and supporting their understanding and effective use of the sandbox environment. Perhaps this should have been considered a technical requirement as well. Having technical and clinical professionals work from the same fundamental principles, while still addressing their differing technical requirements, could enhance collaboration across disciplines and promote a shared understanding between user groups.

Overall, all interviewees recognized significant potential in the sandbox environment, although their visions for its use varied. A clearer definition of the sandbox's purpose and capabilities could support users in better understanding what is possible within the environment. This in turn may clarify their preferences for functionality and interaction, supporting the formulation of more precise user requirements.

The process of identifying and addressing user requirements was grounded in the HCD principles, emphasizing the importance of understanding users' goals, contexts, and challenges. By engaging both clinical and technical stakeholders early in the design process, the project aimed to iteratively refine the sandbox GUI in response to user feedback and real-world needs. This approach ensured that the system was not only technically functional but also aligned with the values and expectations of its intended users.

5.3 GUI

The project initially aimed to develop two distinct user interfaces, one tailored for technical users and another for clinical users. However, during the design process, following the iterative approach of HCD early involving and receiving feedback from the intended user groups, it became evident that integrating these into a shared interface with switchable views offered substantial benefits. This structure enables users to toggle between simplified and advanced configurations, facilitating collaboration across user groups without disrupting the simulation setup. Users can easily return to a simpler view, making the system more user-friendly for first time users and less technically experienced.

From the outset, the design approach emphasized consistency and re-usability. Commonly recognized UI elements such as dropdown menus, toggle buttons, and expandable cards were deliberately chosen and reused for the same functions across views to reduce cognitive load and promote a coherent user experience. This strategy helped streamline the design and provided a more intuitive interface for both user groups. The choice to design around a visual representation of each component, grouped after component type, and clear navigation patterns was introduced to support usability and aid in mental model formation.

The GUI was evaluated using a limited but diverse group of participants, including both technical and clinical users. While the overall feedback was positive, several areas for improvement were identified. One key insight was the need for more interactive hints or guidance during the setup process. Although the current interface includes some hints, like text and information boxes, additional contextual support would be beneficial, especially as more advanced settings is to be introduced in the future. A tutorial video was also suggested as a helpful resource for first-time users navigating the interface. This was notably absent in many sandbox environments reviewed during the literature study and could significantly improve the on-boarding experience. Another requested feature was the inclusion of predefined simulation cases. These would not only assist users in understanding the sandbox functionality but also reduce the number of steps needed to run a basic, standard simulation. While the ability to save and reload simulations is already planned, offering a few example cases upon initial launch would further streamline the user experience. Lastly, integrating support for common accelerators such as copy, paste, and cut (e.g., Ctrl+C, Ctrl+V, Ctrl+X), would enhance usability and align the interface with standard user expectations.

The current GUI design should be seen as a conceptual prototype or an initial implementation that demonstrates how simulation setup could be approached in the sandbox. It was developed and tested based on a limited, predefined case and a relatively small user group. Rather than focusing on determining exactly what elements to include or the precise terminology to use, the project explored how such elements could be represented and structured in the interface. As such, all text in the GUI should be considered provisional, and we recommend further refinement to ensure the terminology accurately reflects the intended context. Still, the modular structure of the interface allows it to be extended to support more complex configurations and diverse use cases in future iterations.

5.3.1 Technical View

The technical view of the GUI provides an intuitive and visual approach to setting up simulations while still offering access to advanced configurations, with potential for further extension in the future. However, one key area for improvement is the method of creating connections between components. Currently, this can be done either by clicking on component arrows that appear by hovering or by using the data transfer toggle bar in the menu bar. Test participants also proposed alternative approaches, such as enabling drag-and-drop functionality between components to create connections more intuitively. One alternative design, which could improve usability and align with the drag-and-drop concept, involves integrating ports into the component blocks. Ports could visually indicate direction and the type of signal being transferred. This feature was included in one of the alternatives tested during the A/B testing of the technical mockups. While that specific mockup was not highly appreciated overall, the port feature itself was not negatively received. In fact, within the resulting design, ports could potentially enhance user-friendliness. Such a design could also reduce confusion regarding signal direction, i.e., which sides of the component different types of data are transmitted from. Currently, arrows at the top and bottom of components represent request-based data transfers, while arrows on the right and left represent push-based transfers. This approach was found to be somewhat unclear by several test participants. Introducing shape- and color-coded ports may help address this issue by making data flow more visually intuitive. For an example of how this was implemented in the mockup test, see Table A.3 in Appendix A.

Additionally, some minor enhancements like adding labels above the toggle button and within the pop-up menus activated via the component arrows could create a great difference when it comes to user friendliness. Overall, offering multiple interaction methods seems beneficial, as users bring varying levels of experience to the interface. A successful example of this approach is the ability to edit component settings through different access points: by clicking on the component directly, navigating via the tab, or using the side menu.

Another important area that requires further investigation is the type of information that should be displayed and which settings should be editable within each tab of the interface. While some parameters have been identified, more detailed requirements or functional specifications from clinicians and technicians working in the field would be highly valuable. For example, one area of uncertainty concerns the Consumer tab, where visualization methods are selected for integrating data into the end user interface. It remains unclear whether users should be able to select the visualization method themselves (like CDS-hooks, Smart on FHIR or own integration method), or if this should be predefined when the interface is integrated into the sandbox. Another open question is whether code input fields should be included in the tabs, allowing users to modify component behavior directly. Some test participants found this functionality helpful, while others did not; it may be worth considering whether such features should depend on the type of component, such as specific providers or consumers. To address these uncertainties, a focused prototyping iteration specifically targeting the settings tabs could be conducted. This would help ensure that the interface is as user-friendly and functional as possible for its intended users. Regardless of the final design choices, the most important principle is that the interface must remain flexible, yet simple, to support intuitive and effective use.

Related to the previously mentioned Consumer tab, the end user interface includes red-marked sections that appear when the user selects where to input information. These elements are not commonly found in standard user interfaces, but they can be implemented effectively through component mapping in Nuxt. Nuxt is a JavaScript framework based on Vue.js that enables component-based implementations and the development of customized, modular web interfaces [92]. By designing these sections as reusable components, they can be integrated into the interface in a technically feasible and maintainable way.

5.3.2 Clinical View

The clinical view of the GUI was designed from a patient-centered perspective, which significantly improved the understanding of the simulation setup and its purpose. Several technically experienced test participants who evaluated both views emphasized that the clinical view offered a much clearer and more intuitive understanding of the simulation configuration. This highlights the value of integrating the clinical and technical views, particularly to support and simplify the experience for first-time technical users.

One of the most important design decisions contributing to the simplicity of the clinical view were changing the sections and structure of the workspace to:

- **Patients** (Providers)
- **Events** (Data selection)
- **Devices** (Providers)
- **End user** (Consumers and Services)

As shown in the list above, services are included in the end user selection. This decision was made to reduce complexity, assuming that end users likely already have cloud services or processing steps in place. This simplifies the setup process, especially for users with limited technical backgrounds. Providers were divided into "patients" and "devices" to reinforce the patient-centered approach and to improve the clarity of the simulation flow. Additionally, data selection was separated from each individual component into the event section, making the simulation process feel more realistic and relatable. Lastly, "consumers" were renamed to "end users" for better clarity.

These key changes played a crucial role in helping clinical users understand the simulation setup and navigate the sandbox environment. Therefore, we recommend that these design principles be carried forward in future development efforts. We believe that this simple structure contributes to lowering the threshold for digital adoption among clinical users, by offering a familiar and intuitive environment.

5.3.2.1 Patient Labeling

In both the GUI and throughout this report, the term patient has been used to describe individuals involved in the simulation scenarios. However, based on the interview with Kelly, it was pointed out that this term is not always accurate. For example, older individuals with fall detection devices are often not considered patients in a clinical sense. In these cases, the responsibility usually lies within the region rather than hospitals. There are situations where people with a high risk of falling, such as those who might fall out of bed during hospital care, would be considered patients. We are aware that the term is not fully appropriate in every situation. Still, when a fall has occurred and a person is injured, it is considered a patient. Therefore, for simplicity, the term patient has been used consistently throughout this work.

5.3.3 Interoperability

While all components integrated into the sandbox are built upon the FHIR standard, which supports syntactic interoperability by enabling structured data exchange, this alone is not sufficient for ensuring that systems can function seamlessly together. Semantic interoperability, the ability of systems to interpret exchanged information in a meaningful and consistent way, is critical for real-world healthcare applications. Achieving semantic interoperability often requires explicit data mapping between different terminologies, data formats, or value sets. Even when using a common standard such as FHIR, data originating from different devices or systems might be encoded differently. For example, one system might express a health condition using ICD-10 codes, while another expects LOINC concepts, patient age may be calculated form birth-date or provided directly, necessitating pre-processing or transformation steps. Including these functionalities in the GUI reinforces the sandbox's role for verifying system interoperability across different integration levels and suggestions for achieving higher levels.

We decided to not incorporate this level of interoperability configuration in the clinical view, as it introduces technical complexity that may be difficult for clinical users to interpret or act upon. Semantic data mapping involves understanding data structures, coding systems, and transformation logic, elements that fall outside the typical domain knowledge of healthcare professionals. Instead, the sandbox could be designed to handle these configurations automatically in the background for clinical users, either by applying recommended default mappings or by alerting technical users through the simulation result protocol when unresolved semantic mismatches exist. This allows clinicians to focus on evaluating the clinical validity and usefulness of the simulation outcomes, without being burdened by integration concerns.

By separating these concerns, the sandbox supports HCD principles, tailoring complexity to the user's expertise, while still maintaining transparency. Any automatic mapping or interoperability adjustments should be clearly documented in the simulation result protocol, ensuring traceability and enabling review by technical users when needed. This approach supports both usability and trust in the system, while still promoting accurate and meaningful data exchange in system simulations.

5.3.4 Testing

Throughout the project, tests were conducted to evaluate the design approaches and choices from a human centered perspective. One difficulty with this process was to recruiting testers who fully represent the intended users. As a result, the focus was primarily on selecting individuals with relevant technical experience. This meant that the evaluation centered more on the GUI's functionality than on identifying missing domain-specific features. While not all participants were familiar with specific concepts like FHIR, their experience with technical tools allowed them to provide valuable feedback on usability and interaction design.

The tests conducted included A/B testing and task-oriented evaluations, both of which provided valuable insights. One interesting approach for future testing could be to apply A/B testing to more specific elements such as individual components, tabs and menus to gain more targeted feedback on particular design choices. When A/B testing was performed on three completely different design alternatives for each view, the feedback tended to focus more on overall impressions rather than specific details. For the task oriented test, only main functionalities needed to set the simulation case were included, ensuring the test would not be too time-consuming or overwhelming for participants. Several task tests could therefore be considered for testing different parts of the flow. Moreover, incorporating test methods such as eye tracking could be valuable for gaining deeper insight into user attention and behavior, particularly to assess which interface elements draw the most focus and whether their placement feels intuitive.

5.3.5 Opportunities for Future Sandbox Design and Simulation Features

Several design directions and functionalities could not be fully explored within the scope of this project but have been identified as potentially valuable for future development.

A key improvement would be to enable users to specify the type and objective of a simulation early in the setup process. This could significantly reduce unnecessary complexity, guide the interface toward context-relevant options, and tailor the user experience. For example, a simulation focused on evaluating interoperability might require different configuration elements than one centered around algorithmic performance, clinical pathway testing, or user training. Similarly, simulations aimed at understanding basic system behavior might benefit from more predefined setups and richer visual explanations of data flows and interactions.

Currently, the design only covers the simulation setup interface. However, the simulation phase itself could benefit from interactive features. One concept discussed during the project was the ability to actively trigger events during simulation, such as initiating a fall scenario, and then following the chain of reactions in real time: when data is requested from other systems, what occurs in the service component, how the algorithm processes the data, and finally how the results are visualized for the user. This could greatly enhance the sandbox's utility for debugging, teaching, and design validation. Another potential future approach could be implementing a two-panel view, similar to what is commonly used in development environments such as Visual Studio Code or MATLAB. This would allow users to build and adjust their simulation setup on one side while simultaneously observing the simulation output on the other. Such a layout could be especially beneficial for technical users, enabling them to iteratively test smaller parts of the simulation before scaling up to more complex configurations.

Future iterations should also consider the design implications of running simulations involving multiple patients, devices, and end-users. For example, simulating multiple fall events occurring simultaneously or in sequence could test how well the system manages concurrent data flows, algorithm prioritization, or bandwidth limitations. The ability to coordinate and schedule such scenarios through a user-friendly interface would open new use cases in system stress testing and capacity planning. One possible approach to achieve this is by implementing the scenario engine as a specific part instead of having it integrated in different parts of the GUI, as it is the current design proposal. In a different design of the scenario engine, specifications for the events and involved patients could be managed separately from the core simulation setup. This could be presented on a dedicated tab or pop-up window, rather than being integrated with the component configuration. Another future opportunity is to support comparative simulations for evaluating different configurations. For instance, how does system performance or alarm accuracy change when a patient is monitored only by a smartwatch versus by both a smartwatch and a radar camera? These types of comparisons could provide valuable insights not only for technical evaluation but also for clinical decision-making and procurement in public healthcare. Over time, simulation outputs might even serve as supporting evidence for health technology assessment or investment decisions in regional care systems.

5.3.6 Reflections on HCD Approach

The use of a HCD approach in the development of the GUI allowed for early feedback from potential users, which contributed to a more intuitive and tailored interface. By involving users in design phase, the interface could be adapted to better support their needs and reduce cognitive load. However, the process also had limitations. The small number of users involved may have introduced bias, and not all feedback could be incorporated due to technical or scope constraints. Additionally, while HCD is effective in optimizing usability, it may limit innovation by focusing too closely on current user expectations rather than exploring new interaction methods. This may be a particular limitation when designing a sandbox GUI, as such interfaces are often unfamiliar to users and therefore harder to evaluate using their existing expectations.

5.4 Simulation Result Protocol

The simulation result protocol was designed as a downloadable PDF document, enabling it to be shared with individuals outside the sandbox environment. Due to uncertainty regarding what specific information would be outputted from the simulation, the protocol was kept as a sketch. The results to be displayed appear to vary significantly depending on the type of simulation performed, which highlights the importance for a high level of flexibility in the document's structure.

The simulation result protocol was developed through both a card sorting test and an A/B test. The card sorting provided valuable insights into how users prefer and expect information to be presented. However, it can be argued that this user test may introduce bias due to the participants' pre-existing expectations. Another consideration is that participants might not actually know how they would prefer the information to be displayed. In such cases, the results from the card sorting test may be less valuable, or even misleading. This uncertainty carries over into the sketches used for the A/B test, meaning that a different method could potentially have led to entirely different outcomes. Therefore, the current sketch should be regarded as a preliminary draft that requires further evaluation. Nevertheless, card sorting is a well-established method, and the fact that many participants organized the items in similar ways suggests that it was an effective approach, yielding meaningful and reliable insights.

The main focus for future development could be to continue considering the ability to share the protocol, both within and outside the sandbox, while also capturing the version of the sandbox used and the blueprint for understanding the simulations. This would allow the simulation protocol, in the future, to serve as a decision-support tool in the implementation of medical technology, as well as in the development of new technologies.

5.5 Reflections on the Sandbox Concept

The use of sandbox environments is expanding across multiple sectors, including healthcare. These environments offer significant potential for safe experimentation, prototyping, and evaluation of digital systems before real-world implementation. In healthcare, where system complexity and regulatory requirements are high, sandboxes can serve as invaluable tools for testing interoperability, refining workflows, and supporting innovation. However, to fully realize this potential, the sandbox must be accessible and usable for all intended users, something that hinges on the design of a user-friendly interface.

One of the key benefits of sandbox environments is their ability to simplify the development and integration of new digital systems or devices. By offering a standardized simulation platform, sandboxes enable developers to ensure that their solutions interact correctly with other parts of the healthcare ecosystem. This supports both monolithic architectures, where systems are developed and deployed as single unified applications, and microservices architectures, where systems are composed of independent, loosely coupled components that interact through defined interfaces. For microservices-based systems in particular, sandbox environments provide a valuable way to test individual services, data flows, and interoperability without needing to deploy the entire system. This modular testing approach improves flexibility, accelerates development, and supports reusability of services. It also makes it easier to iteratively build and validate systems, even across organizational boundaries. However, for this vision to be practically adopted, the simulation environment must offer intuitive tools for managing, configuring, and understanding system behavior, making the GUI a critical part of the overall solution.

5.5.1 Clinical Acceptance

One of the persistent barriers to successful digital transformation in healthcare is the slow pace of adoption and resistance to new technologies among clinical staff. This can be due to a lack of familiarity, poor usability of digital tools, or insufficient perceived value in everyday practice.

By providing clinicians with an intuitive and exploratory environment, such as this sandbox, there is a potential to increase the understanding and acceptance of digital systems. Allowing clinical users to simulate patient journeys, test new workflows, and visualize data integration could demystify technical solutions and strengthen confidence in their use. In this way, the sandbox can act not only as a technical testing platform but also as an educational and engagement tool for facilitating digital transformation in care.

5.5.2 Integrating Technology into Healthcare

One thing to consider when creating the sandbox, as well as performing simulations within it, is that healthcare might not yet be fully ready for it. By this, we mean that healthcare systems do not currently have infrastructure where systems like fall detection can be integrated effectively. Trygghetsjouren is not healthcare-related but is an assistance support service from the region. This means that if systems, like the case used in this project, are to function for patients (not only those with age related fall risk) a department needs to receive these kinds of alerts. This is a kind of monitoring department that does not exist today. Therefore, the implementation of solutions that might be simulated might become more of an organizational issue then a technical, according to Kelly. This emphasizes the importance of not only considering the technical aspects in simulations but also incorporating the workflows of healthcare staff and patient circumstances. The ultimate goal of the simulated system is to hopefully be realized in practice, and therefore, it is important to consider factors beyond just the technology.

5.6 Ethical, Social, and Environmental Factors

This project aimed to contribute to making the digitalization of healthcare easier to implement, resulting in fewer patients being affected by issues when new technologies or updates to existing devices are introduced. The sandbox environment also has the potential to improve decision-making within healthcare, as more scenarios can be tested to identify the optimal solution. This could lead to better healthcare for all individuals and demonstrates how this project can address both important social and ethical aspects.

MDR introduces strict requirements for the development and evaluation of medical devices, aiming to protect patients. This is achieved through rigorous testing of the device before it enters the market. While these regulations are essential for ensuring safety and efficacy, they also create challenges for technical development. Obtaining the necessary ethical approvals and perform testing is time-consuming. Having a sandbox environment that enables testing prior to these approvals ensures sufficient testing can be conducted beforehand, helping to decide whether it is worth to apply for ethical approval to test the device in real clinical settings. The ASAP sandbox could simplify the testing process by providing a controlled and compliant space for validating new technologies, thereby accelerating innovation while maintaining high ethical standards. Ensuring compliance with data privacy regulations, such as the GDPR and HIPAA, is another key ethical consideration for this project. The tool must be designed to handle sensitive user and patient data securely, upholding trust and legal compliance. The upcoming regulations around AI, including the AI Act, also emphasize the ethical importance of sandbox environments. Being able to test AI-powered algorithms in safe, controlled settings can enhance the development of healthcare solutions, improve their practical application and potentially result in better patient outcomes. It also has the potential to support the development of transparency in AI algorithms, ensure patient privacy, and detect bias in algorithms. All of these aspects are crucial for ensuring patient safety and protecting individual rights.

This project involved a small number of test participants to evaluate design suggestions for the GUI. To ensure participant anonymity, all feedback forms were fully anonymized and did not include any identifiable personal information such as names and contact details. Participation was entirely voluntary, and all participants were informed of the purpose of the testing and the anonymity of their involvement. Given the limited scope of the user tests and the non-sensitive information collected, the ethical risks to participants were assessed as minimal.

Considering ecological aspects, a sandbox that performs simulations may require substantial energy to operate. Therefore, it is valuable to consider how easy it should be to run simulations. Maybe it should be a control mechanism in place to ensure that simulations are correctly configured before execution, in order to avoid unnecessary or poorly set up simulations. Despite the potential environmental impact in terms of energy consumption, the sandbox's potential contribution to the implementation of medical technology, possibly saving time and perhaps patient lives, could justify the resource usage.

5.7 Future Work

Based on the discussion presented, we suggest the following points to be considered for future work:

- **User Requirements:**
  - Conduct additional interviews.
  - Define more specific user requirements.
  - Consider whether the technical user also should have a patient-centered perspective.
- **GUI:**
  - To support multiple scenarios, investigate how the scenario engine can be implemented more efficiently.
  - Explore how components can be connected.
  - Enable interactive simulation – define what should happen when "Start Simulation" is pressed.
  - Evaluate whether there should be two sandbox views or a unified view suitable for both user types.
- **Simulation Result Protocol:**
  - Identify what users expect to gain from the simulation.
  - Define how each part of the result should be structured and presented.
  - Determine appropriate fonts, colors, etc., that convey a professional appearance.

To summarize, the results presented lay a strong foundation for future work. By addressing the points outlined above, the full potential of the sandbox can be realized. This project should be seen as just the first step toward a truly adaptable, interdisciplinary platform for simulation and innovation with significant future potential.

6 Conclusion

This project aimed to design a web-based GUI for managing DT simulations in healthcare within the ASAP Sandbox environment. To answer the research questions, they are summarized and addressed as follows:

- *How can a sandbox GUI be designed, using a human-centered design approach, to effectively address the needs, workflows, and usability expectations of both technical and clinical users in the context of healthcare simulations?*

A HCD approach can be used to develop a sandbox GUI that accommodates both clinical and technical users through a multi-view interface. This involves gathering user requirements through interviews and applying iterative prototyping cycles to maximize user involvement. By dividing the interface into two views, a patient-centered view for clinical users and a system-oriented view for technical users, the GUI supports different levels of complexity and interaction styles. Using familiar GUI components and tailoring functionality to user needs improves usability and adoption.

- *Do digital sandbox environments exist within the healthcare domain, and if so, how are they designed, to what extent are they accessible, and what purposes do they serve?*

Digital sandboxes do exist within the healthcare domain, but within the scope of this project, their availability was found to be limited. No sandbox identified fully corresponds to the type of sandbox this project aimed to explore. Of the sandboxes examined, few were designed for clinical users. Most were code-based and required advanced technical knowledge. Many were not directly accessible, only descriptive information about them could be found.

- *Which types of fall detection devices and stakeholders are of relevance when simulating fall-related incidents in a healthcare context, and what commercially available devices could be used in such simulations?*

Many types of wearable and non-wearable fall detection devices are commercially available and relevant for healthcare simulations. Wearables like smartwatches offer motion-based detection but rely on users remembering to wear them, while non-wearables, such as radar cameras, enable continuous passive monitoring. Including both types enhances the realism of simulations and supports testing of diverse use cases and data flows.

- *How should a simulation result protocol be designed to ensure that simulated outcomes are interpretable, reusable, and valuable to different user groups?*

To ensure that simulation outcomes are interpretable, reusable, and valuable to different user groups, the simulation result protocol should be designed based on insights from interviews and user testing. The protocol should present the simulation blueprint first, as this is essential for repeatability and understanding of the setup. A consistent structure should be kept and include sections such as notices, warnings, and recommendations to supports traceability and clarity. An alternative approach is to provide a first summary page for quick overview, followed by more in-depth technical details in subsequent pages. Delivering the protocol in PDF format facilitates easy storage, sharing, and cross-functional collaboration. Together, these design choices promote effective decision-making and enable reuse or replication in future scenarios.

Bibliography

[1] C. Chen, S. Ding, and J. Wang, "Digital health for aging populations", en, *Nature Medicine*, vol. 29, no. 7, pp. 1623–1630, Jul. 2023. DOI: 10.1038/s41591-023-02391-8.

[2] F. Apelgren, M. Seth, H. Jalo, B. A. Sjöqvist, and S. Candefjord, "Introducing virtuality - virtual care process simulator: A concept utilizing synthetic data and a digital health sandbox for care process simulations", in *Proceedings of the 18th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2025)*, Portugal, 2025, pp. 1127–1135. [Online]. Available: https://doi.org/10.5220/0013401300003911 (visited on 01/14/2025).

[3] E. Leckenby, D. Dawoud, J. Bouvy, and P. Jónsson, "The Sandbox Approach and its Potential for Use in Health Technology Assessment: A Literature Review", en, *Applied Health Economics and Health Policy*, vol. 19, no. 6, pp. 857–869, Nov. 2021. DOI: 10.1007/s40258-021-00665-1.

[4] J. Yao, X. Zhang, Y. Wang, and L. Liu, "Digital twin technology: Concept, development, and applications in healthcare", *Journal of Medical Systems*, vol. 47, no. 137, pp. 1–15, 2023. DOI: 10.1007/s42492-023-00137-9.

[5] S. Tavakoli, "Modern IT-teknologi i sjukvården", sv, *Chalmers*, Dec. 2022. [Online]. Available: https://www.chalmers.se/aktuellt/nyheter/e2-modern-it-teknologi-i-sjukvarden/ (visited on 01/14/2025).

[6] F. HL7, *Summary - FHIR v5.0.0*, 2023. [Online]. Available: https://hl7.org/fhir/summary.html (visited on 03/13/2025).

[7] M. El Ghosh, V. Kalokyri, M. Sambres, *et al.*, "From Syntactic to Semantic Interoperability Using a Hyperontology in the Oncology Domain", in *Studies in Health Technology and Informatics*, J. Mantas, A. Hasman, G. Demiris, *et al.*, Eds., IOS Press, Aug. 2024. DOI: 10.3233/SHTI240670.

[8] M. M. Keenan, "Simulation", in *Salem Press Encyclopedia*, Ipswich, MA, USA, 2025. [Online]. Available: https://research.ebsco.com/linkprocessor/plink?id=206bfd10-d3b1-37ae-a609-1c96be773ee3 (visited on 03/13/2025).

[9] M. Shrivastava, R. Chugh, S. Gochhait, and A. B. Jibril, "A review on digital twin technology in healthcare", in *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, Uttarakhand, India, 2023, pp. 741–745. DOI: 10.1109/ICIDCA56705.2023.10099646.

[10] M. A. Hasan, R. Mustofa, N. U. I. Hossain, and M. S. Islam, “Smart health practices: Strategies to improve healthcare efficiency through digital twin technology”, *Smart Health*, vol. 36, p. 100 541, Feb. 2025. DOI: 10.1016/j.smhl.2025.100541.

[11] “Standard for models and simulations”, National Aeronautics and Space Administration (NASA), Washington, DC, USA, NASA-STD-7009, 2008, Available: https://standards.nasa.gov/sites/default/files/standards/NASA/w/CHANGE-1/1/Historical/nasa-std-7009.pdf.

[12] J. Elvidge, N. Crabb, D. Delnoij, *et al.*, “Implementing a sandbox approach in health technology assessment: Benefits and recommendations”, *International Journal of Technology Assessment in Health Care*, vol. 40, no. 1, e44, 2024. DOI: 10.1017/S0266462324000412.

[13] World Bank Group, *Regulatory Sandboxes for Digital Health (English)*, 2021. [Online]. Available: http://documents.worldbank.org/curated/en/099011825011040830/P17507510c729100419bb214127f301c037 (visited on 01/28/2025).

[14] N. Pantuwong, N. Chutchomchuen, and P. Wacharawisoot, “Interactive topography simulation sandbox for geography learning course”, in *Proceedings of the 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Yogyakarta, Indonesia, 2016, pp. 1–4. [Online]. Available: 10.1109/ICITEED.2016.7863262 (visited on 02/21/2025).

[15] MeHI, *The digital health sandbox program*, 2025. [Online]. Available: https://mehi.masstech.org/programs/digital-health-sandbox-program (visited on 02/24/2025).

[16] J. Davids, M. ElSharkawy, H. Ashrafian, E. Herlenius, and N. Lidströmer, “Technical sandbox for a Global Patient co-Owned Cloud (GPOC)”, en, *BMC Digital Health*, vol. 2, no. 1, p. 75, Oct. 2024. DOI: 10.1186/s44247-024-00128-2.

[17] A. Gonzales, G. Guruswamy, and S. R. Smith, “Synthetic data in health care: A narrative review”, en, *PLOS Digital Health*, vol. 2, no. 1, pp. 1–16, Jan. 2023. DOI: 10.1371/journal.pdig.0000082.

[18] Läkemedelsverket, *Regelverk för medicintekniska produkter | Läkemedelsverket.* [Online]. Available: https://www.lakemedelsverket.se/sv/medicinteknik/regelverk (visited on 03/13/2025).

[19] U.S. Food and Drug Administration, *Fda's role in regulating medical devices*, 2020. [Online]. Available: https://www.fda.gov/medical-devices/home-use-devices/fdas-role-regulating-medical-devices (visited on 05/02/2025).

[20] European Commision, *Regulation - 2016/679 - EN - gdpr - EUR-Lex*, en. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng (visited on 03/13/2025).

[21] U. D. of Health and H. Services, *Summary of the HIPAA privacy rule*. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html (visited on 03/13/2025).

[22] Sveriges Riksdag, *Patientdatalag (2008:355)*, sv. [Online]. Available: https://www.riksdagen.se/sv/dokument-och-lagar/dokument/svensk-forfattningssamling/patientdatalag-2008355_sfs-2008-355/ (visited on 03/14/2025).

[23] EU Artifical Intelligence Act, *EU Artificial Intelligence Act | Up-to-date developments and analyses of the EU AI Act*. [Online]. Available: https://artificialintelligenceact.eu/ (visited on 05/15/2025).

[24] European Commision, *Article 57: AI Regulatory Sandboxes | EU Artificial Intelligence Act*. [Online]. Available: https://artificialintelligenceact.eu/article/57/ (visited on 02/24/2025).

[25] European Commision, *CE marking - European Commission*. [Online]. Available: https://single-market-economy.ec.europa.eu/single-market/ce-marking_en (visited on 03/13/2025).

[26] Z. He, "From Privacy-Enhancing to Health Data Utilisation: The Traces of Anonymisation and Pseudonymisation in EU Data Protection Law", *Digital Society*, vol. 2, no. 2, p. 17, Aug. 2023. DOI: 10.1007/s44206-023-00043-5.

[27] D. Susser, D. S. Schiff, S. Gerke, *et al.*, "Synthetic Health Data: Real Ethical Promise and Peril", *The Hastings Center Report*, vol. 54, no. 5, pp. 8–13, Nov. 2024. DOI: 10.1002/hast.4911.

[28] A. S. Tanenbaum and M. v. Steen, *Distributed systems: principles and paradigms*, eng, Second edtion. Upper Saddle River, NJ: Pearson, Prentice Hall, 2007.

[29] M. Goodwin, "What Is an API (Application Programming Interface)?", *IBM*, Apr. 2024. [Online]. Available: https://www.ibm.com/think/topics/api (visited on 02/21/2025).

[30] C. Guynes and J. Windsor, "Revisiting client/server computing", *Journal of Business Economics Research*, vol. 9, Jul. 2011. DOI: 10.19030/jber.v9i1.935.

[31] *What Is REST API? | IBM*, Apr. 2024. [Online]. Available: https://www.ibm.com/think/topics/rest-apis (visited on 02/21/2025).

[32] W. L. Martinez, "Graphical user interfaces", en, *WIREs Computational Statistics*, vol. 3, no. 2, pp. 119–133, Mar. 2011. DOI: 10.1002/wics.150.

[33] T. Biscontini, *Application programming interface (api)*, Salem Press Encyclopedia of Science, Nov. 2024. [Online]. Available: https://research.ebsco.com/linkprocessor/plink?id=3ef3574f-c975-3565-8026-b8024a623b63 (visited on 02/21/2025).

[34] B. S. Sarierao and A. Prakasarao, "Smart healthcare monitoring system using mqtt protocol", in *Proceedings of the 2018 3rd International Conference for Convergence in Technology (I2CT)*, Pune, India, 2018, pp. 1–5. DOI: 10.1109/I2CT.2018.8529764. (visited on 02/20/2025).

[35] Health Level Seven International, *Fhir overview*, 2025. [Online]. Available: https://www.hl7.org/fhir/overview.html (visited on 02/19/2025).

[36] Amazon Web Services, *What is a web application?*, 2023. [Online]. Available: https://aws.amazon.com/what-is/web-application/ (visited on 05/02/2025).

[37] G. Blinowski, A. Ojdowska, and A. Przybyłek, “Monolithic vs. microservice architecture: A performance and scalability evaluation”, *IEEE Access*, vol. 10, pp. 20 357–20 374, 2022. DOI: 10.1109/ACCESS.2022.3152803.

[38] M. Hussein, A. I. Galal, E. Abd-Elrahman, and M. Zorkany, “Internet of Things (IoT) Platform for Multi-Topic Messaging”, en, *Energies*, vol. 13, no. 13, p. 3346, Jun. 2020. DOI: 10.3390/en13133346.

[39] International Telecommunication Union (ITU), “Harnessing the internet of things for global development”, 2015. [Online]. Available: https://www.itu.int/en/action/broadband/Documents/Harnessing-IoT-Global-Development.pdf (visited on 02/19/2025).

[40] S. Abdulmalek, A. Nasir, W. A. Jabbar, *et al.*, “IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review”, en, *Healthcare*, vol. 10, no. 10, p. 1993, Oct. 2022. DOI: 10.3390/healthcare10101993.

[41] D. M. Mathkor, N. Mathkor, Z. Bassfar, *et al.*, “Multirole of the internet of medical things (iomt) in biomedical systems for managing smart healthcare systems: An overview of current and future innovative trends”, *Journal of Infection and Public Health*, vol. 17, no. 4, pp. 559–572, 2024. DOI: 10.1016/j.jiph.2024.01.013.

[42] A. Martikkala, A. Lobov, M. Lanz, and I. F. Ituarte, “Towards the Interoperability of IoT Platforms: A Case Study for Data Collection and Data Storage”, *IFAC-PapersOnLine*, vol. 54, no. 1, pp. 1138–1143, 2021. DOI: 10.1016/j.ifacol.2021.08.134.

[43] M. Seth, H. Jalo, Å. Högstedt, O. Medin, B. A. Sjöqvist, and S. Candefjord, “Technologies for Interoperable Internet of Medical Things Platforms to Manage Medical Emergencies in Home and Prehospital Care: Scoping Review”, en, *Journal of Medical Internet Research*, vol. 27, e54470, Jan. 2025. DOI: 10.2196/54470.

[44] F. R. di Torrepadula, A. Somma, A. De Benedictis, and N. Mazzocca, “Smart ecosystems and digital twins: An architectural perspective and a fiware-based solution”, *IEEE Software*, vol. 42, no. 2, pp. 38–46, 2025. DOI: 10.1109/MS.2024.3518752.

[45] D. Arney, J. Plourde, and J. M. Goldman, “OpenICE medical device interoperability platform overview and requirement analysis”, *Biomedical Engineering / Biomedizinische Technik*, vol. 63, no. 1, pp. 39–47, Feb. 2018. DOI: 10.1515/bmt-2017-0040.

[46] S. S. Albouq, A. A. A. Sen, N. Almashf, M. Yamin, A. Alshanqiti, and N. M. Bahbouh, "A survey of interoperability challenges and solutions for dealing with them in iot environment", *IEEE Access*, vol. 10, pp. 36 416–36 428, 2022. DOI: 10.1109/ACCESS.2022.3162219.

[47] World Heath Organization, *Classifications and terminologies*. [Online]. Available: https://www.who.int/standards/classifications (visited on 03/17/2025).

[48] SNOMED International, *SNOMED CT maps*. [Online]. Available: https://www.snomed.org/maps (visited on 03/17/2025).

[49] Regenstrief Institute, *What loinc is*. [Online]. Available: https://loinc.org/get-started/what-loinc-is/ (visited on 05/16/2025).

[50] M. Ayaz, M. F. Pasha, M. Y. Alzahrani, R. Budiarto, and D. Stiawan, "The Fast Health Interoperability Resources (FHIR) Standard: Systematic Literature Review of Implementations, Applications, Challenges and Opportunities", *JMIR Medical Informatics*, vol. 9, no. 7, e21929, Jul. 2021. DOI: 10.2196/21929.

[51] J. C. Mandel, D. A. Kreda, K. D. Mandl, I. S. Kohane, and R. B. Ramoni, "SMART on FHIR: A standards-based, interoperable apps platform for electronic health records", *Journal of the American Medical Informatics Association*, vol. 23, no. 5, pp. 899–908, Sep. 2016. DOI: 10.1093/jamia/ocv189.

[52] HL7 International, *App Launch: Launch and Authorization - SMART App Launch v2.2.0*, 2024. [Online]. Available: https://build.fhir.org/ig/HL7/smart-app-launch/app-launch.html (visited on 05/16/2025).

[53] HL7 Boston Children's Hospital., *Overview*. [Online]. Available: https://cds-hooks.org/ (visited on 03/17/2025).

[54] HL7 Boston Children's Hospital, *1.0 - cds hooks overview*. [Online]. Available: https://cds-hooks.hl7.org/1.0/#providing-fhir-resources-to-a-cds-service (visited on 03/18/2025).

[55] *Ergonomics of human-system interaction – part 210: Human-centred design for interactive systems*, ISO 9241-210:2019, Stockholm, Sverige, 2019. [Online]. Available: https://www.sis.se/produkter/informationsteknik-kontorsutrustning/terminalutrustning-och-ovrig-kringutrustning/ss-en-iso-9241-2102019/.

[56] D. A. Norman, *The design of everyday things*, Revised and expanded edition. New York, US: Basic Books, 2013, ISBN: 9780465050659.

[57] T. Green and J. Labrecque, *A Guide to UX Design and Development: Developer's Journey Through the UX Process* (Design Thinking). Berkeley, CA: Apress, 2023, ISBN: 9781484295755.

[58] J. Tidwell, C. Brewer, and A. Valencia, *Designing interfaces: patterns for effective interaction design*, Third edition. Beijing [China] ; North Sebastopol, CA: O'Reilly, 2020, ISBN: 9781492051961.

[59] L. Nielsen, *Personas: user focused design* (Human-computer interaction series 15). New York, US: Springer, 2013, ISBN: 9781447140849.

[60] T. Lowdermilk, *User-centered design*, eng, 1st ed. Sebastopol, CA: O'Reilly Media, 2013, ISBN: 9781449359805.

[61] A. Cooper, R. Reimann, D. Cronin, C. Noessel, J. Csizmadi, and D. LeMoine, *About face: the essentials of interaction design*, Fourth edition. Indianapolis, Indiana: Wiley, 2014, ISBN: 9781118766583.

[62] H. Jung, E. Stolterman, W. Ryan, T. Thompson, and M. Siegel, "A Study on Understanding of UI and UX", *International Journal of Applied Engineering Research*, vol. 12, no. 20, pp. 9675–9685, 2017, ISSN: 0973-4562.

[63] G. Goel, P. Tanwar, and S. Sharma, "Ui-ux design using user centred design (ucd) method", in *Proceedings of the 2022 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, Jan. 2022, pp. 1–8. DOI: 10.1109/ICCCI54379.2022.9740997.

[64] H. Rex and P. Pardha, "28.6.2 rules of thumb abound", in *UX Book - Agile UX Design for a Quality User Experience (2nd Edition)*, 2nd. Elsevier, 2019, ISBN: 978-0-12-805342-3.

[65] *ISO 9241-11:2018 - Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts*, ISO 9241-11:2018, Stockholm, Sweden, 2018. [Online]. Available: https://www.iso.org/obp/ui/en/#iso:std:iso:9241:-11:ed-2:v1:en.

[66] J. Rubin, *Handbook of usability testing: how to plan, design, and conduct effective tests*, 2nd ed. Indianapolis, US: Wiley Pub, 2008, ISBN: 9780470185483.

[67] E. Norlin and C. Winters, *Usability testing for library web sites : a hands-on guide*. Chicago, US: American Library Association, 2002, ISBN: 0838935117.

[68] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces", in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Seattle, Washington, USA: Association for Computing Machinery, Mar. 1990, pp. 249–256. DOI: 10.1145/97243.97281.

[69] Bulma, *Overview of Bulma*. [Online]. Available: https://bulma.io (visited on 05/16/2025).

[70] Bootstrap, *Get started with Bootstrap*. [Online]. Available: https://getbootstrap.com/docs/5.3/getting-started/introduction/ (visited on 05/16/2025).

[71] Tailwindcss, *Get started with tailwind css*. [Online]. Available: https://tailwindcss.com/docs/installation/using-vite (visited on 05/16/2025).

[72] I. Benbasat and P. Todd, "An experimental investigation of interface design alternatives: Icon vs. text and direct manipulation vs. menus", *International Journal of Man-Machine Studies*, vol. 38, no. 3, pp. 369–402, 1993. DOI: 10.1006/imms.1993.1017.

[73] K. Gourab, "An electronic medical record simulator and sandbox for informatics education", in *Proceedings of the 2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, Orlando, FL, USA, 2024, pp. 737–738. DOI: 10.1109/ICHI61247.2024.00117.

[74] Epic on Fhir, *Home - Epic on FHIR*. [Online]. Available: https://fhir.epic.com/ (visited on 02/24/2025).

[75] Centre for Healthcare Innovation, *About CHISEL*. [Online]. Available: https://www.chi.sg/platformprogrammes/chisel/ (visited on 01/28/2025).

[76] Stanford Univeristy, *Spezi features*. [Online]. Available: https://spezi.stanford.edu/spezi-features (visited on 02/24/2025).

[77] Git, *Git - About Version Control*, 2025. [Online]. Available: https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control (visited on 05/16/2025).

[78] S. Chaudhuri, H. Thompson, and G. Demiris, "Fall Detection Devices and Their Use With Older Adults: A Systematic Review", *Journal of Geriatric Physical Therapy*, vol. 37, no. 4, pp. 178–196, Oct. 2014. DOI: 10.1519/JPT.0b013e3182abe779.

[79] J. Fleming and C. Brayne, "Inability to get up after falling, subsequent time on floor, and summoning help: Prospective cohort study in people over 90", *The BMJ*, vol. 337, a2227, Nov. 2008. DOI: 10.1136/bmj.a2227.

[80] U. Björner, private communication, Jan 2025.

[81] R. W. Broadley, J. Klenk, S. B. Thies, L. P. J. Kenney, and M. H. Granat, "Methods for the Real-World Evaluation of Fall Detection Technology: A Scoping Review", *Sensors*, vol. 18, no. 7, p. 2060, Jun. 2018. DOI: 10.3390/s18072060.

[82] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches", *Neurocomputing*, vol. 100, pp. 144–152, 2013. DOI: https://doi.org/10.1016/j.neucom.2011.09.037.

[83] D. J. Warrington, E. J. Shortis, and P. J. Whittaker, "Are wearable devices effective for preventing and detecting falls: An umbrella review (a review of systematic reviews)", en, *BMC Public Health*, vol. 21, no. 1, p. 2091, Dec. 2021. DOI: 10.1186/s12889-021-12169-7.

[84] 8 West Consulting, *Remote Patient Monitoring | IoT Healthcare Device | SafeTrx Care*. [Online]. Available: https://www.safetrxapp.com/products/safetrx-care/ (visited on 03/11/2025).

[85] Europe, Sony Network Communications, *The technology behind the mSafety solution | mSafety from Sony*. [Online]. Available: https://www.sonynetworkcom.com/msafety/solution (visited on 03/26/2025).

[86] Skyresponse AB, *Technical Integration Report for SafeTrx and Skyresponse System*, unpublished.

[87] B. A. Medical, *SOS Micro: Mobile Medical Alert*. [Online]. Available: https://www.bayalarmmedical.com/medical-alert-system/sos-micro/ (visited on 03/12/2025).

[88] Essence SmartCare, *ES700EPAJ Emergency Pendant Advanced User Manual*, 2021. [Online]. Available: https://manuals.plus/essence/es700epaj-emergency-pendant-advanced-manual (visited on 05/05/2025).

[89] E. Smartcare, *Indoor Radar-based Fall Detection for Seniors | Essence*. [Online]. Available: https://www.essencesmartcare.com/products/mdsense-fall-detector/ (visited on 03/12/2025).

[90] KamiVision, *Kami Cloud*. [Online]. Available: https://kamihome.com/products/fall-detect/ (visited on 03/12/2025).

[91] Z. Jiang, M. A. A. Al-Qaness, D. Al-Alimi, *et al.*, "Fall detection systems for internet of medical things based on wearable sensors: A review", *IEEE Internet of Things Journal*, vol. 11, no. 21, pp. 34 797–34 810, 2024. DOI: 10.1109/JIOT.2024.3421336.

[92] Nuxt, *Nuxt: The Progressive Web Framework*. [Online]. Available: https://nuxt.com/ (visited on 04/30/2025).

A

Appendix: GUI for Technical Users

A.1 Sketching & Wireframes Results

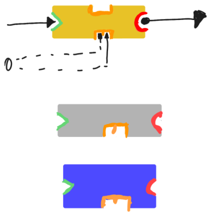
The sketch prototyping resulted in three distinct alternatives, each representing a different approach to constructing the simulation. The characteristics of the sketches is presented in Table A.1 and the sketches are visualized in Figure A.1.

Prototype	**A**	**B**	**C**
Page Layout	Two pages	Single page	Single page
Workspace Style	Free placement with drag-and-drop	Fixed sections (3 columns)	Tree structure with no free workspace
Component Selection	In first page then appear in side menu (left)	Top menu bar	Drop-down menus

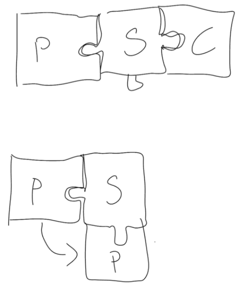
Table A.1: Summary of key characteristics for the three sketch prototypes: A, B and C for the technical interface.

Figure A.1: Sketches of GUI, including three different prototypes: A,B and C for the technical interface.

Different sketches illustrating potential component designs were also created and two are shown in Figure A.2.



(a) Sketch of components as blocks with connections in various shapes and colors to symbolize data transfers and requests.



(b) Sketch of components represented as puzzle pieces.

Figure A.2: Component sketches exploring different shape designs and strategies for visualizing connections.

Table A.2 and A.3 shows an alternative for block design used to indicate type of component and port.

Component	Color
Provider	Blue
Service	Yellow
Consumer	Grey

Table A.2: Components and representing color.

Port	Color	Shape
Input	Green	Triangular
Request & Response	Orange	Rectangular
Output	Red	Semicircular

Table A.3: Port and representing color and shape.

Figure B.2 - B.4 presents the wireframes of prototype A,B and C.

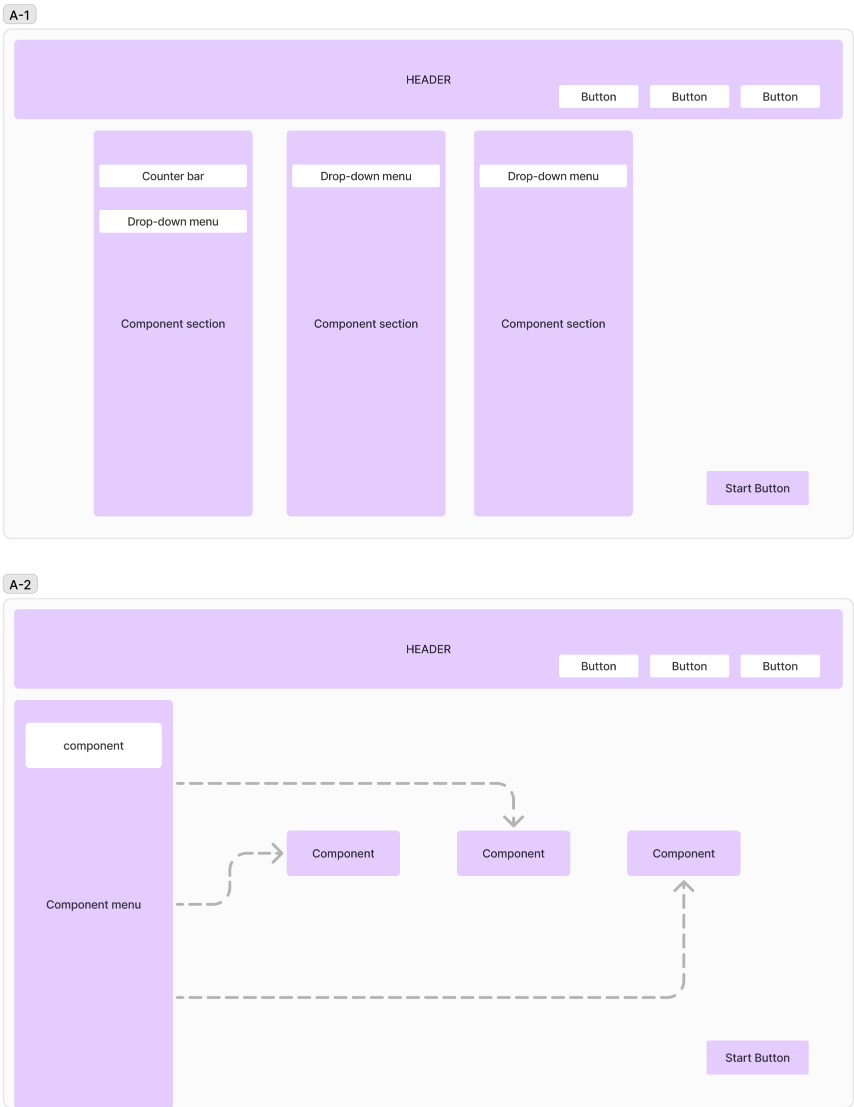


Figure A.3: Wireframe of Alternative A for the technical interface, shown in purple. Illustrating the two-page GUI for simulation creation.

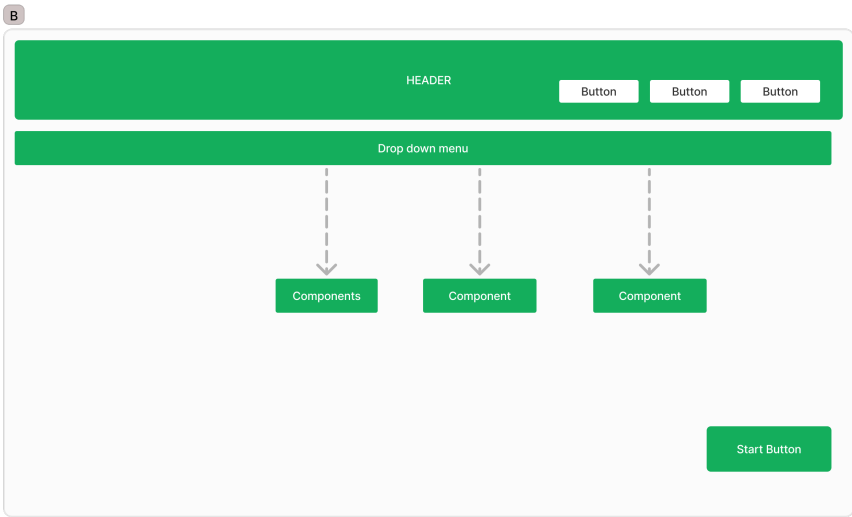


Figure A.4: Wireframe of Alternative B for the technical interface, shown in green. Illustrating the two-page GUI for simulation creation.

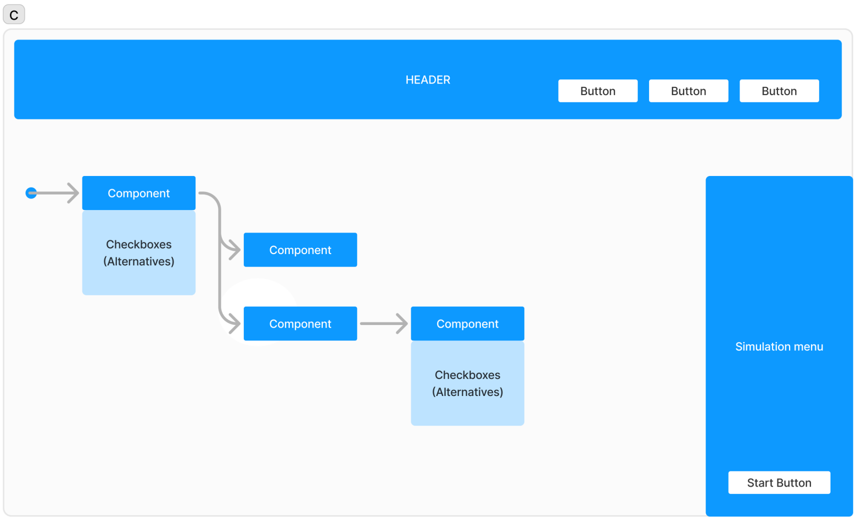


Figure A.5: Wireframe of Alternative C for the technical interface, shown in blue. Illustrating the two-page GUI for simulation creation.

An evaluation of the three design suggestions can be seen in table A.4.

Wireframe:	A	B	C
Strengths	Simplicity, flexibility with drag and drop components	Flexible workspace, no need for navigating to another page	Good for unexperienced users
Drawbacks	Need to navigate between pages, harder to add new components afterward	Harder for unexperienced users to draw connections	More restricted workspace
Comments	Nice to set up the structure before, easy to see that all components have been inserted in the working space	Everything easily accessible in menu field	Easy with good compatibility between components

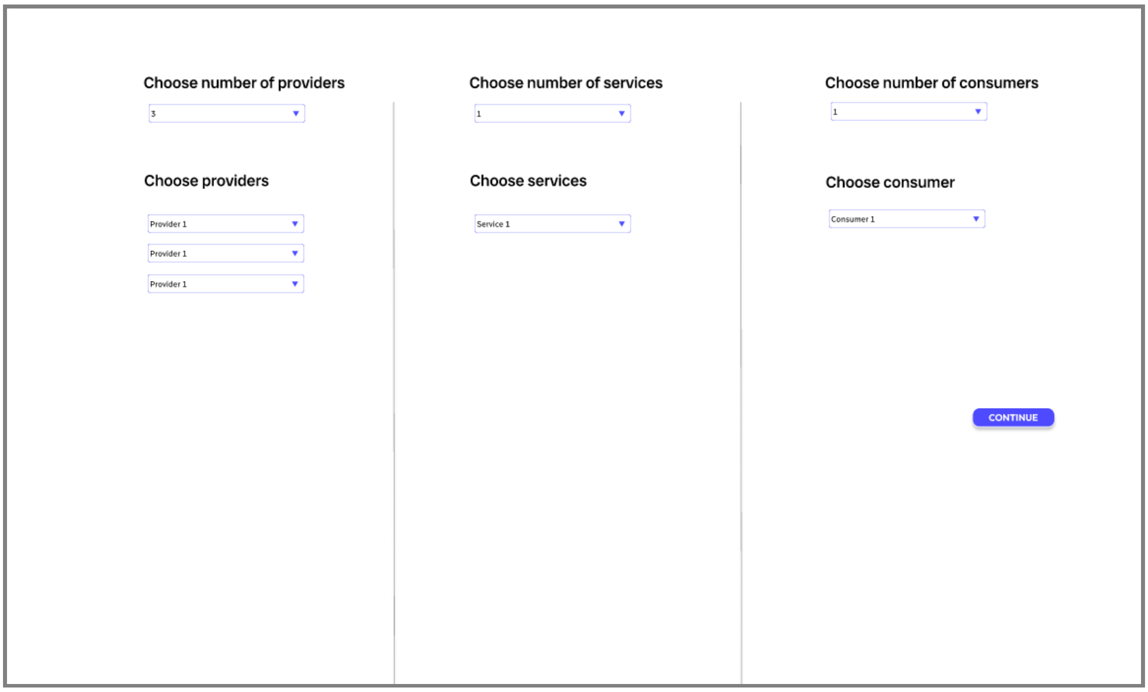
Table A.4: Evaluation of wireframe prototypes for the technical interface.

A.2 A/B Testing Result

The A/B testing was conducted comparing three mockups: A, B and C. The key characteristics of the mockups are summarized in Table A.5 and visual representation of each mockup are shown in Figures A.6, A.7 and A.8.

Mockup:	**A**	**B**	**C**
Page Layout	Two pages	Single page	Single page
Workspace Style	Free placement with drag-and-drop	Fixed sections (3 columns)	Tree structure with no free workspace
Component selection	In first page then appear in side menu (left)	Top menu bar	Drop-down menus
Component connection	Click on connection ports	Click on toggle bar, then target components	Menu-based connection via drop-downs
Connection Style	Distinct ports for data input, request, and output; numbers indicate data transfer order	Solid lines for input/output, dashed lines for requests	Menu-based, created when component is selected
Settings Access	Via double-click on component	Through the side menu	Displayed directly beneath the component

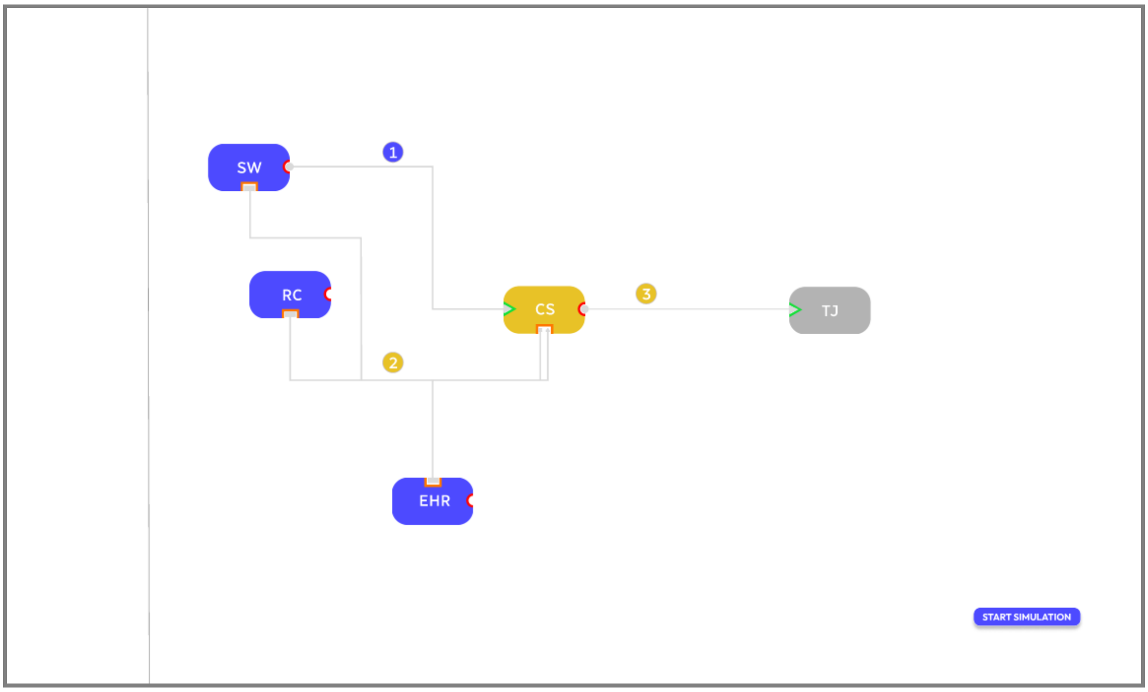
Table A.5: Summary of key characteristics for the three mockup prototypes: A, B and C for the technical interface.



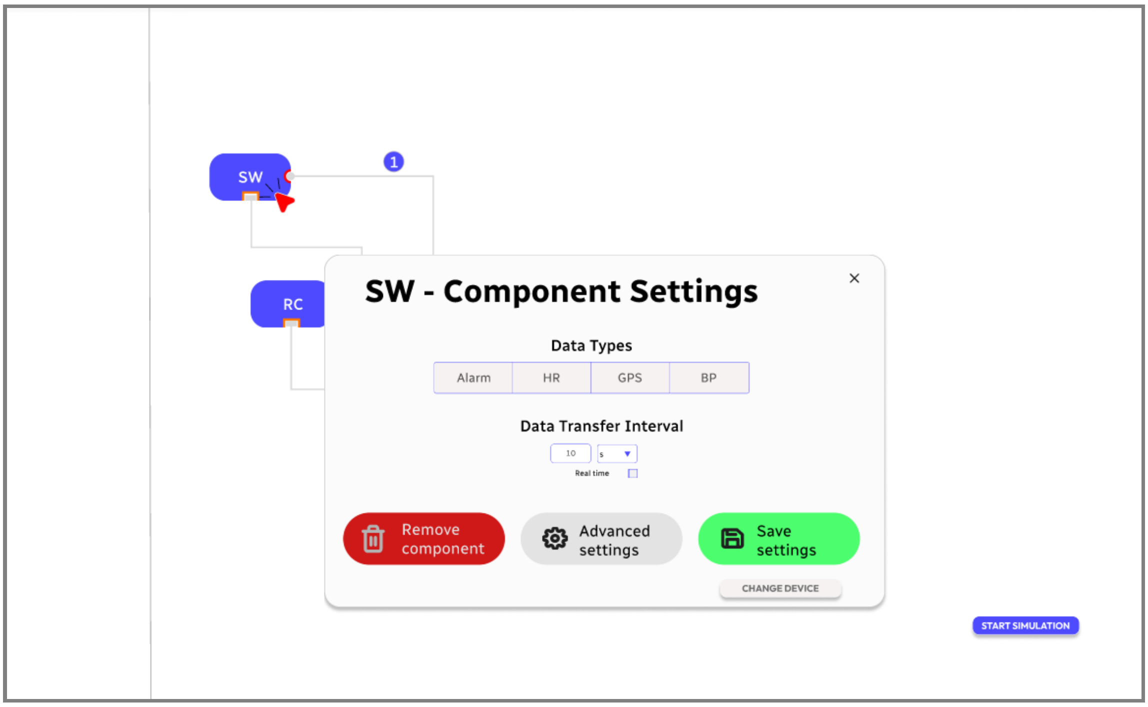
(a) First page of the GUI. The user selects the number of components from a drop-down menu, after which the corresponding number of component selection menus is displayed.

(b) Initial view of the workspace page. The selected components appear on the left side of the window, and a blank workspace is displayed on the right.

Figure A.6: Mockup alternative A, for the technical interface, with two pages and an open workspace.



(c) Workspace with a completed simulation setup. Components are placed using drag-and-drop and connected by clicking and dragging from the component ports.



(d) Visualization of the settings pop-up menu for a component, opened by clicking on the component.

Figure A.6: Mockup alternative A, for the technical interface, with two pages and an open workspace.

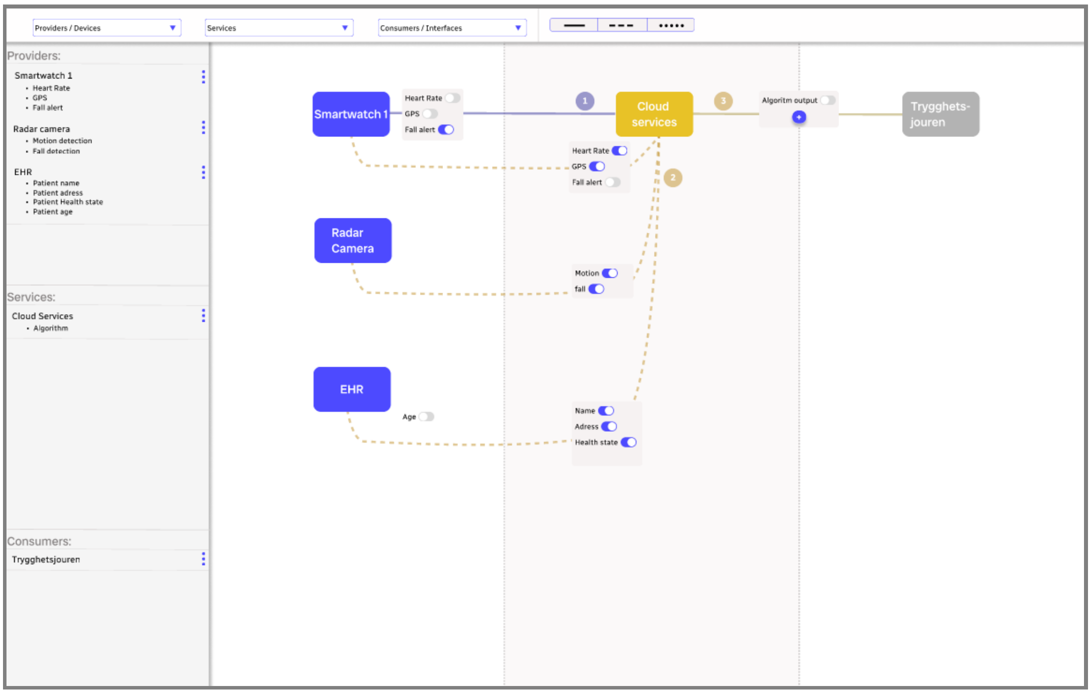
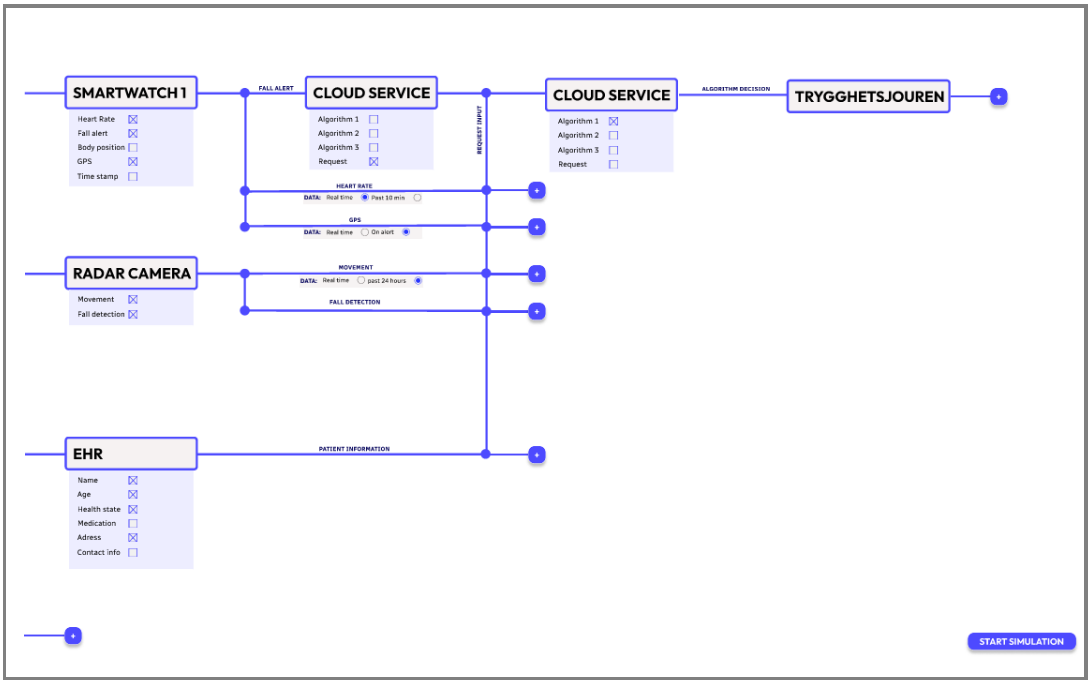


Figure A.7: Mockup of Alternative B, for the technical interface. Featuring a single-page layout and a workspace divided into three columns. Selected components appear in the corresponding workspace column and are connected using the toggle bar and click interactions.

(a) Initial empty workspace with drop-down menus for selecting component.

Figure A.8: Mockup of alternative C for the technical interface. Featuring a single-page layout and with a tree structured workspace.



(b) Finalized simulation setup.

Figure A.8: Mockup of alternative C for the technical interface. Featuring a single-page layout and with a tree structured workspace.

The test was evaluated using a questionnaire in which participants ranked the ease of understanding different design concepts. A score of 1 represents 'Hard' and 5 represents 'Very Simple'. The results for the five participants are presented in Table A.6.

	1	**2**	**3**	**4**	**5**	**Total Points**	**Mean Value**	**Median**
A								
Choose and add components	0	0	1	3	1	15	3	3
Change Settings of components	0	0	2	3	0	18	3,6	4
Connect Components	0	1	3	0	1	16	3,2	3
Chnage transfered data types?	0	1	1	3	0	17	3,4	4
Summation	0	2	7	9	2	66	3,3	3,5
B								
Choose and add components	0	0	0	2	3	23	4,6	5
Change Settings of components	0	0	0	4	1	21	4,2	4
Connect Components	0	0	1	2	2	21	4,2	4
Chnage transfered data types?	0	1	0	2	2	20	4	4
Summation	0	1	1	10	8	85	4,25	4
C								
Choose and add components	0	0	1	2	2	21	4,2	4
Change Settings of components	0	0	1	1	3	22	4,4	5
Connect Components	0	0	1	1	3	22	4,4	5
Chnage transfered data types?	0	0	2	2	1	19	3,8	4
Summation	0	0	5	6	9	84	4,2	4,5

Table A.6: Results from the A/B testing of mockup prototypes for the technical interface.

Participants were asked to select their most and least preferred design. The results are presented in Table A.7.

	A	B	C
Most preferred	0	2	3
Least preferred	3	0	2

Table A.7: Table showing the participants most and least preferred mock-up alternative for the technical interface.

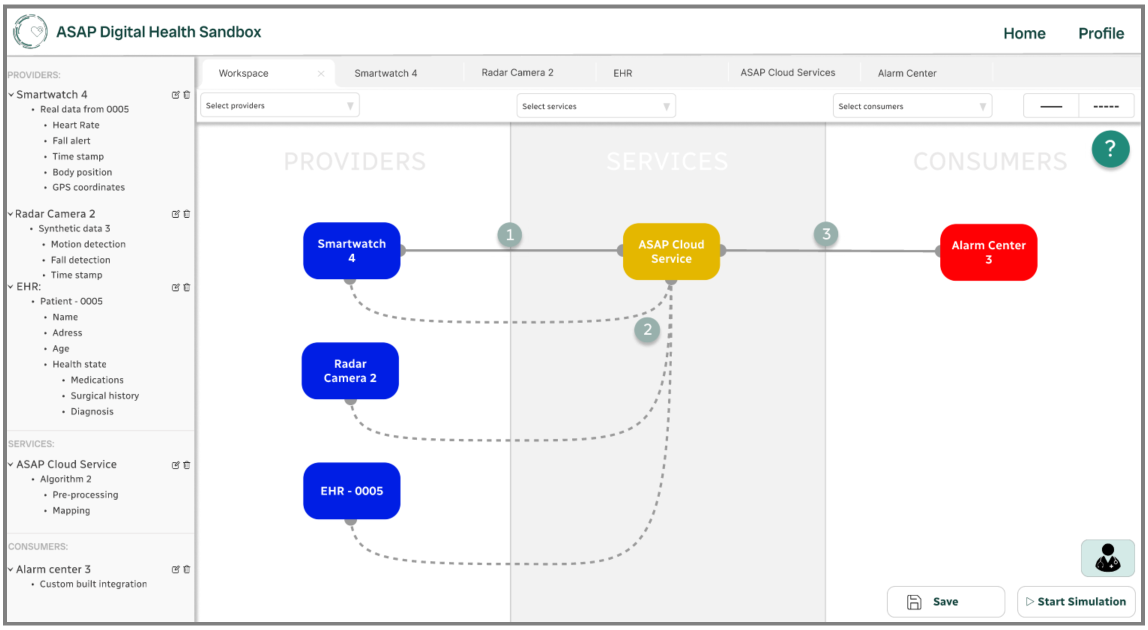
Some comments from test participants, offering important insights, are presented in Table A.8 below

Preferred design	Comment
Alt. C	*I like alternative C, But i also like the component menu in alternative A.*
Alt. C	*I think a combination of B and C would be easier to understand, I liked the concept with the side meny on b. This would make it easier to understand how to change the settings and types and so on.*
Alt. B	*It feels like alternative A is easier to connect the components and the layout of B is more intuitive with every choice on the same page. So this combination would be nice.*
Alt. C	*I like C best, it is similar to many other programs i worked in.*

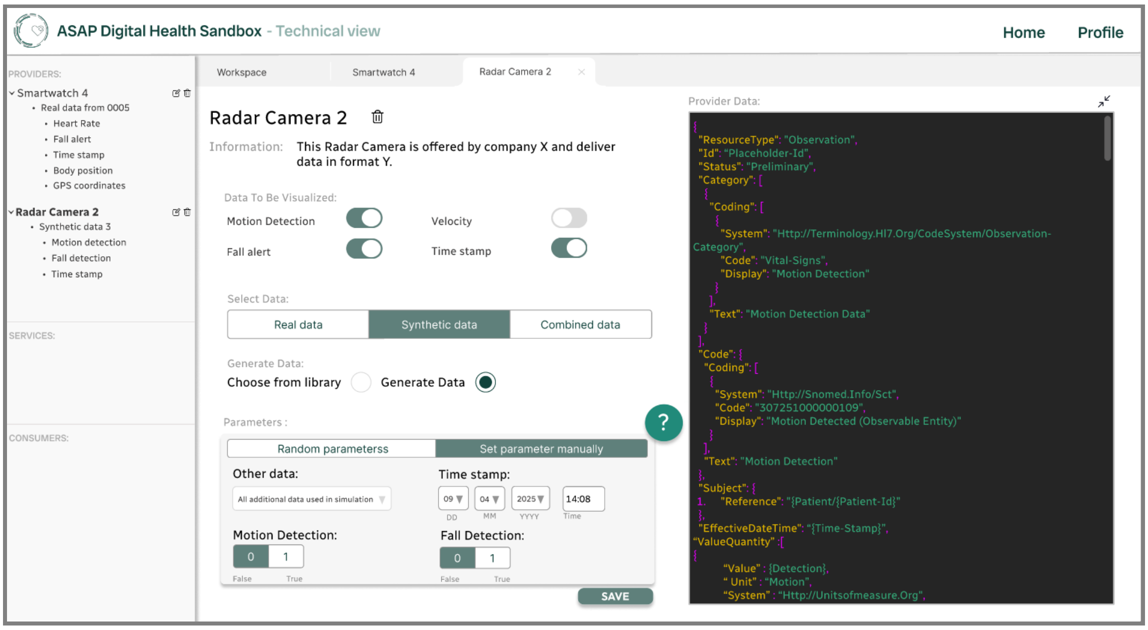
Table A.8: Feedback from A/B test participants for the technical interface.

A.3 Task Oriented Test Result

The task-oriented test was conducted on an interactive prototype. The main pages are presented in Figure A.9

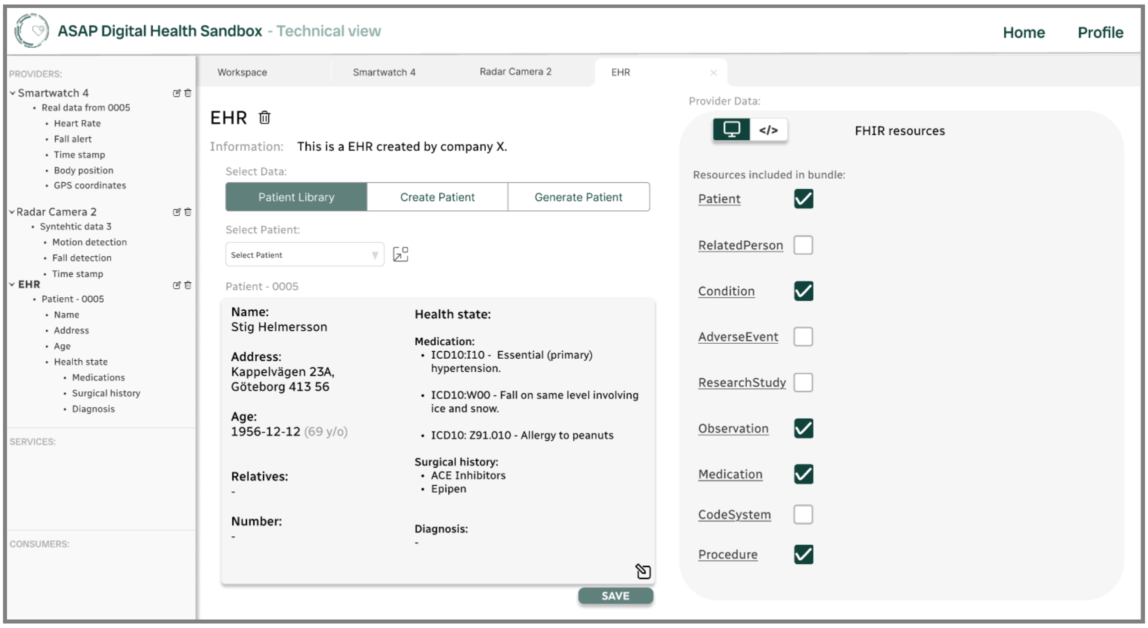


(a) Workspace page, displaying a final simulation setup.

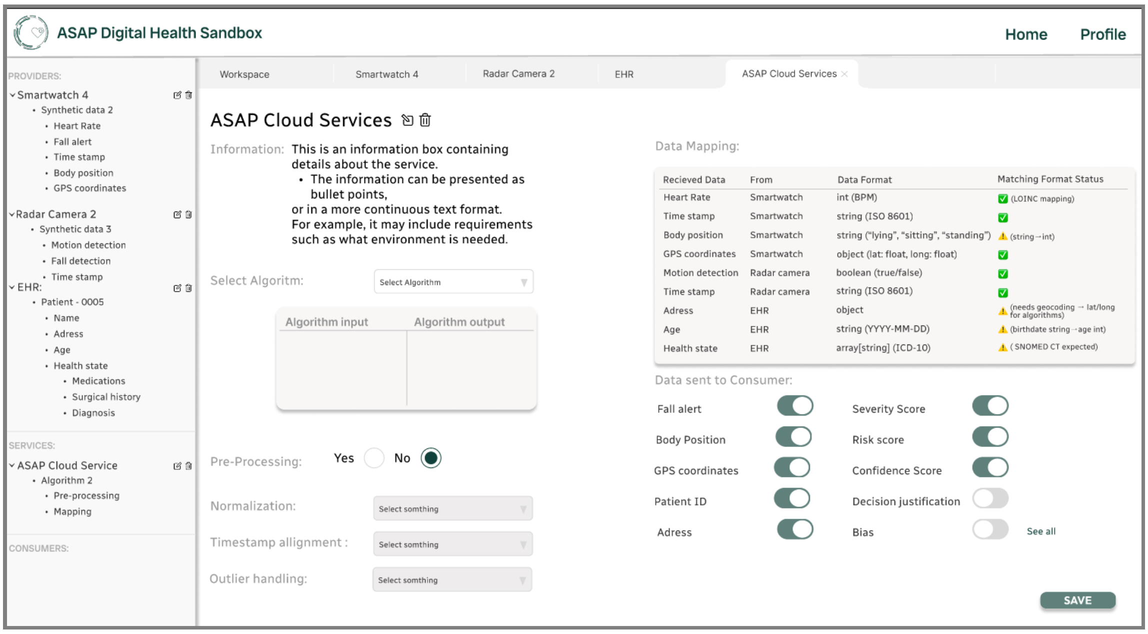


(b) Provider page, displaying how to select data and adjust settings for an added device.

Figure A.9: Interactivity prototype of technical interface used for the task oriented test.

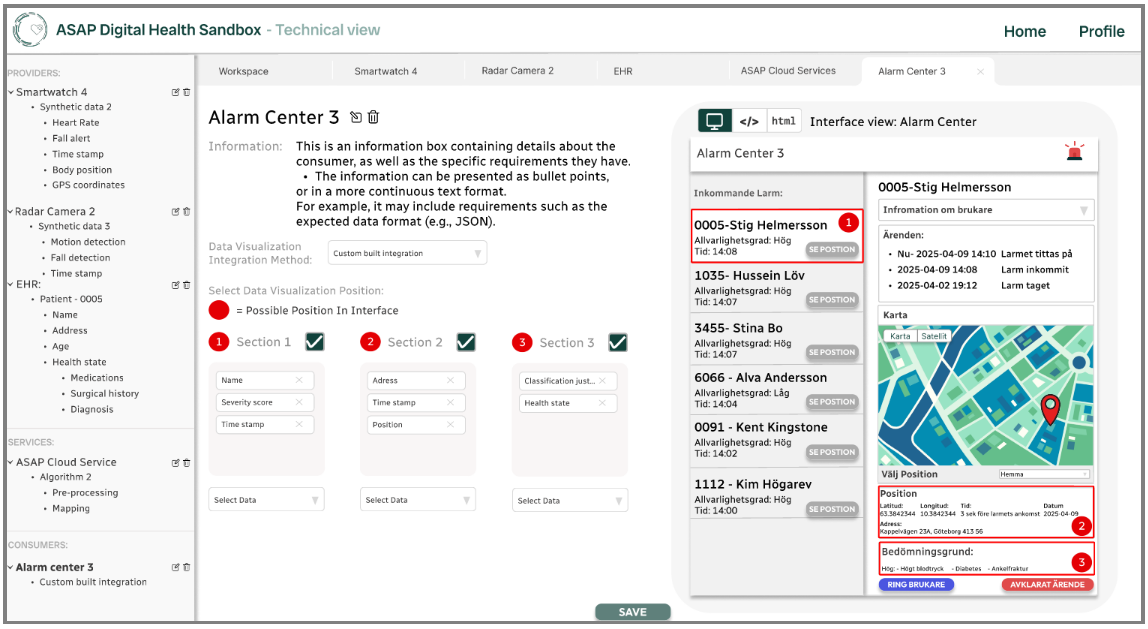


(c) Provider page, displaying how to select data adjust settings for an added EHR.



(d) Service page, displaying how select algorithm, perform data mapping and pre-processing of data.

Figure A.9: Interactivity prototype of technical interface used for the task oriented test.



(e) Consumer page, displaying how to integrate data into the end users interface.

Figure A.9: Interactivity prototype of technical interface used for the task oriented test.

The results from the task-oriented test include metrics on task duration and the occurrence of misclicks. Participants were also invited to provide feedback on aspects of the design that they either liked or disliked. The results are presented in Table A.9.

Task Time	**Misclick**	**Comments**
7 min 22 sec	2	*It's hard to tell that you've reached the clinical view. I also found the lines connecting the devices a bit confusing, could it maybe be a single line that represents both data transfer and requests?*
15 min 8 sec	8	*I thought you could click directly on the corresponding workspace to add components. It wasn't very clear that the down arrow represented a data request.*
11 min 42 sec	3	*Hard to find how to switch to the clinical view. Hard to understand how the connect the components. Would prefer to click and drag between them. Not clear how to choose what data that is transferred between components. Did not like that the service component was red. Felt like an error.*
9 min 40 sec	3	*It was a bit confusing how to connect the boxes and the options that was available for it. Drawing a line by dragging between boxes would been more intuitive. It was also hard to find the clinical view and understand how to change transferred data.*
9 min 55 sec	2	*It is easy to add components, though adding connections can be a bit challenging initially. However, once you become familiar with the process, I believe it will work smoothly. It would be helpful to have more descriptive instructions explaining what each element does. While most of the tasks were easy to complete, I didn't fully understand the purpose of some steps along the way.*

Table A.9: Test result from task oriented test on interactivity prototype of the technical interface.

A.4 Test Scripts

Below the test scripts used in the A/B test and task oriented test are visualized.

Test and Mockups: Technical User

This design project aims to develop a web-based interface for a digital health sandbox. A Sandbox is a test environment where users can simulate real-world settings. In this case, a simulation of a fall detection setup will be created. The case simulated is shown in figure 1. The case includes data from a smartwatch, a radar camera and an electronic health record (EHR) that is sent to a cloud service where an algorithm uses the data for deciding if the Alarm center should be noted/contacted/alarmed and proceed with further actions to help a patient.

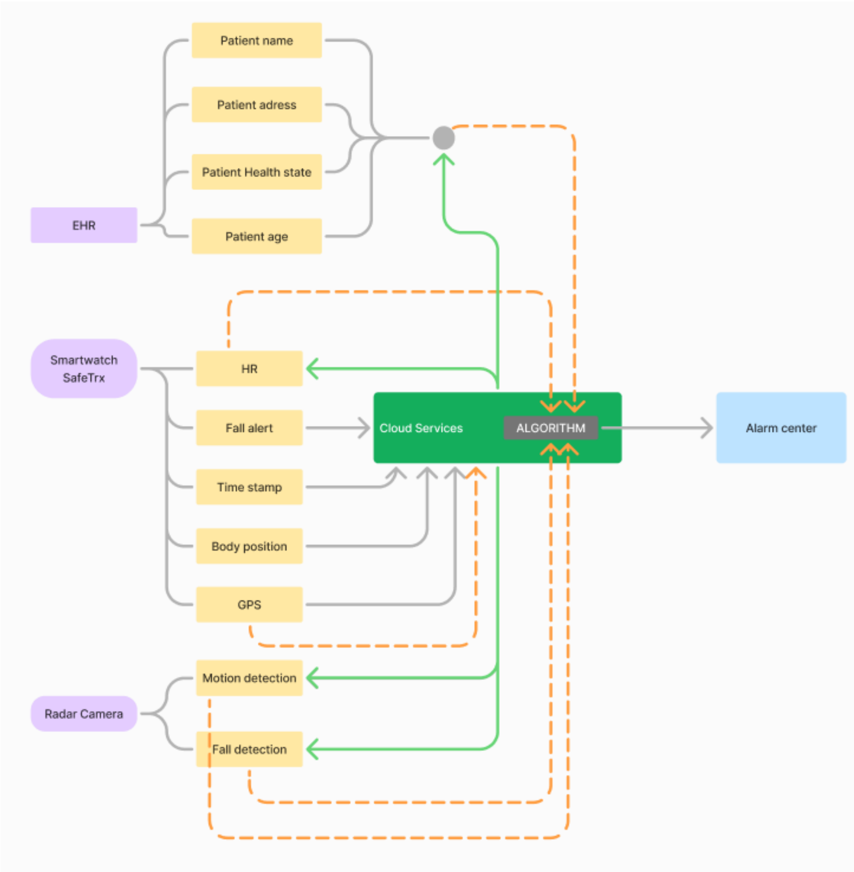


Figure 1: Simulation case used in the GUIs

In this test, you will be presented three different examples of graphical user interfaces (GUIs) concepts that illustrates how the user can interact with the system to build a simulation based on the case in 1. The simulation itself has already been constructed in the last image - you only need to observe the interfaces and asses how intuitive and easy they are to understand. The purpose of this test is to evaluate and compare different GUI design concepts. The interfaces are presented as static images of a web page with limited interactivity such as clickable drop down menus and pop-up windows displaying additional information.

We want you to review the functionality of each GUI and its components from the questions provided in the form. Please be as specific as possible when explaining why you like or dislike about each design. You will review the GUIs one by one, but are allowed to go back and forth between them. You can fill in the form either while reviewing a specific interface or after you have seen them all, depending on your preference. Focus on the functionality and how different actions work, such as selecting a new component or connecting devices. Try avoid commenting on visual aspects like color, font, or overall aesthetics.

If you have any questions about the case, the GUIs or something else fell free to ask.

Approximate time: 15 minutes.

Test of Interactivity prototypes

This design project aims to develop a web-based interface for a digital health sandbox. A Sandbox is a test environment where users can simulate real-world settings. In this case, a simulation of a fall detection setup will be created. The case simulated is shown in figure 1.

The case includes data from a smartwatch, a radar camera and an electronic health record (EHR) that is sent to a cloud service where an algorithm uses the data for deciding if the Alarm center should be noted/contacted/alarmed and proceed with further actions to help a patient.

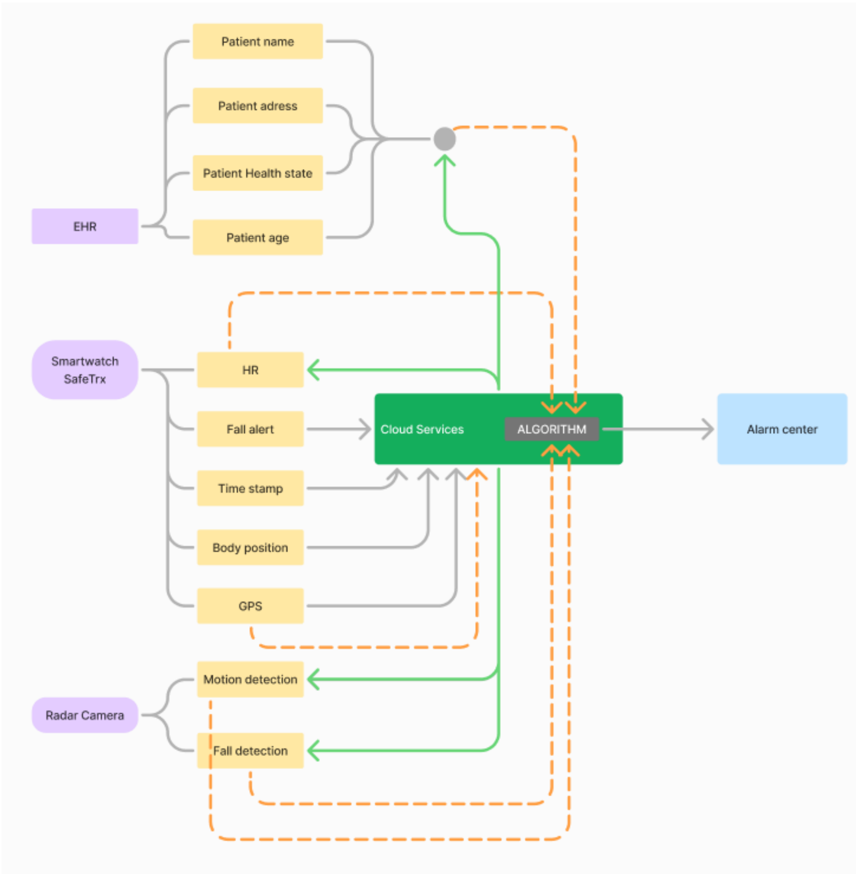


Figure 1: Simulation case used in the GUI

You will be asked to perform tasks with the goal of building the simulation setup shown in Figure 1.

Approximate time: 15 min

Task Set 1

1. *Are you able to change to a clinical view?*
2. Go to the technical view
3. Add a provider called **Smartwatch 4**.
4. Edit the Smartwatch.
5. Select **Real Data & patient 0005**.
6. *Can you close the code box?*
7. Return to the workspace.
8. Add a provider called **Radar Camera 2**.
9. Edit the Radar Camera.
10. Select **Synthetic Data**.
11. Return to the workspace.
12. Add a provider called **EHR**.
13. Edit the EHR.
14. Select a patient from the **library**, then choose **patient 0005**.
15. *Are you able to find the code view?*
16. Return to the workspace.

Task Set 2

1. Add a service called **ASAP Cloud Service**.
2. Connect the Smartwatch to the ASAP Cloud Service with a line representing **data transfer**.
3. Connect the ASAP Cloud Service to indicate that data is **requested** from the following, in order:
   - Smartwatch 4
   - Radar Camera 2
   - EHR
4. *Do you understand how to select which data is transferred/requested?*
5. Edit the ASAP Cloud Service.
6. Select **Algorithm 1**.
7. Choose **No pre-processing**.
8. In the data mapping box on the right, you can check whether the formats match. Click the warning icon for **health state** and accept the suggested fix.
9. Return to the workspace.

Task Set 3

1. Add a consumer called **Alarm Center 3**.
2. Connect the ASAP Cloud Service to the Alarm Center with a line representing **data transfer**.

3. Edit the Alarm Central.
4. Select **Custom-Built Integration**.
5. Select all three sections.
6. For each section, select:
    - Section 1:
        - Name
        - Severity score
        - Timestamp
    - Section 2:
        - Address
        - Timestamp
        - Position
    - Section 3:
        - Classification justification
7. *Do you understand how to view the HTML and FHIR code?*
8. Return to the workspace.
9. **Save** the simulations setup
10. **Start** the Simulation

Thank you for your time!

B

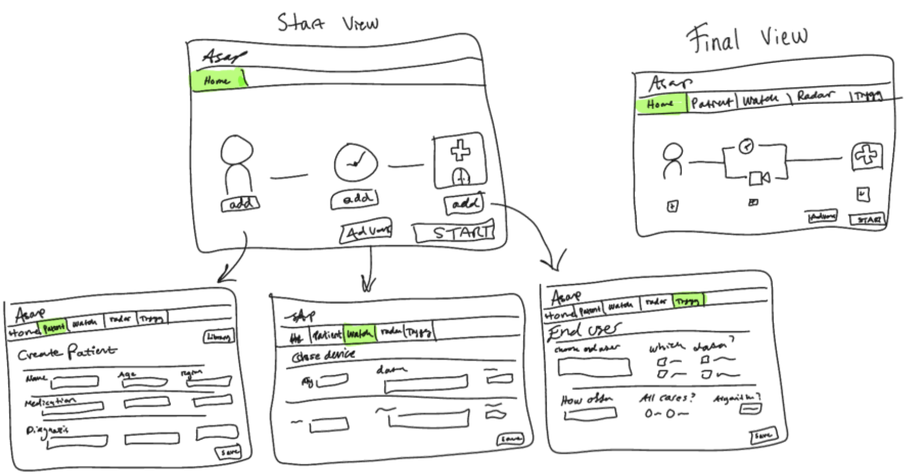
Appendix: GUI for Clinical Users

B.1 Sketching & Wireframes Result

The sketch prototyping resulted in three distinct alternatives, each representing a different approach to constructing the simulation. The characteristics of the sketches is presented in Table B.3 and the sketches are visualized in Figure B.1.

Prototype	**A**	**B**	**C**
Page Layout	Single-page layout with tabs.	Five-page layout	Three-page layout
Layout for Simulation Setup	Visual workspace with auto-connected flow and example preview.	Simple pages with brief inputs per step.	Compact pages with detailed input fields.
Simulation Order	Flexible: components can be added in any order.	Sequential: components must be added in a specific order.	Sequential: components must be added in a specific order.

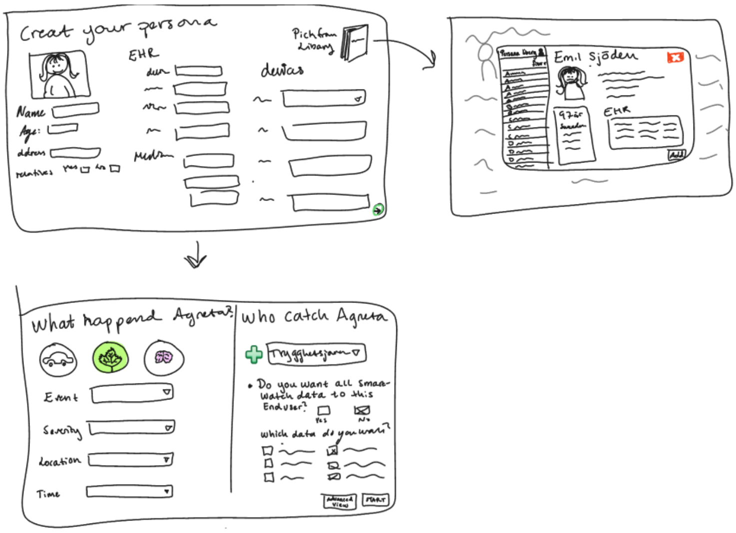
Table B.1: Summary of key characteristics for the three sketch prototypes: A, B and C of the clinical interface.



(a) Alternative A: Single-page layout with tabs.

Figure B.1: Sketches of GUI, including three different prototypes: A,B and C of the clinical interface.

(b) Alternative B: Multi-page layout with detailed inputs per step.



(c) Alternative C: Three-paged layout with breif inputs per step.

Figure B.1: Sketches of GUI, including three different prototypes: A,B and C of the clinical interface.

Figure B.2 - B.4 presents the wireframes of prototype A,B and C.

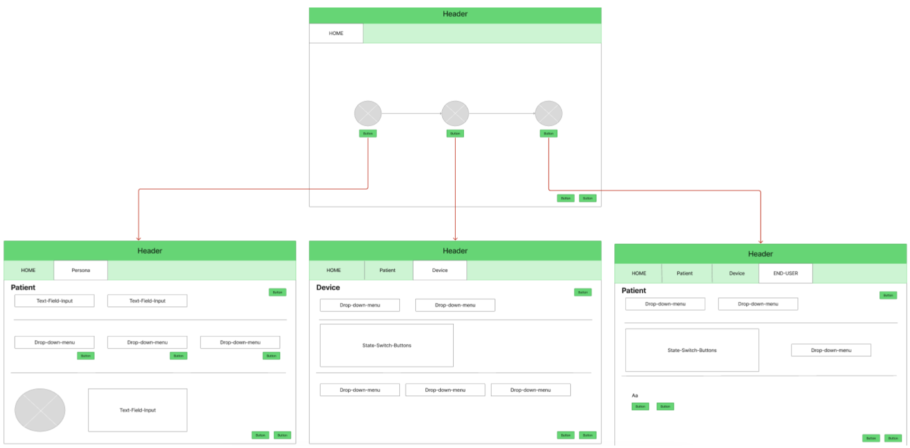


Figure B.2: Wireframe of alternative A of the clinical interface, shown in green. Illustrating the single page GUI with tabs for simulation creation.

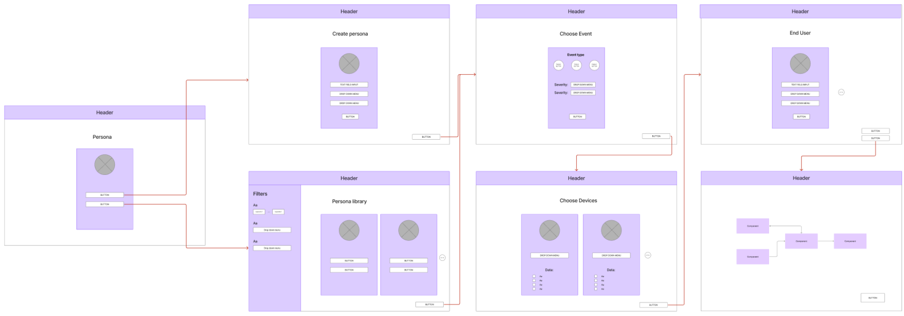


Figure B.3: Wireframe of Alternative B of the clinical interface, shown in purple. Illustrating the multi-page GUI for simulation creation.

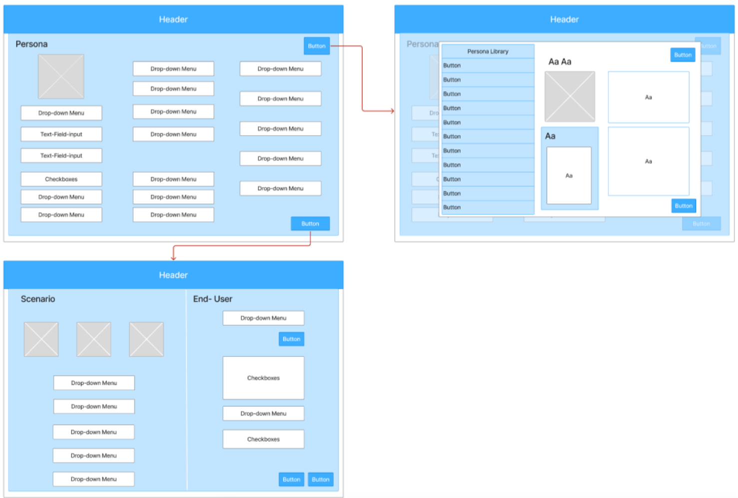


Figure B.4: Wireframe of Alternative C of the clinical interface, shown in blue. Illustrating the three-paged GUI for simulation creation.

An evaluation of the three design suggestions can be seen in table B.2.

Wireframes:	A	B	C
Strengths	Easy to follow the flow of signals and access settings trough tabs.	Hard to miss any step since guided trough everything	Few pages, minimizing risk of going forth and back.
Drawbacks	Need to navigate trough several tabs to change settings.	Need to go back and forth trough many pages if mistakes is made early.	A lot of information on same page.
Comments	Nice to have a view of the flow between patient, devices and end user from start.	Easy to follow for even the most inexperienced user.	Nice to not need to go between pages, will save time.

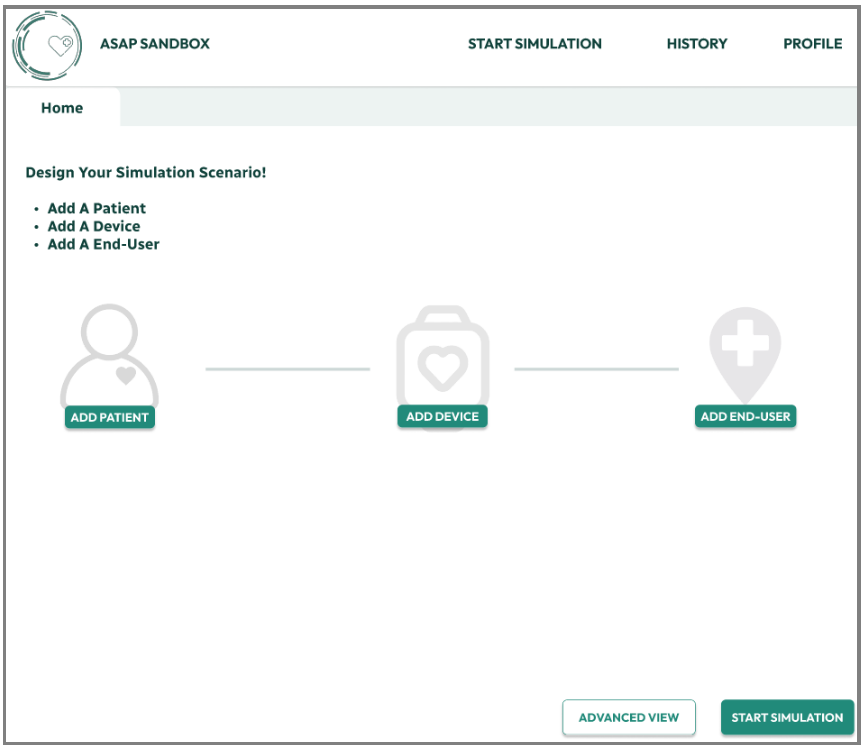
Table B.2: Evaluation of low-fidelity prototypes of the clinical interface.

B.2 A/B Testing Result

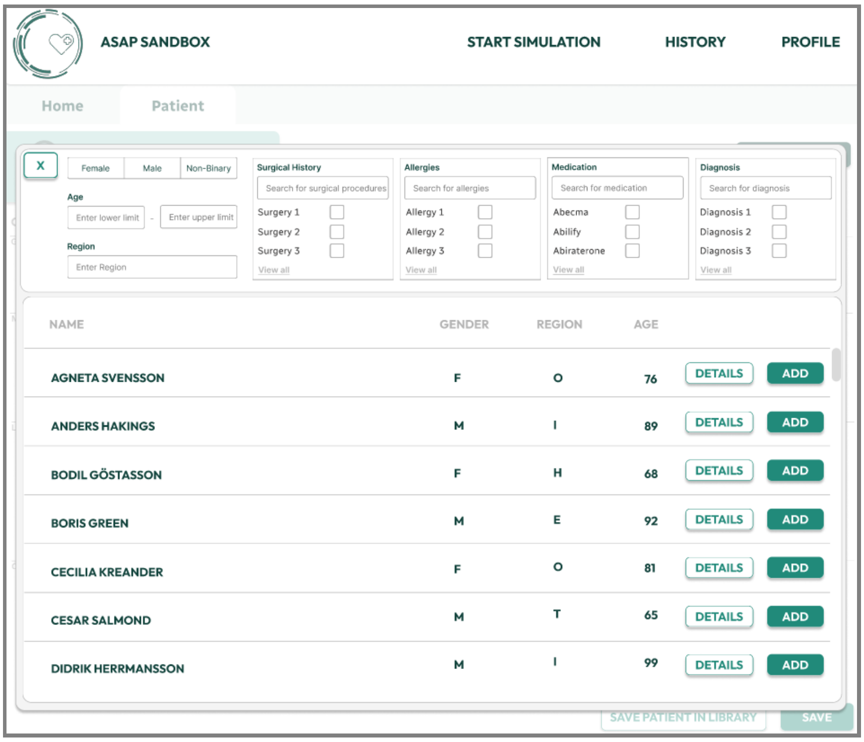
The A/B testing was conducted comparing three mockups: A, B and C. The key characteristics of the mockups are summarized in Table A.5 and visual representation of each mockup are shown in Figures B.5, B.6 and B.7.

Prototype	**A**	**B**	**C**
Page Layout	Single-page layout with tabs.	Five-page layout	Three-page layout
Layout for Simulation Setup	Visual workspace with auto-connected flow and example preview.	Simple pages with brief inputs per step.	Compact pages with detailed input fields.
Simulation Order	Flexible: components can be added in any order.	Sequential: components must be added in a specific order.	Sequential: components must be added in a specific order.
Patient Library Structure	Pop-up menu with filters at the top and patients listed below.	Separate page with filters on the left and patients displayed as cards on the right.	Pop-up view with filters on the left and a patient list below. The marked patient is visualized on the right.

Table B.3: Summary of key characteristics for the three wireframe prototypes: A, B and C of the clinical interface.

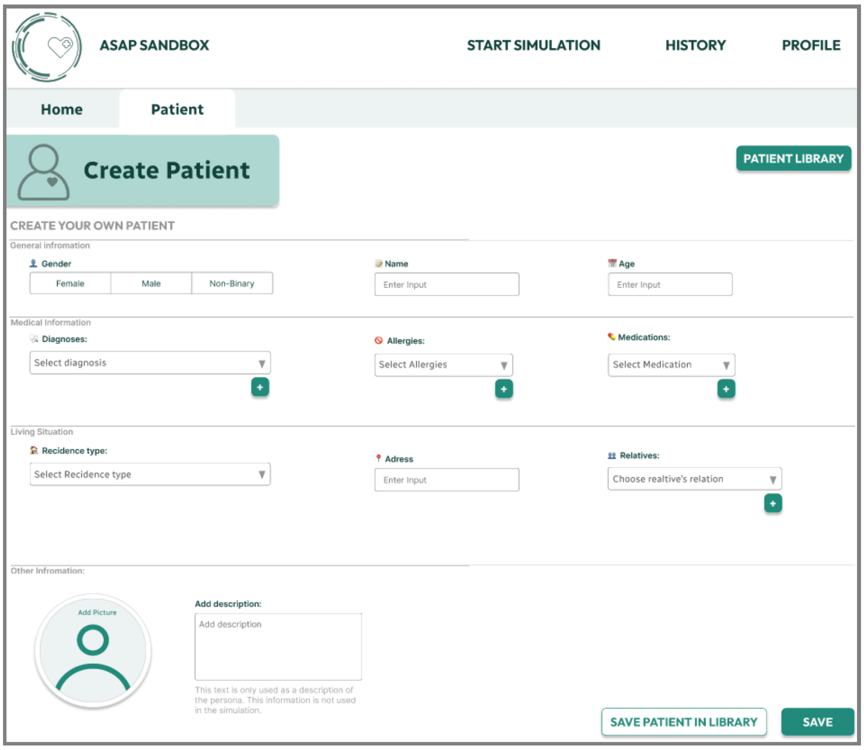


(a) Home page, showing the initial view for setting up a new simulation.

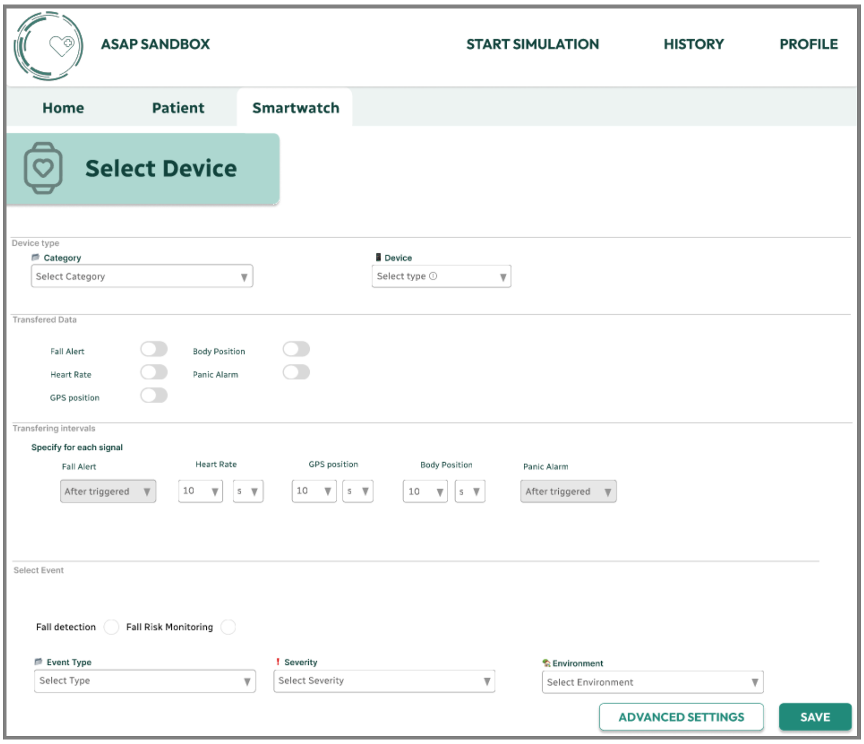


(b) Patient Library, showing how to select a patient from the library.

Figure B.5: Mockup A of the clinical interface used in A/B testing.

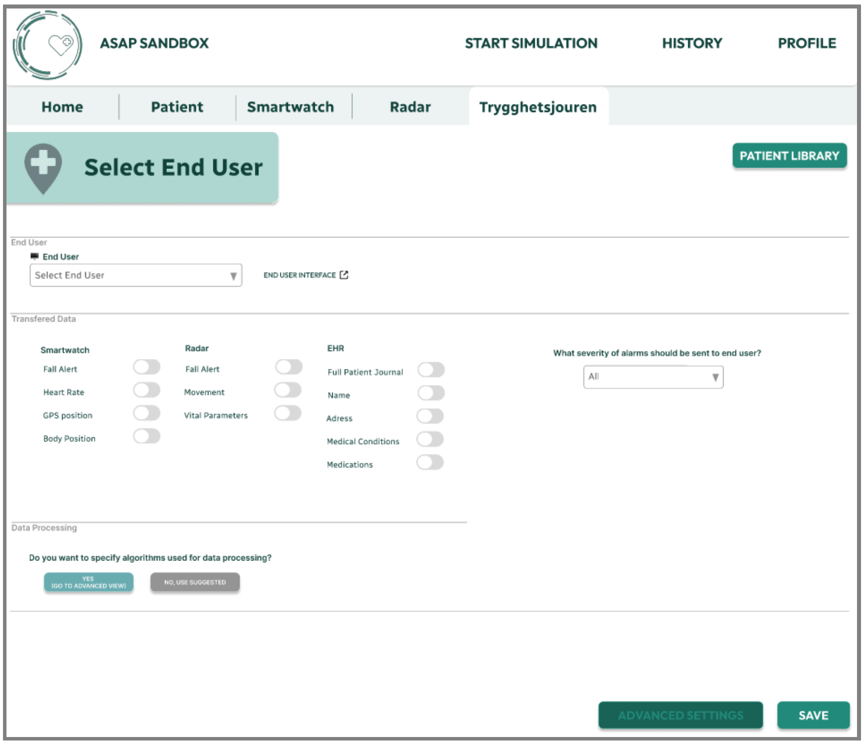


(c) Patient page, showing how to create a patient for the simulation.

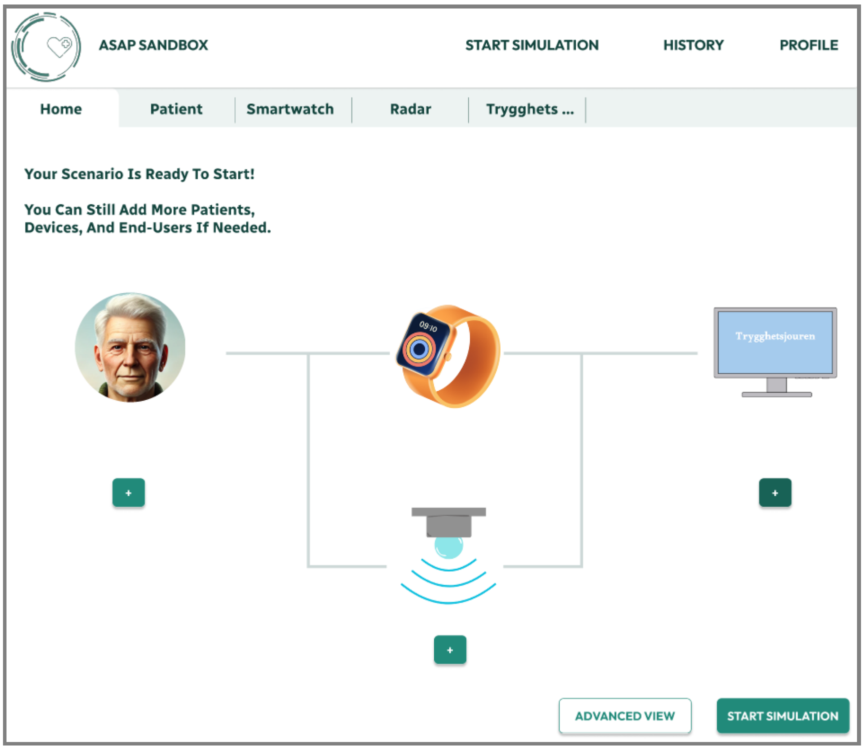


(d) Device page, showing how to select devices along with their corresponding data and settings.

Figure B.5: Mockup A of the clinical interface used in A/B testing.

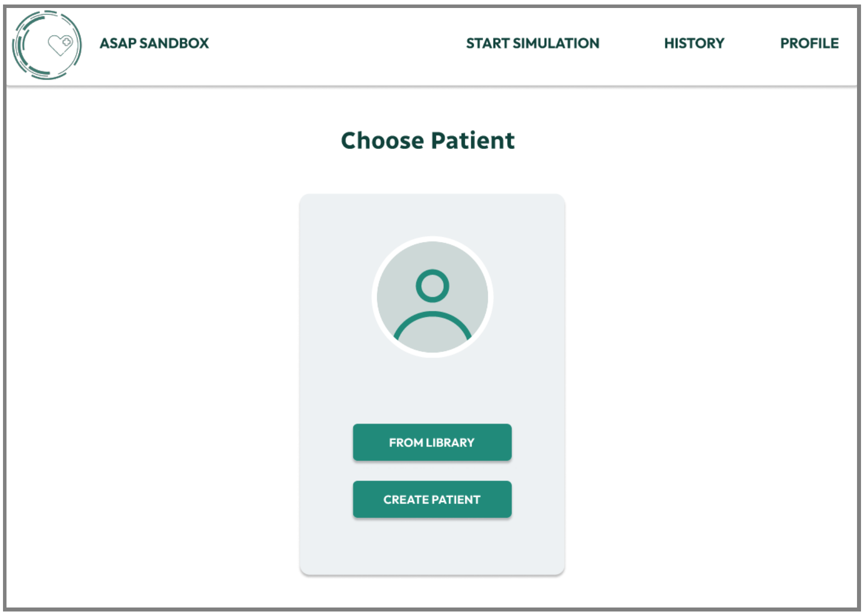


(e) End User page, showing how to select an end user and edit settings.

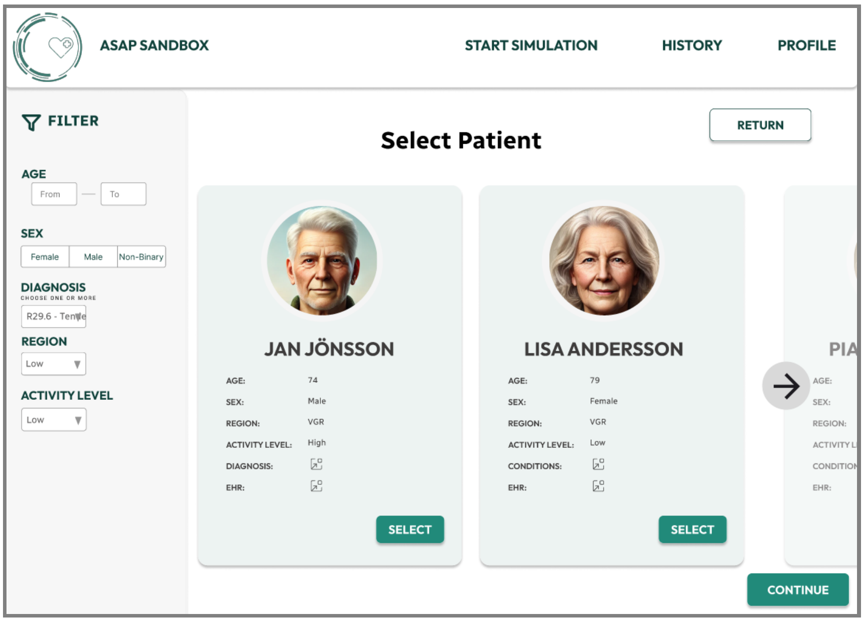


(f) Home page, displaying the completed simulation setup. (Patient images generated with ChatGPT using the prompt "elderly person, profile view", 2025)

Figure B.5: Mockup A of the clinical interface used in A/B testing.

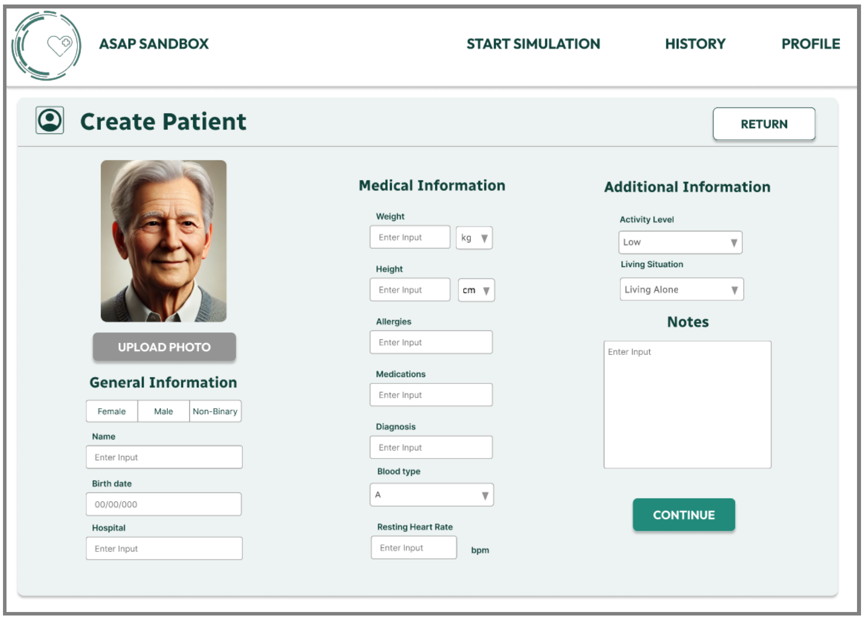


(a) Initial page, displaying the first step of the simulation: selecting or creating a patient.

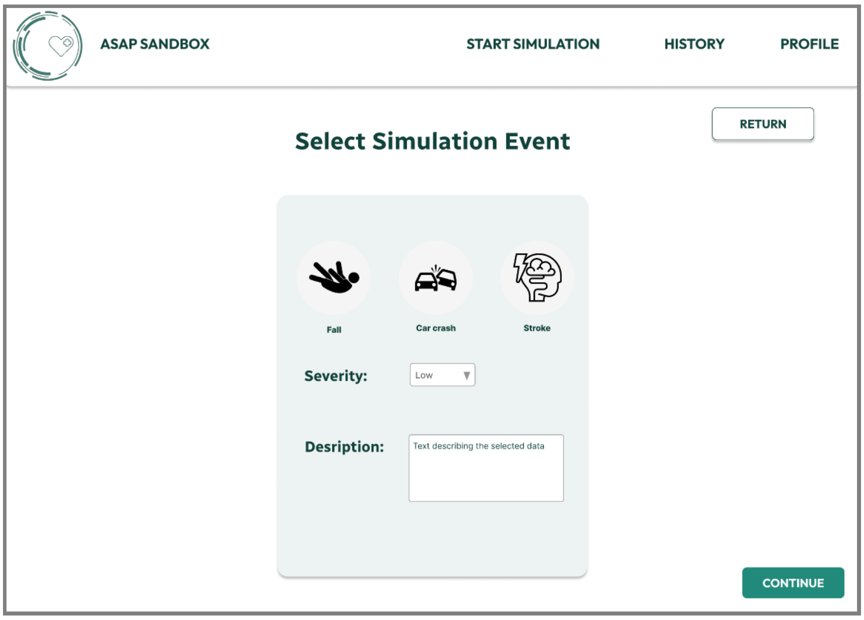


(b) Patient Library, displaying how to select patient from library. (Patient images generated with ChatGPT using the prompt "elderly person, profile view", 2025)

Figure B.6: Mockup B of the clinical interface used in A/B testing.

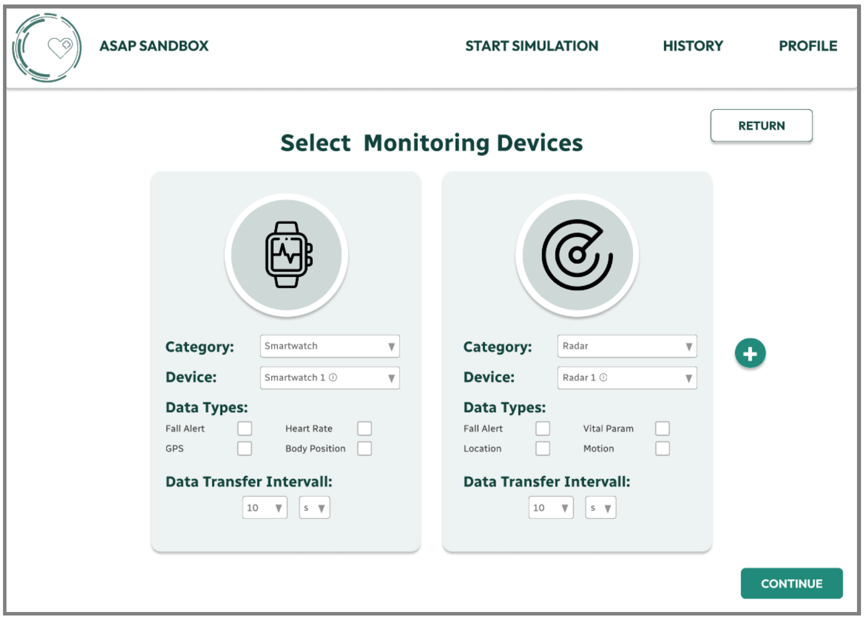


(c) Patient page, displaying how to create a patient for the simulation. (Patient images generated with ChatGPT using the prompt "elderly person, profile view", 2025)

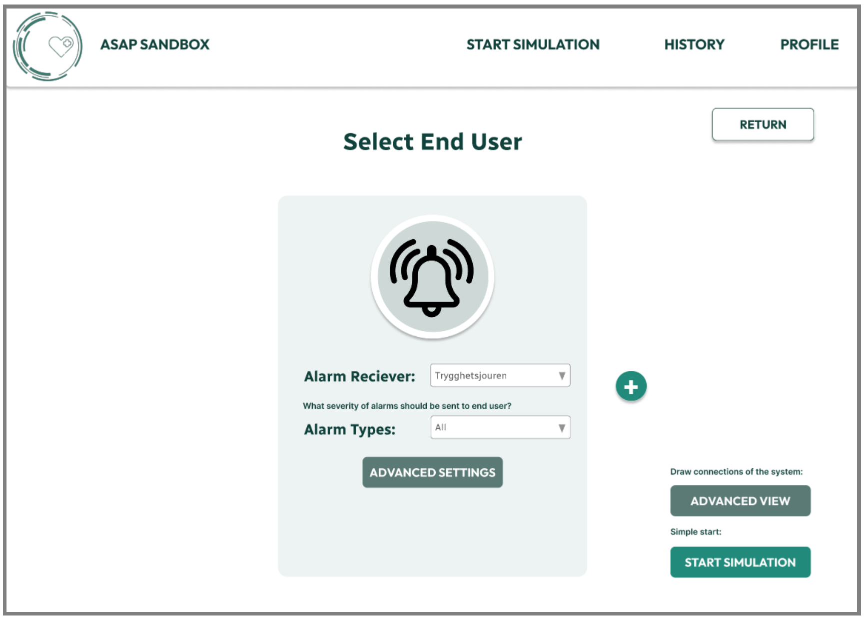


(d) Event page, displaying how to create an event for the simulation.

Figure B.6: Mockup B of the clinical interface used in A/B testing.

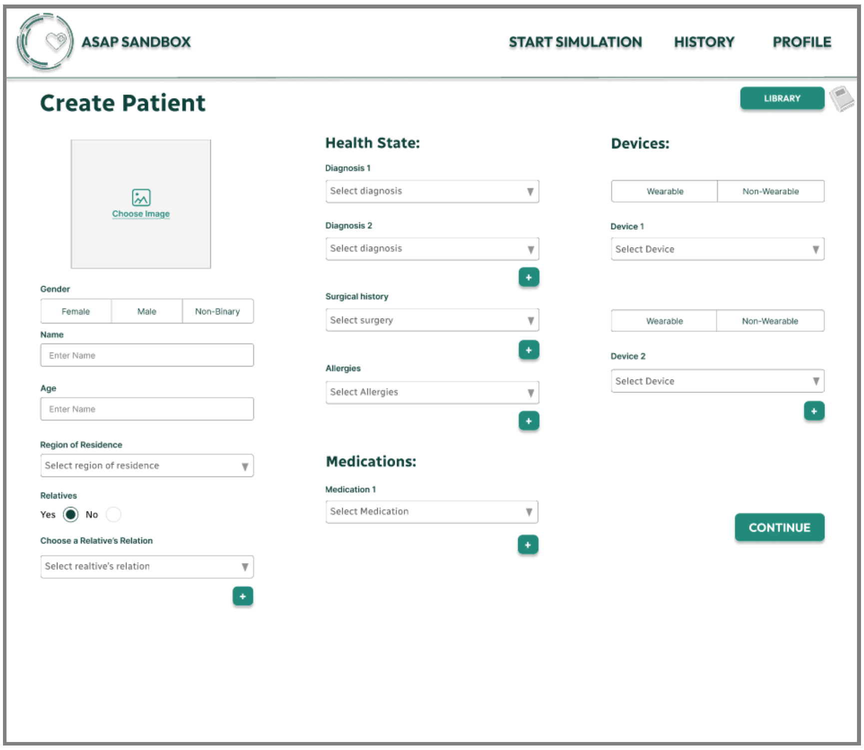


(e) Device page, displaying how to select and add devices to the simulation.

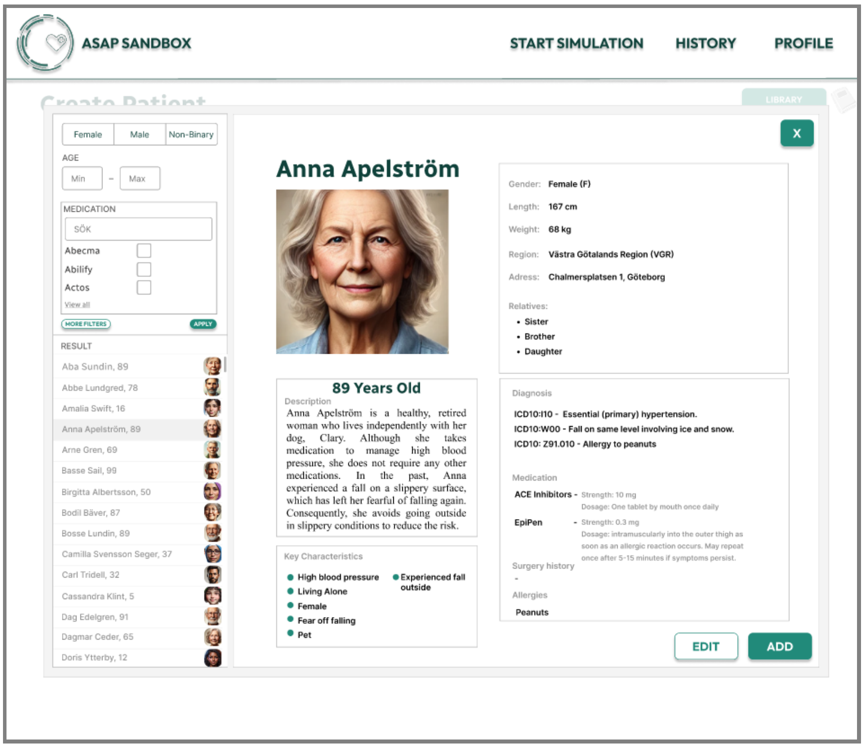


(f) End User page, displaying how to select and add end users, as well as configure advanced settings.

Figure B.6: Mockup B of the clinical interface used in A/B testing.

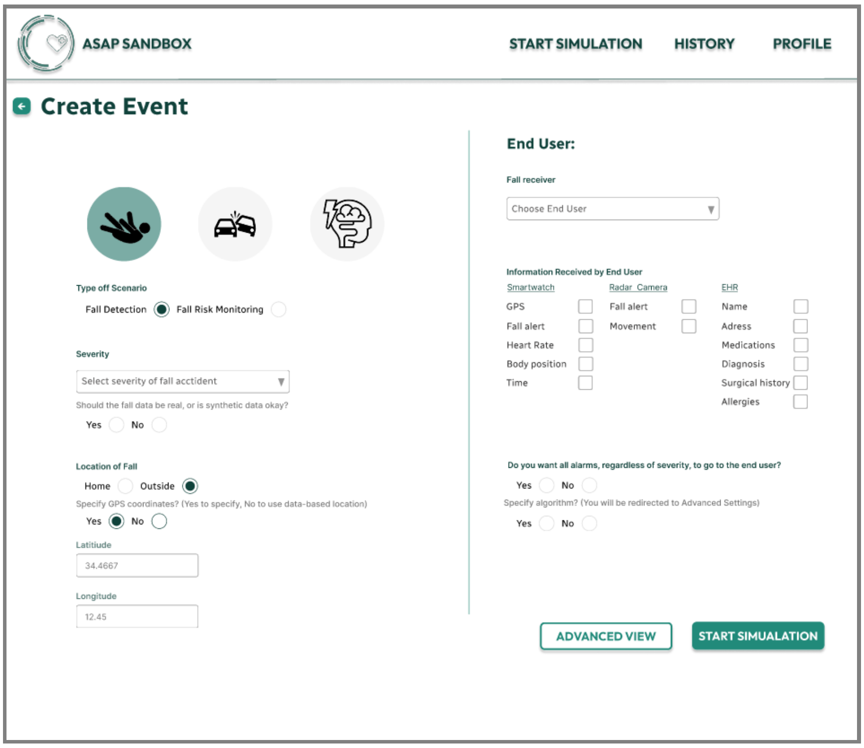


(a) Patient page, displaying how to create or select a patient, as well as select the patient's devices.



(b) Patient library, displaying how to select and filter patients from the library. (Patient images generated with ChatGPT using the prompt "elderly person, profile view", 2025)

Figure B.7: Mockup C of the clinical interface used in A/B testing.



(c) Event page, displaying how to select an event and an end user for the simulation, as well as how the end user receives event information.

Figure B.7: Mockup C of the clinical interface used in A/B testing.

The test was evaluated using a questionnaire in which participants ranked the ease of understanding different design concepts. A score of 1 represents 'Hard' and 5 represents 'Very Simple'. The results are presented in Table B.4.

	1	**2**	**3**	**4**	**5**	**Total Points**	**Mean Value**	**Median**
A								
Create a new patient	1	0	0	0	9	46	4,6	5
Select a patient from library	0	1	3	2	4	39	3,9	4
Select device	1	0	0	2	7	44	4,4	5
Add several devices	1	1	1	4	3	37	3,7	4
Create event for simulation	4	0	3	2	1	26	2,6	3
Select End user	1	0	0	1	8	45	4,5	5
Summation	8	2	7	11	32	237	3,95	4,5
B								
Create a new patient	0	0	0	2	8	48	4,8	5
Select a patient from library	1	0	0	0	9	46	4,6	5
Select device	0	0	2	2	6	44	4,4	5
Add several devices	0	0	0	1	9	49	4,9	5
Create event for simulation	0	1	0	0	9	47	4,7	5
Select End user	0	0	0	0	10	50	5	5
Summation	1	1	2	5	51	284	4,73	5
C								
Create a new patient	0	1	0	2	7	45	4,5	5
Select a patient from library	0	1	4	1	4	38	3,8	3,5
Select device	1	2	4	2	1	30	3	3
Add several devices	1	2	1	2	4	36	3,6	4
Create event for simulation	1	1	2	3	3	36	3,6	4
Select End user	1	1	2	3	3	36	3,6	4
Summation	4	8	13	13	22	221	3,68	4

Table B.4: Result from A/B test on mockup prototypes for clinical user.

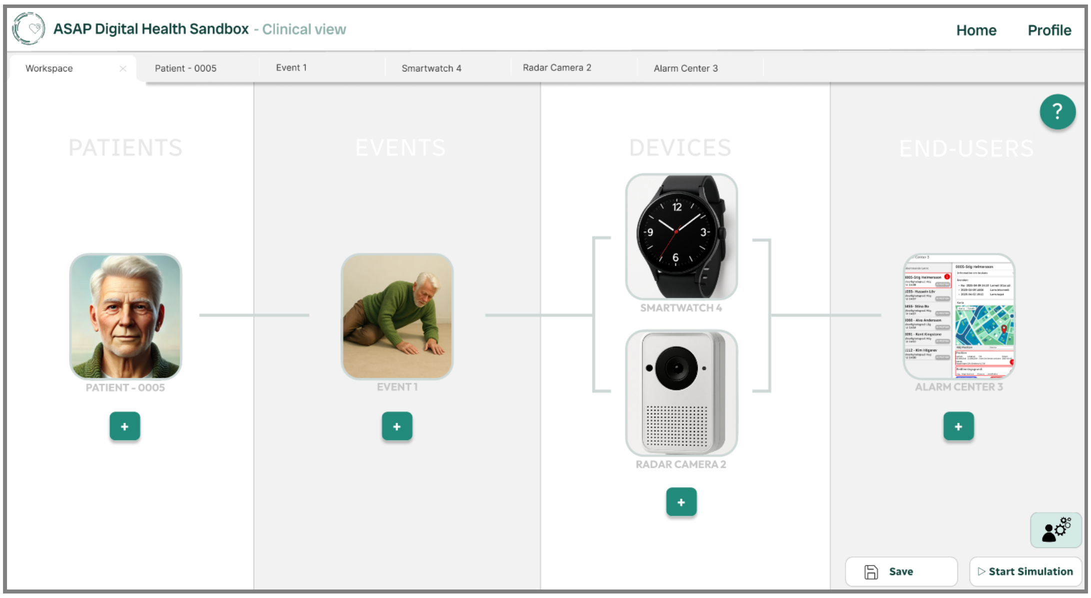
Some comments from test participants, offering important insights, are presented in Table B.5 below

From design	**Comment**
Alt. A	*I did not find how to select event.* *Very Easy, but i did miss the event.* *Did not understand how to add several devices.* *I did like that images did appear when you had picked components* *Follows a logical order.*
Alt. B	*I missed the Event.* *Very nice prototype. Easy to understand when you did it in the order it has been designed for.* *Not as clear summary as in alternative A.* *For me it felt like B could complement A.*
Alt. C	*Liked this model too. Easier to understand than A, but the library was a bit hard.* *A bit unclear how to add several devices.* *Great, but felt more messy then B.* *Great layout!* *Easy to miss things when several things is on the same page.*

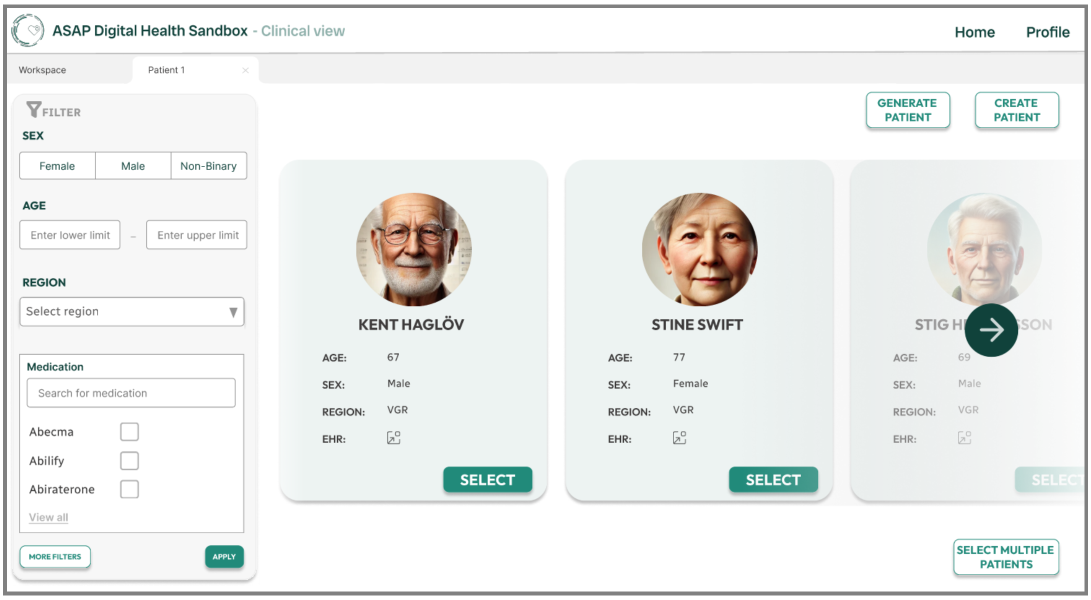
Table B.5: Feedback from A/B test participants on Clinical mockups.

B.3 Task Oriented Test Result

The task-oriented test was conducted with an interactive prototype, with the main pages presented in Figure B.8

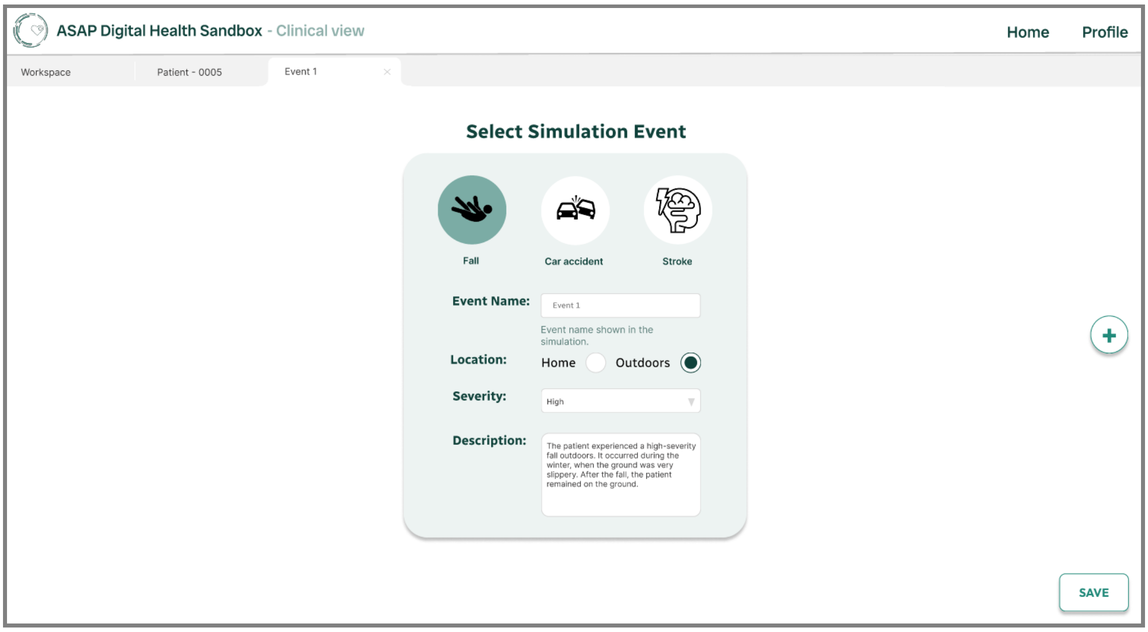


(a) Home page, displaying the final simulation setup. (Patient and device images generated with ChatGPT using the prompt “elderly person, profile view”, “elderly person falling, neutral background” "smartwatch, neutral background" and "radar camera, neutral background", 2025)

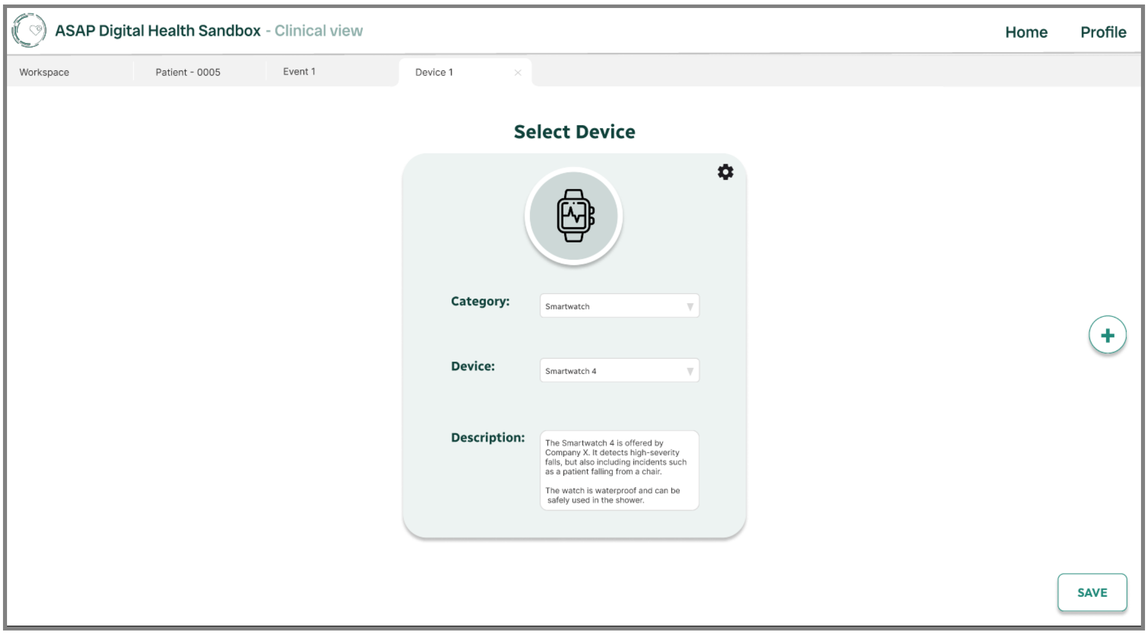


(b) Patient page, displaying how to select patient from library or create patient. (Patient images generated with ChatGPT using the prompt “elderly person, profile view”, 2025)

Figure B.8: Interactivity prototype of clinical view used for task oriented test.

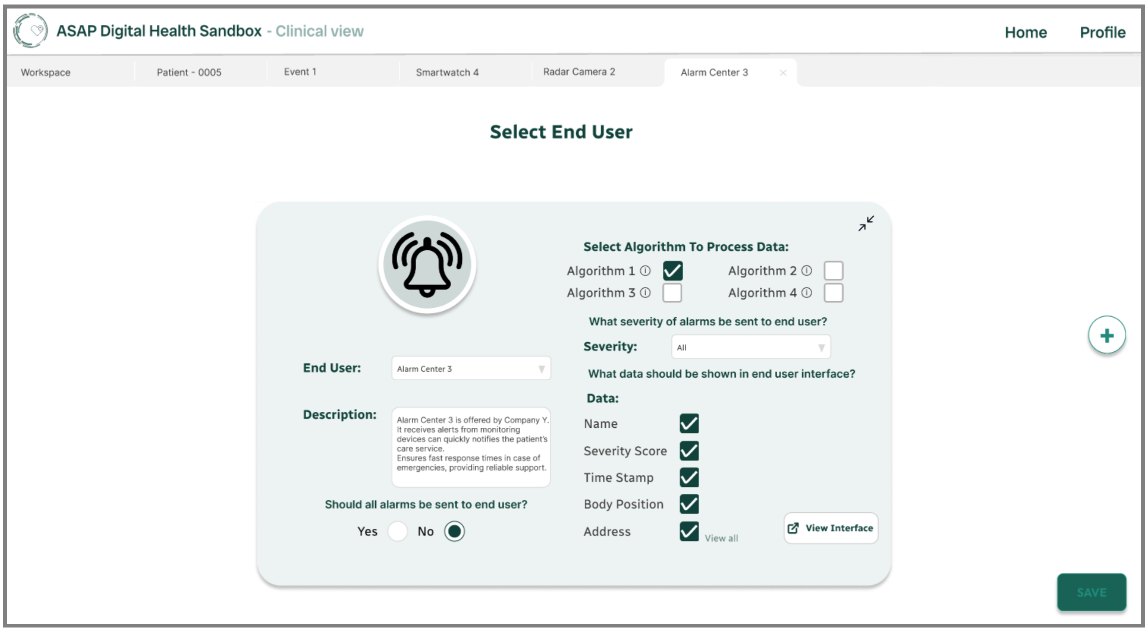


(c) Event Page, displaying how to create an event for the simulation.



(d) Device page, showing how to select a device and view its description, with access to advanced settings such as modifying the transferred data.

Figure B.8: Interactivity prototype of clinical view used for task oriented test.



(e) End User page, displaying how to select an end user and, if desired, specify where to integrate the data into the end user's interface.

Figure B.8: Interactivity prototype of clinical view used for task oriented test.

The result from the task oriented test is presented in Table B.6.

Task Time	Misclick	Comments
4 min 37 sec	0	*It was hard to tell that the checkbox belonged to that specific section (in the end user view). Otherwise Great!*
8 min 8 sec	2	*I like the flow showing images, it makes it easier to understand.*
4 min 2 sec	1	*Easy to set up the simulation but a bit hard to find the technical view.*
4 min 10 sec	0	*Liked that the images came up once a part was added. Very clear.*
4 min 0 sec	1	*Hard to find the technical view but otherwise very simple.*
4 min 12 sec	0	*Liked the images and the preview of the technical view. A bit confusing that some active text was gray.*
4 min 17 sec	1	*Nice and easy to set up the simulation. A bit hard to understand how to switch between views.*
4 min 38 sec	1	*Great! I like that you include images of the patient. Using a personal ID number or something similar creates a sense of distance from the patient.*
3 min 29 sec	0	*Intuitive way to set up the simulation. Aesthetic GUI.*
3 min 0 sec	0	*Easy to follow. Hard to make mistakes. Clear layout. Could be some more colors for highlighting things, for guidance.*

Table B.6: Test result from task oriented test on interactivity prototype of clinical view.

B.4 Test Scripts

Below, the test scripts used for the A/B test and task oriented test are visualized.

A/B test of mockups: Clinical User

This design project aims to develop a web-based interface for a digital health sandbox. A Sandbox is a test environment where users can simulate real-world settings. In this case, a simulation of a fall detection setup will be created. The case simulated is shown in figure 1.

The image below shows the simulation scenario: A patient falls, and the event is detected by the patient's smartwatch or radar camera. Signals from the patient's devices are processed, and the alarm is then forwarded to the emergency response center.

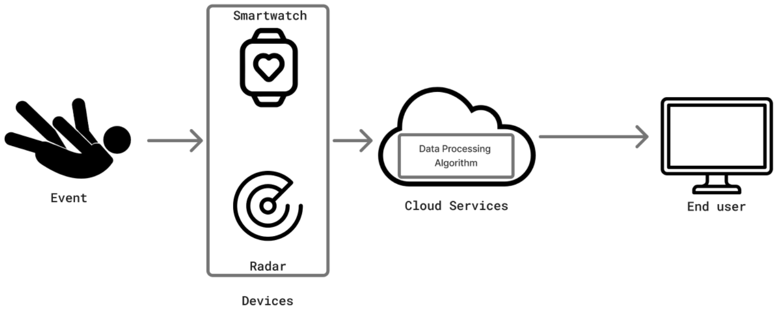


Figure 1: Simulation case used in the GUI

In this test, three different GUI concepts will be presented, illustrating how users can interact with the system to build a simulation based on the scenario shown in the image above. Your task is to evaluate them based on questions in the given form.

Your task is solely to evaluate the interfaces and assess how intuitive and easy they are to understand based on the questions in the form.

Approximate time: 15 min

Thank you for your time!

Test of Interactivity prototype: Clinical view

This project aims to design a website for simulating various fall detection setups. The image below shows a possible scenario: a patient falls, and the incident is detected by either the patients smartwatch or radar camera. The signal is then processed and transmitted to the alarm center.

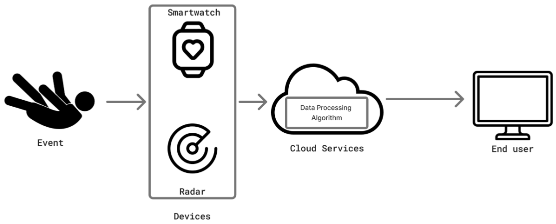


Figure 1: Simulation case used in the GUIs

You will be asked to perform tasks with the goal of building the simulation setup shown in Figure 2.

Approximate time: 5 min

Task Set 1

1. Add a patient.
2. *Are you able to find how to create a new patient?*
3. From the patient library, select the patient named **Stig Helmersson**.
4. Add a **Fall Event**, **Outdoors** and with **High Severity**.
5. Return to the Workspace
6. Add a smartwatch **device** named **Smartwatch 4**.
7. Open the advanced settings for **Smartwatch 4** and change the **Transfer Interval** for **Body Position** to 5s.
8. Add a radar camera device named **Radar camera 2**.
9. Return to the workspace.
10. Add an end user called **Alarm Center 3**.
11. Configure the end user so that **not** all alarms are sent to them.
12. Select **Algoritm 1**.
13. Go to Interface view
    - For **Section 1**, choose the data:
        - **Name**
        - **Severity Score**
        - **Time Stamp**
14. Return to the workspace.
15. *Do you understand how to change back and forth between the technical and clinical view?*
16. Save and Start the simulation

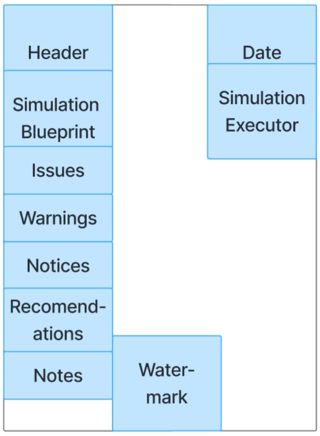
Thank you for your time!

C

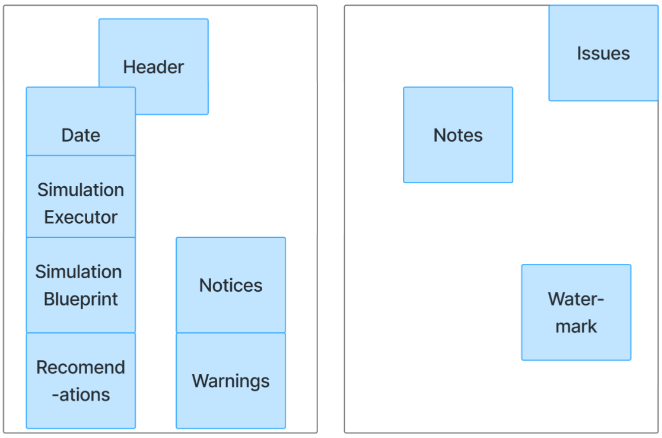
Appendix: Simulation Result Protocol

C.1 Card Sorting Test Result

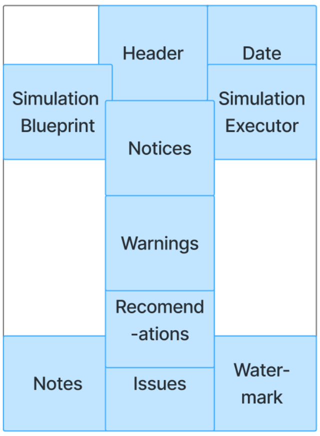
In Figure C.1 computer made copies of the card sorting test results is visualized.



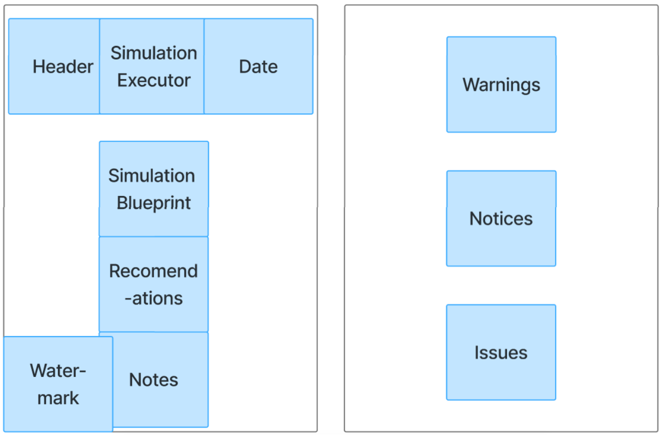
(a) Participant 1



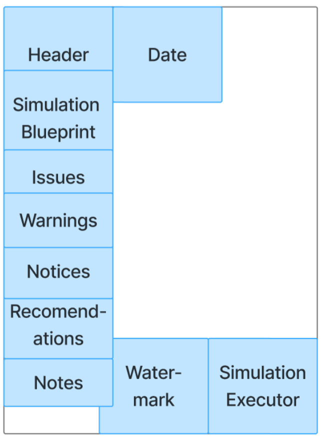
(b) Participant 2



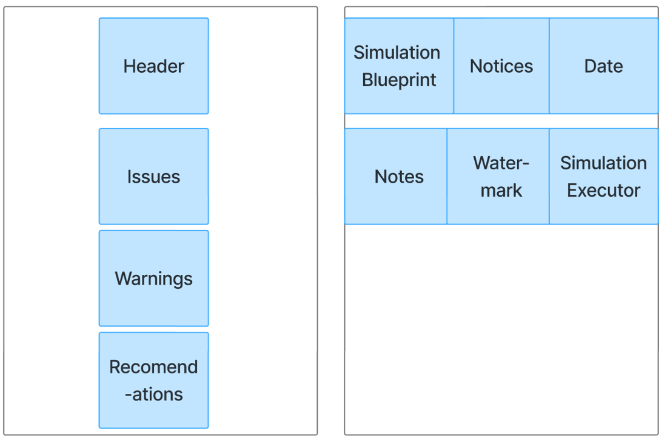
(c) Participant 3



(d) Participant 4

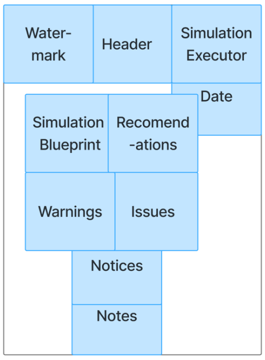


(e) Participant 5

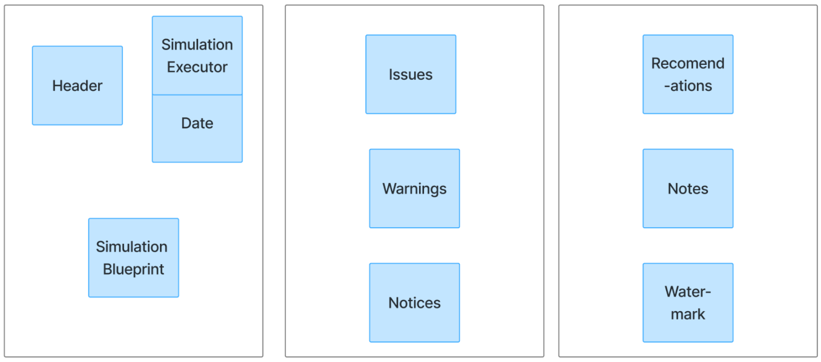


(f) Participant 6

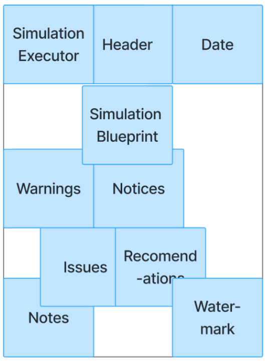
Figure C.1: Card sorting test result from 10 participants.



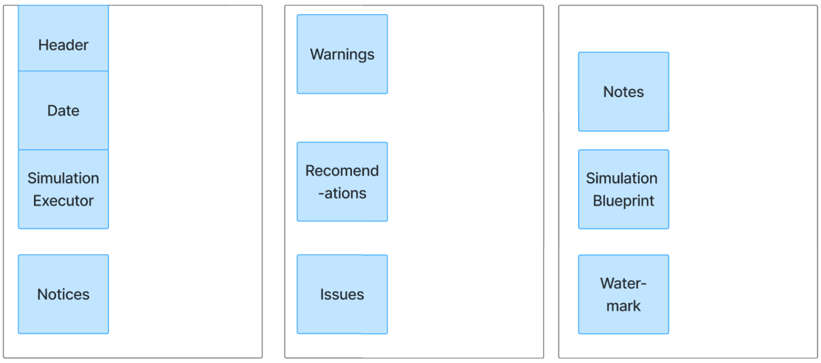
(g) Participant 7



(h) Participant 8



(i) Participant 9

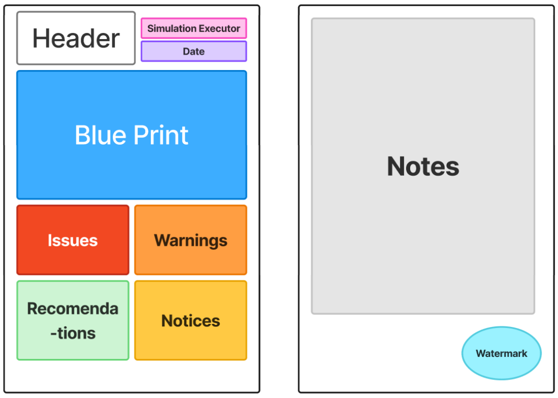


(j) Test Result 10

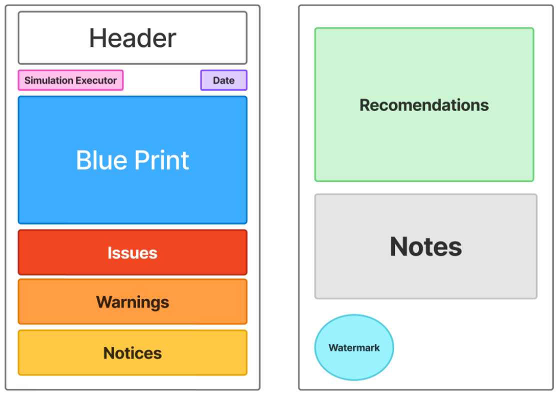
Figure C.1: Card sorting test result from 10 participants.

C.2 A/B Test Result

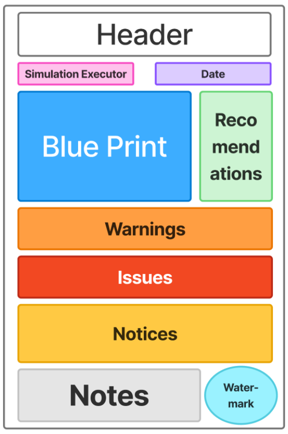
The four sketches of simulation result protocol evaluated in the A/B test is visualized in Figure C.2.



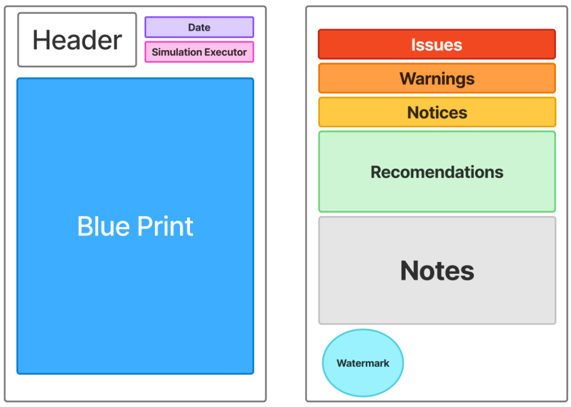
(a) Alternative 1



(b) Alternative 2



(c) Alternative 3



(d) Alternative 4

Figure C.2: Wireframe sketches showing different alternatives for Simulation result protocol layout.

In the A/B test, participants ranked the alternatives from their most to least preferred option. To determine the most preferred alternative, a scoring system was applied: 0 points for the least liked, 1 point for the second least, 2 points for the second best, and 3 points for the most liked option. The points were then summed up, resulting in Alternative 2 being the most preferred choice, which is presented in table C.1.

	Least (0)	Second Least (1)	Second Best (2)	Best (3)	Total Points
Alt. 1	3	1	2	4	17
Alt. 2	0	4	2	4	20
Alt. 3	2	4	2	2	14
Alt. 4	5	1	4	0	9

Table C.1: Summation fo test result for simulation result protocol sketches.

Some relevant comments from the participants that was used for motivating design choices is presented in Table C.2

Preferred design	**Comment**
Alt. 2	*I like the concept of a summery page but not the layout chosen in the example. I prefer the information going from top to bottom only.*
Alt. 1	*Maybe the watermark should be on all pages.*
Alt. 3	*Should be an watermark on all pages for alternative 1,2 and 4. Maybe also have different design on the protocol for different kind of receivers.*
Alt. 3	*Great to have everything on one page, makes it easier to follow. I would like a clean and open layout.*

Table C.2: Feedback from A/B test participants on the Simulation Result Protocol sketches.

C.3 Test Scripts

The script given to the participants for the card sorting test and A/B test is shown below:

Card sorting test: Simulation Result Protocol

A simulation result protocol is meant to show the results of a completed simulation. It should present what worked well and highlight any problems that occurred, as well as details about the performed simulation. This protocol will be in PDF format on A4 pages.

You will receive 10 post-it notes, each representing a section that should be included in the protocol. Your task is to arrange them on one or more A4 pages to illustrate how you would structure a simulation protocol. Place the post-it notes where you think the information should appear.

The provided sections are:

- **Header** - Specifies the type of document, including the title and other relevant details.
- **Date** - the date the simulation was performed
- **Simulation blueprint** - describing the simulation so it can be replicated
- **Watermark** - Indicates that the simulation was performed by the ASAP sandbox and can be used as a reliable source.
- **Simulation executor** - Name or ID of the person setting up and performing the test
- **Notices** - Important information, but still works.
- **Warnings** - Problematic areas with the set up.
- **Issues** - Parts that will not work in the set up.
- **Notes** - Space for the simulation executor to leave notes/thoughts about the simulations.
- **Recommendations** - Recommendations for the set up

Approximate time: 5 minutes.

A/B test of Simulation result protocol

A simulation result protocol is meant to show the results of a completed simulation. It should present what worked well and highlight any problems that occurred, as well as details about the performed simulation. This protocol will be in PDF format on A4 pages.

In this test you will review four different versions of the protocol, all containing the same sections but in different arrangement and answer the corresponding questions in a form about which version you preferred. We want you to evaluate the protocols on what version you considered most intuitive.

The sketches do not specify the number of pages the document should have, but rather define how the components should be arranged in relation to each other and in what order they should appear. (Except for alternative 3, which will summarize everything on the first page and provide a more detailed description of technical details on additional pages.)

The provided sections are:

- **Header** - Specifies the type of document, including the title and other relevant details.
- **Date** - The date the simulation was performed
- **Simulation blueprint** - Describing the simulation so it can be replicated
- **Watermark** - Indicates that the simulation was performed by the ASAP sandbox and can be used as a reliable source.
- **Simulation executor** - Name or ID of the person setting up and performing the test
- **Notices** - Important information, but still works. (e.g. GPS position not possible when user is in basement)
- **Warnings** - Problematic areas with the set up. (e.g. Data transfer with this high interval will cause device running out of battery)
- **Issues** - Parts that will not work in the set up. (e.g. Data formats not compatible between component A and B)
- **Notes** - Space for the simulation executor to leave notes/thoughts about the simulations.
- **Recommendations** - Recommendations for the setup (e.g. Add data pre processing for component B to match format of B)

When you have looked at all the sketches, you will fill out a form, ranking the alternatives according to your preferences.

Approximate time: 5 minutes.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY