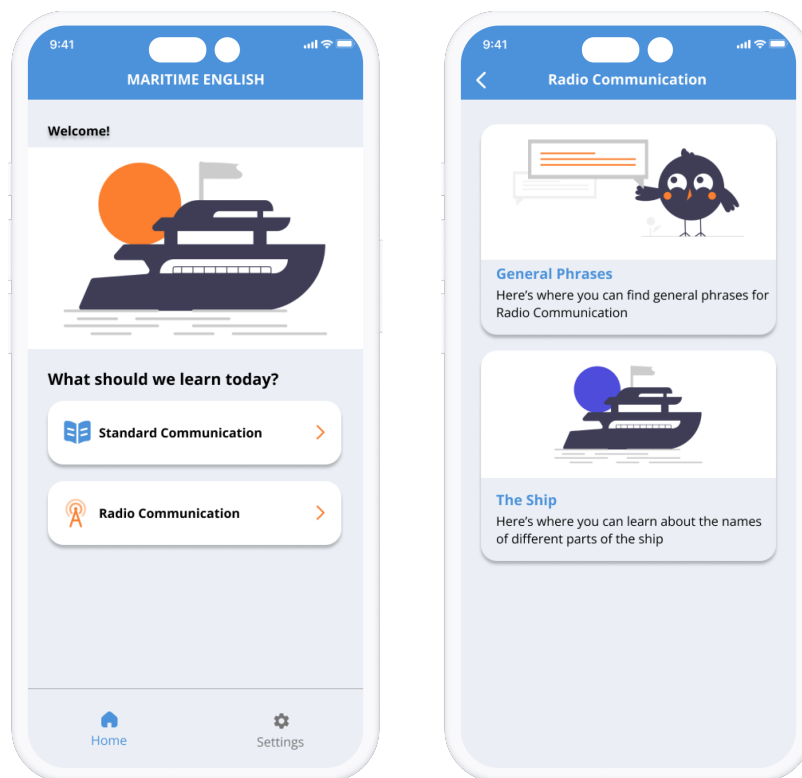




CHALMERS



GÖTEBORGS UNIVERSITET



# Utveckling av en språkinlärningsapplikation inom maritim engelska

Genom en användar-centrerad design process

Examensarbete inom Data- och Informationsteknik

ELIAS CARLSSON, HANNA PHAM

**INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK**

CHALMERS TEKNISKA HÖGSKOLA, GÖTEBORGS UNIVERSITET  
Göteborg 2023  
www.chalmers.se

EXAMENSARBETE 2023

# Utveckling av en språkinlärningsapplikation inom maritim engelska

Genom en användar-centrerad design process

ELIAS CARLSSON, HANNA PHAM



GÖTEBORGS  
UNIVERSITET

---



**CHALMERS**

Institutionen för Data- och Informationsteknik  
CHALMERS TEKNISKA HÖGSKOLA  
GÖTEBORGS UNIVERSITET  
Göteborg 2023

Utveckling av en språkinlärningsapplikation inom maritim engelska  
Genom en användar-centrerad design process

ELIAS CARLSSON  
HANNA PHAM

© ELIAS CARLSSON, HANNA PHAM 2023.

Handledare: Pelle Evensen, Chalmers  
Examinator: Jonas Almström Duregård, Chalmers

Examensarbete 2023  
Institutionen för Data- och Informationsteknik  
Chalmers Tekniska Högskola  
Göteborgs Universitet  
SE-412 96 Göteborg  
Telefon +46 31 772 1000

Omslagsbild: Skärmdump på sidorna “Home” och “Radio Communication” i applikationen.

Skriven i L<sup>A</sup>T<sub>E</sub>X  
Göteborg 2023

Utveckling av en språkinlärningsapplikation inom maritim engelska  
Genom en användar-centrerad design process  
ELIAS CARLSSON  
HANNA PHAM  
Institutionen för Data- och Informationsteknik  
Chalmers Tekniska Högskola  
Göteborgs Universitet

## **Abstract**

This report presents the methodology, the work done and the results achieved during the development of Maritime English application. The purpose of this project is to develop an application that can be downloaded on a mobile phone. The goal of the application is to provide various exercises that can help students to learn Maritime English, including defining, pronouncing and listening to Standard Marine Communication Phrases (SMCP) as well as identify different ship parts.

The application is developed on the Flutter framework along with open-source design material to ensure that future Chalmers students have the opportunity to continue the development of the application. The main focus of the application is implementing exercises for learning ship vocabulary.

The results in an application that allows students to learn ship vocabulary through various exercises. The application also provides definitions of SMCP phrases, pronunciation practice and listening to SMCP. The application follows the project owner's requirements and there is potential to expand it with more exercise for SMCP phrases.

# Akronymer

Nedan är listan över beteckningar som har använts i rapporten. Listan är sorterad i alfabetisk ordning:

MVC	Model-View-Controller; på svenska: Modell-Vy-Kontroller
IMO	International Maritime Organization; på svenska: Internationella sjöfartsorganisationen
OEM	Original Equipment Manufacturer; på svenska: Ursprunglig utrustningstillverkare
SMCP	Standard Marine Communication Phrases; på svenska: Standardiserade marina kommunikationsfraser
SUS	System Usability Scale; på svenska: Systemets användbarhetsskala

# Innehåll

<b>Akronymer</b>	<b>iv</b>
<b>1 Inledning</b>	<b>2</b>
1.1 Bakgrund . . . . .	2
1.2 Syfte . . . . .	2
1.3 Mål . . . . .	3
1.4 Avgränsningar . . . . .	3
<b>2 Metod</b>	<b>4</b>
2.1 Verktyg . . . . .	4
2.2 Utvecklingsmodeller . . . . .	5
<b>3 Teknisk Bakgrund</b>	<b>8</b>
3.1 Flutter . . . . .	8
3.2 MVC . . . . .	8
3.3 Dart, Dartpaket . . . . .	9
<b>4 Genomförande</b>	<b>10</b>
4.1 Kravinsamling . . . . .	10
4.2 Prototyp . . . . .	10
4.3 Fartygsskisser . . . . .	15
4.4 Applikationsflöde . . . . .	16
4.5 Systemkonstruktion . . . . .	18
4.6 Använda Dart-paket . . . . .	20
4.7 Användarutvärdering . . . . .	20
<b>5 Resultat</b>	<b>23</b>
<b>6 Diskussion</b>	<b>25</b>
<b>Bibliography</b>	<b>27</b>
<b>A Appendix A</b>	<b>II</b>
<b>B Appendix B</b>	<b>V</b>

# 1

## Inledning

Syftet med detta avsnitt är att ge en översikt av projektets bakgrund, syfte och avgränsningar. Dessutom presenteras mål för projektet.

### 1.1 Bakgrund

Den maritima industrin är en global bransch som är beroende av internationellt samarbete för fortsatt utveckling. Säkerheten, både på fartyg och land, är beroende av god kommunikation mellan parterna. På grund av detta, har International Maritime Organisation (IMO), ett FN-organ som grundades 1948 för att reglera sjöfartsindustrin globalt, infört ett antal föreskrifter och standarder för att säkerställa kommunikationsförmågan hos sjöfarare. En av dessa föreskrifter är att alla officerare inom ledande och operativa positioner ska vid behov kunna konversera endast genom Standard Marine Communication Phrases (SMCP) [1].

SMCP är en standardiserad uppsättning fraser och frågor på engelska som används för att kommunicera vanliga meddelanden och nödsituationer inom maritima industri. Syftet med att använda SMCP är att säkerställa att oavsett bakgrund, kan all manskap ombord kommunicera tydligt och effektivt, både med interna och externa aktörer [2]. Därför ingår undervisning av SMCP i alla högskoleutbildningar på Chalmers inom frakt och marin teknologi.

Produkten är beställd av undervisare på Chalmers.

### 1.2 Syfte

Projektets syfte är skapa en mobil applikation som underlättar lärandet av Maritime English för Chalmersstudenter. För närvarande lär sig studenter Maritime English och fartygsdelar genom ett äldre program, Lingus, som endast är installerat på Chalmers datorer. Att programmet inte är tillgängligt för nedladdning till egna enheter och att det är omodernt gör att ett nytt program efterfrågas. Med detta arbete kommer en förbättrad applikation skapas för framtida användning på Chalmers utbildningar. Applikationen ska kunna vidareutvecklas av framtida Chalmersstudenter så att den kan användas för alla delar av Maritime English utbildningen.

## 1.3 Mål

Målet med projektet är att skapa en applikation som hjälper studenter att förbättra sina kunskaper inom Maritime English och uppfyller beställarens krav och specifikationer.

Applikationen kommer att innehålla olika typer av övningar, såsom definition av SMCP-fraser, uttal av SMCP-fraser, hörförståelse för SMCP-fraser och identifiering av olika delar på fartyg. En konkret specifikation är att övningen av olika delar på fartyg ska täcka alla delar som finns i tre skisser som kan ses i bilaga A.

Applikationen kommer att vara tillgänglig för operativsystemen Android och iOS.

## 1.4 Avgränsningar

Projektet kommer inte att ta ställning till de pedagogiska aspekterna av språkinläring. Projektet ska förhålla sig till de föreskrifter och standarder som IMO infört genom att inte avvika från SMCP-fraserna och ge korrekt beskrivning till dessa. Projektet ska också förhålla sig till Sveriges Civilingenjörsförbunds hederskodex. Inga ekologiska aspekter kommer att beröras i projektet eftersom projektet inte kommer att ha någon påverkan på miljön. Inga samhällliga aspekter kommer heller att beröras eftersom projektet kommer ha obetydlig påverkan på samhället.

En annan avgränsning är att applikationen endast kommer att vara tillgänglig på engelska och SMCP-fraserna kommer bara att visas på engelska på grund av beställarens nuvarande specifikationer.

# 2

## Metod

Detta avsnitt beskriver de verktyg och utvecklingsmodeller som har använts under projektet. De verktyg som presenteras är Flutter, Git, GitHub Project Board, Figma, LottiesFile, unDraw och WebAIM Contrast Checker. Sedan presenteras utvecklingsmodeller Scrum, användar-centrerad design och användbarhetstest.

### 2.1 Verktyg

#### **Flutter**

Applikationen utvecklas i Flutter ramverket eftersom detta ger möjligheter att utveckla en applikation som fungerar på flera plattformar utan stora ändringar i källkod. Användning av Flutter bestämdes gemensamt med beställaren eftersom ett annat liknande projekt genomförs med ramverket. Detta ger möjligheter för beställaren att vidareutveckla båda dessa applikationer gemensamt. Flutter använder sig av programspråket Dart.

#### **Git**

För versionshantering används Git med lagring på GitHub. GitHub ger möjligheter för flera personer att samtidigt skriva kod till projektet genom att lagra källkoden och använda Git för att uppdatera och hämta koden från egen maskin.

#### **GitHub Project Board**

GitHub project board används för att planera och organisera uppgifter. Det är ett visuellt arbetshanteringsverktyg, där alla gruppmedlemmar får tillgång till en Project-board för att kunna signa sig på vilka uppgifter de kommer utföra och samtidigt få en överblick av arbetsframstegen.

#### **Figma**

Figma används för att framställa en prototyp av applikationens användargränssnitt. Figma är ett onlineverktyg där en prototyp av hög kvalitet och trovärdighet kan framställas. Den fungerar som en tidigt version av applikationen, vilket ger gruppen möjlighet att testa funktionaliteten och göra justeringar innan implementeringen. Dessutom kan gruppen använda Figma-filen för att hämta designinformation, såsom färgschema, typografi och dimensioner för att skapa applikationen som den har designats.

#### **LottieFiles, unDraw**

För att göra applikationen mer attraktiv har LottieFiles och unDraw använts. LottieFiles är ett bibliotek för gratis animationsfiler i Lottie-formatet. Lottie är ett format för lättvikts-animationer med syfte för användning inom webbsidor eller applikationer [3]. LottieFiles används för att skapa rörlig grafik och animationer i applikationen. unDraw är en samling illustrationer med öppen källkod som är utformade för användning inom webbsidor och applikationer. I unDraw finns det funktionalitet för att ändra färger på illustrationer, vilket gör att det är enkelt att anpassa dem efter det specifika färgschema som används i användargränssnittet [4].

### **WebAIM Contrast Checker**

WebAIM Contrast Checker är ett webbaserat verktyg som kontrollerar kontrasten mellan två färger och ser om de uppfyller tillgänglighetskrav av Web Content Accessibility Guidelines [5]. Detta hjälper till att säkerställa att text och dess bakgrundsfärg som valts har tillräcklig kontrast för att vara läsbara för användare, inklusive användare med synnedsättning.

## **2.2 Utvecklingsmodeller**

### **Scrum**

För projektledning använder gruppen den agila metoden Scrum som anges av [6]. Syftet med Scrum är att kunna reagera snabbt på de förändringar som sker under projektets gång. Dessa förändringar kan till exempel vara nya krav och specifikationer, tidsram för projektet eller ny teknologi som ska användas. Scrum föreslår ett arbetssätt i tre faser. En före-fas, en utvecklingsfas och en efter-fas. Detta projekt kommer endast följa utvecklingsfasen av Scrum då de andra faserna utförs med annan metodik. Scrum-arbetssättet betyder att utvecklingen sker i sprints, som är iterativa cykler som alla innehåller kravhantering, analys, design, utveckling och leverans. I början av varje cykel kommer sprinten att planeras utifrån prioriterade krav. Under sprintens gång kommer nya krav att analyseras utifrån tid och nytta, för att sedan prioriteras upp eller ner. I slutet av varje sprint kommer framstegen redovisas för beställaren. Utifrån gruppens kunskap inom teknologin som kommer användas samt projektets utformning, kommer tidsramen för varje sprint variera. När gruppen känner att tillräckliga framsteg har gjorts kommer dessa att redovisas. Regelbundna möten med handledare kommer att ske för att hålla koll på framsteg samt diskutera förändringar.

Arbetet organiseras med hjälp av en Scrum-board på GitHub. Scrum-boarden är indelad i tre delar: att göra, pågående och färdigt. Alla uppgifter som behöver utföras läggs upp under “att göra”. Eftersom gruppen är endast två medlemmar, ser båda till att de uppgifter som arbetar med flyttas till “pågående” för att undvika att båda arbetar på samma uppgift. När en uppgift är implementerad så meddelar gruppen varandra och den är redo att testas. Efter att uppgiften har testats flyttas den till “färdigt”. Detta gör att gruppen kan säkerställa att uppgiften fungerar som den ska samt få en överblick på vad som har gjorts och vilka som finns kvar att göra.

### **Användar-centrerad design**

För att skapa och utveckla användarvänlig applikation har användar-centrerad design processmetod använts. Detta fokuserar på att förstå användarnas behov, beteende och förväntningar för att skapa en produkt som är optimerad för användarupplevelsen. Inom användar-centrerad design finns det fyra faser som är viktiga att känna till: förstår sammanhanget för användning, specificerar användarbehov, designar lösningar och utvärderar lösningarna mot kraven [7].

I detta projekt fanns det en produktbeskrivning som ger sammanhanget för användning samt krav och specifikationer från beställaren. Således kommer gruppen påbörja processen att designa lösningar, vilket innebär att designa applikationen så att den uppfyller kraven och specifikationerna. Under designfasen kommer gruppen att skapa en prototyp. Genom att skapa olika skärmar kan gruppen testa idéer och applikationens funktionalitet för att säkerställa att applikationen uppfyller kraven.

När alla gruppmedlemmar är nöjda med prototypen skickas den till beställaren för att utvärdera om applikationen uppfyller förväntningarna. Feedback från utvärderingen fungerar som användarupplevelse, vilket innebär att hur användaren uppfattar, reagerar med applikationen och hur den uppfyller deras förväntningar. Eftersom beställaren undervisar studenter inom sjöfart, har hon expertis om vad studenter förväntar sig av applikationen. Feedbacken kommer sedan att användas för att förbättra prototypen. Denna iterativa process säkerställer att design-lösningarna uppfyller kraven innan de implementeras.

### **Användbarhetstest**

För att utvärdera användbarheten och användarupplevelsen av applikationen kommer ett användbarhetstest använts. Syftet med användbarhetstestet är att identifiera eventuella användarproblem, att upptäcka förbättringsmöjligheter och att lära sig om användarnas beteende och preferenser. Inom användbarhetstester finns det tre viktiga element: facilitator, uppgifter och deltagare [8]. Facilitatorn kommer att guida, ge instruktioner och svara deltagarnas frågor under testprocessen. Dessutom observerar och dokumenterar deltagarnas interaktion med applikationen. Testuppgifterna utgör de aktiviteter som deltagarna kommer utföra när de använder applikationen.

Antalet deltagare rekommenderas vanligtvis att ha mellan 5 och 12. Det kan dock variera beroende på studiemål, budget, tid och testets komplexitet. Dessutom bör deltagarna vara personer som kommer att använda applikationen regelbundet. Alternativt kan deltagarna ha en liknande bakgrund som målgruppen eller samma behov [8]. I detta projekt är målgruppen Chalmersstudenter som studerar frakt och marin teknologi. Således kommer projektgruppen att sikta på att involvera högskolestudenter med fokus på sjöfart.

Tiden som deltagarna spenderar på varje uppgift kommer att spelas in och jämföras med den tid som uppgiften förväntas vara avklarad inom. Om en uppgift slutförts korrekt och inom den angivna tidsramen räknas det som att den har avklarats och markeras som 1, annars markeras det som 0. Genom andelen avklarade uppgifter

kan applikationens effektivitet bedömas. Om uppgifterna slutförts korrekt kan det indikera att deltagarna har en positiv upplevelse. Däremot om flera deltagare misslyckas på samma uppgift, kan det betyda att deltagarna stöter på svårigheter och att applikationen behöver förbättras.

System Usability Scale (SUS) standard metod används för att skapa en enkät med 10 frågor (se tabell 2, fråga 1-10) som deltagarna svarar på efter de har slutfört testuppgifterna [9]. Enkätfrågorna besvaras med en skala från 1 till 5, där 1 motsvarar "håller inte alls med" och 5 motsvarar "håller helt med". SUS standard metod anger att frågor med jämnt nummer skall ha en negativ ton och frågor med udda nummer ska ha en positiv ton. Genom deltagarnas svar kan en användbarhets poäng på en skala från 0 till 100 beräknas genom att för varje fråga med udda nummer subtrahera 1 från poängen och för varje fråga med jämnt nummer subtrahera värdet från 5. Därefter summeras den genomsnittliga poängen från frågorna och multipliceras med 2.5. Enligt metoden innebär poäng under 68 vanligtvis att applikationen har allvarliga brister och kommer att behöva förbättras. Poäng över 80.3 betyder vanligtvis att deltagarna uppskattar applikationen. Utöver detta kommer deltagarnas feedback och förbättringsförslag samlas in genom två ytterligare frågor (se tabell 2, fråga 11-12). Syftet med dessa frågor är att få ytterligare insikt i deltagarnas upplevelser, identifiera specifika problem för att ta reda på förbättringsområden.

# 3

## Teknisk Bakgrund

### 3.1 Flutter

Flutter är ett open-source ramverk skapat av Google med syfte för applikationer som fungerar på flera plattformar med samma kodbas [10]. Byggstenarna i en Flutter applikation kallas för widgets. En widget ansvarar för allt användaren ser i applikationen. Användningsområdena för widgets varierar, det kan till exempel vara funktionella widgets som knappar, eller widgets som ändrar utseende för applikationen. Widgets kan kombineras för att skapa en komplex layout. Det finns två olika typer av widgets, stateful och stateless. Stateful widgets har ett ändringsbart tillstånd som kan uppdateras vilket ger ändringar i användargränssnittet. Stateless widgets har inget ändringsbart tillstånd, vilket betyder att de inte kan uppdateras med ny information om de inte byggs om från en stateful widget. En widget innehåller en build-metod, som bestämmer utseende för widgeten med hjälp av andra, mindre komplexa, widgets. För en stateful widget kallas build-metoden när ramverket meddelas att dess inre tillstånd uppdaterats med hjälp av setState-metoden.

Det finns flera anledningar till valet av Flutter istället för ett annat ramverk som React, som också gör det möjligt för utveckling till flera plattformar [11]. Flutter är byggt med inspiration av React och har flera funktioner som saknas i ramverket. Till en början är Flutters widgets inte OEM (Original Equipment Manufacturer), vilket betyder att utseendet på en Flutter applikation inte ändras beroende på operativsystem. Detta leder till mer kontroll hos utvecklaren för hur applikationen bör se ut och kännas. Dessutom har Flutter en widget för navigation mellan olika skärmar, till skillnad från React där utvecklaren behöver ett paket för att uppnå lika smidig navigation.

### 3.2 MVC

Programkodens arkitektur följer Modell-Vy-Kontroller (MVC)-mönstret där modellklasser innehåller programmets data, vy-klasser innehåller grafiska komponenter som visar datan för användaren och kontroller klasser hanterar användarens input för att manipulera data i modell-klasser [12]. MVC-mönstret gör det enkelt att implementera ändringar i källkoden genom att dela upp front-end och back-end. På så sätt påverkas inte front-end av ändringar i back-end och viceversa.

I Flutter kan MVC-mönstret uppnås genom att modell är en ren Dart-klass, kontroller är en stateful widget och vy är en stateless widget. En vy-klass innehåller endast grafiska element och refererar till modell-klassen för värden den behöver visa. En kontroller-klass innehåller funktioner som uppdaterar modell-klassen via `setState`-metoden. Build-metoden för en kontroller-klass returnerar en vy-klass. För att vy-klassen ska kunna använda funktionerna i kontroller-klassen skickas dessa funktioner med i vy-klassens konstruktor.

### 3.3 Dart, Dartpaket

Flutter använder programspråket Dart, till skillnad från React som använder JavaScript. Ett benchmark visar att Dart i de flesta fall har lägre minnesutnyttjande men längre kompilationstid [13]. Dart utvecklades av Google för enkel utveckling av program till flera plattformar. Språket har mycket likheter med Java vilket gör att det är lätt att lära sig för Java-utvecklare. Dart har ett eget system för att hantera utvecklarpaket, `pub.dev`, där utvecklare kan dela paket med kod som uppfyller funktioner som saknas i Dart. Paketerna används genom att lägga till dem i en källfil, Dart hämtar sedan paketerna automatiskt. Enligt `pub.dev` hemsidan kan paket som laddas upp inte tas bort [14]. Detta ger utvecklare en trygghet i att deras program inte plötsligt slutar fungera. Paketerna är open-source.

# 4

## Genomförande

Detta avsnitt presenterar hur applikationen har utvecklats under projektets gång. Först presenteras kravinsamlingen. Sedan den slutliga prototypen samt målet för varje sida och de fartygskisser som har ändrats för att uppnå beställarens specifikationer. Dessutom beskrivs applikationsflödet samt de nya funktioner som har lagts till under implementeringsfasen. Vidare presenteras systemkonstruktion samt användning av Dart-paket. Efter det presenteras användares testning som används för att utvärdera applikationens funktionalitet och användarupplevelse.

### 4.1 Kravinsamling

I en användar-centrerad design process finns det processer som ska hjälpa utvecklare att hitta krav för systemet. Då det redan fanns en kravbild för applikationen samlas kraven för detta projekt från produktbeskrivningen som beställaren angett. Inledningsvis utgick dessa endast från SMCP-fraserna. Vid fortsatt diskussion utökades kravbild till att inkludera en lista med vanliga meningar för olika ämnen inom radiokommunikation, samt inlärning av delar på fartyg.

Första kravet från produktbeskrivningen är en ordlista för SMCP-fraserna. Ordlistan innehåller alla SMCP-fraser från [2] med tillhörande definition. Andra kravet är att användaren ska kunna träna på sitt uttal och hörförståelse. Detta uppnås genom att användaren spelar in sin egen röst och jämför med det korrekta uttalet. Att träna på fraser inom radiokommunikation uppnås genom att användaren lyssnar på vanliga meningar. Sista kravet om inlärning av delar på fartyg implementeras genom att användaren matchar ord med rätt plats i en bild. På önskemål av beställaren ges detta krav mest uppmärksamhet.

### 4.2 Prototyp

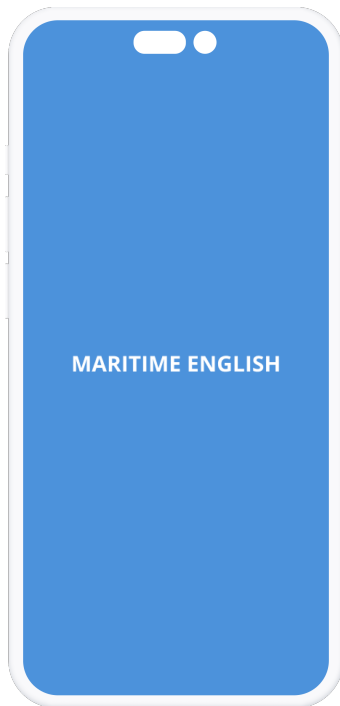
Prototypen skapas för att säkerställa att applikationen uppfyller de krav och behov som beställaren har specificerat. Prototypen fungerar som en tidig version av applikationen och används för att testa och utvärdera dess funktionalitet och användbarhet. Detta gör det möjligt att göra nödvändiga justeringar och förbättringar som beställarens förväntningar innan applikationen implementeras.

Nedanför visas slutliga prototypen av applikationen för implementering. Den presenterar hur sidorna i applikationen ser ut samt vilka mål som finns för varje sida.

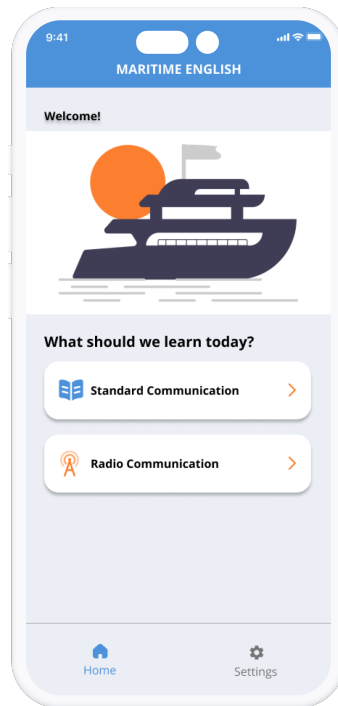
Beskrivningarna tar från vänster till höger.

Sida	Beskrivning
Splash	Splash-sidan visar först när applikationen öppnas. Dess mål är att visa användaren vad applikationen heter och ge en visuell indikation på att applikationen laddar.
Home	På home-sidan kan användaren välja olika ämnen som de vill lära sig mer om inom maritim engelska. Målet med den sidan är att ge användaren en enkel navigation möjlighet till det ämne som intresserar dem mest. Dessutom möjliggör sidan att fler ämnen kan läggas till i framtiden.
Standard Communication	Sidan finns för att presentera olika inlärningsaktiviteter som kan hjälpa användaren lära sig om SMCP-fraserna. Användaren kan också navigera till den aktivitet som intresserade dem mest.
General Phrases (Standard communication)	Sidan visar SMCP-fraserna och deras definitioner. Dess mål är att ge användaren snabb tillgång till alla fraser som används inom maritim standardiserad kommunikation, samt att användaren kan söka efter specifika ord med hjälp av sökrutan.
Practice Listening	På practice listening-sidan visas frasen och dess definition. Användaren får höra uttalet av frasen, spela in sig själv och lyssna på sitt eget uttal. Sidan visar också en alfabetisk ordlista som hjälper användaren att enkelt navigera till den lista som de vill öva på.
Radio Communication	Sidan presenterar olika inlärningsaktiviteter som finns i applikationen för att hjälpa användaren lära sig om maritim radiokommunikation.
General Phrases (Radio communication)	Sidan visar fraser som används inom maritim radiokommunikation. Målet är att hjälpa användaren att enkelt hitta fraser om specifika ämnen med hjälp av sökrutan.
The Ship	På the ship-sidan visar en bild av fartygsdelar samt en ordlista med vad delarna heter. Målet med sidan är att hjälpa användaren att lära sig olika delar av fartyg genom att dra orden till den matchande siffran som visas på bilden.
Pop up	Sidan finns för att bekräfta om användaren vill pausa övningen. Den informerar också till användaren att om de väljer att pausa övningen, kommer processen att förloras. Användaren behöver därför börja om från början nästa gång de vill öva.
Celebrate	Sidan visar ett meddelande som gratulerar användare för att ha slutfört övningarna.
Settings	På settings-sidan kan användaren anpassa applikationens utseende genom att välja mellan ljus och mörkt läge. Dessutom kan användaren dela applikationen med sina kontakter och läsa mer information om applikationen.

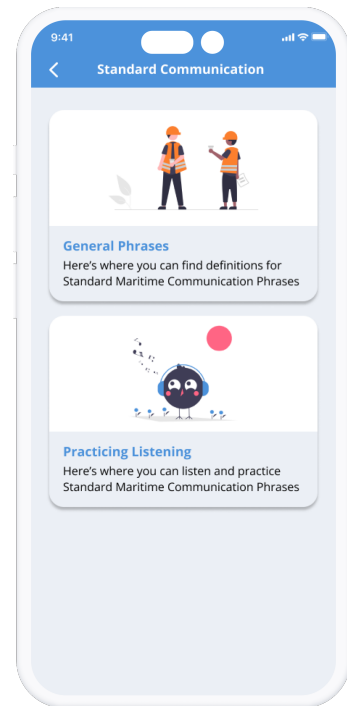
## 4. Genomförande



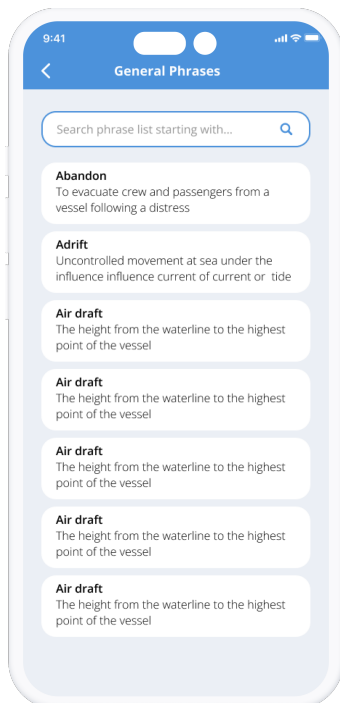
Splash



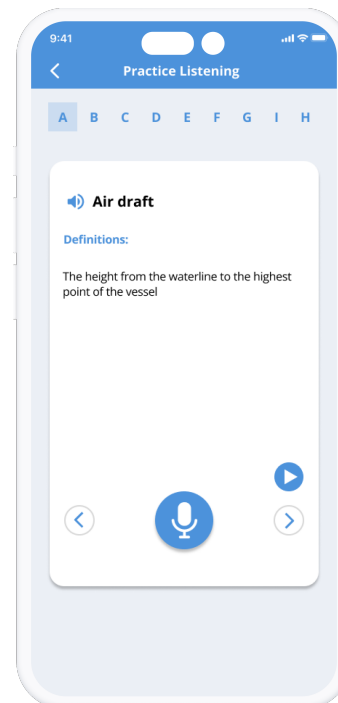
Home



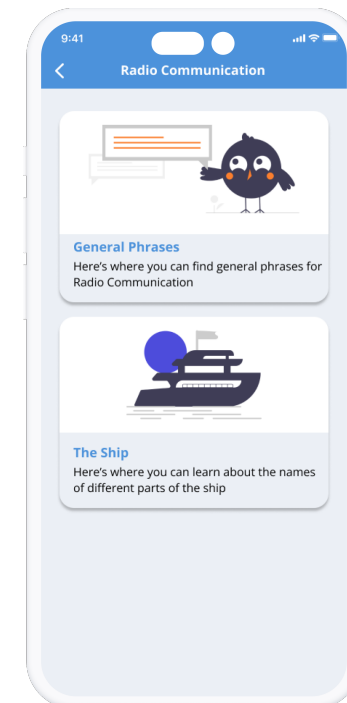
Standard Communication



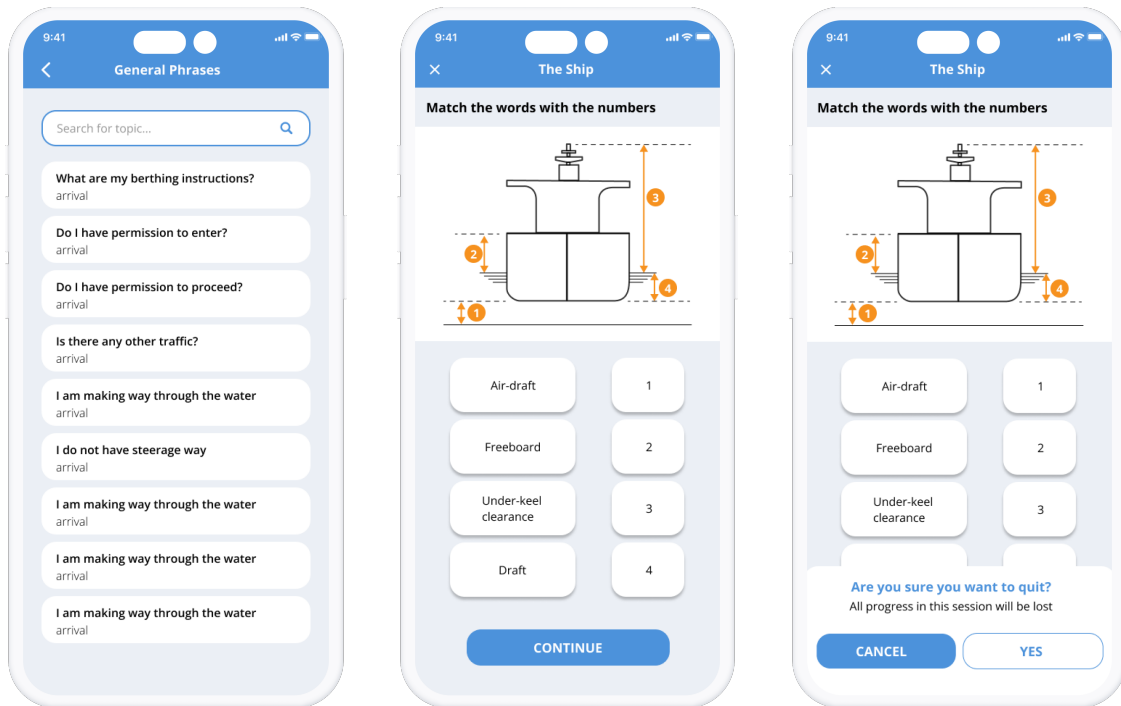
General Phrases



Practice Listening



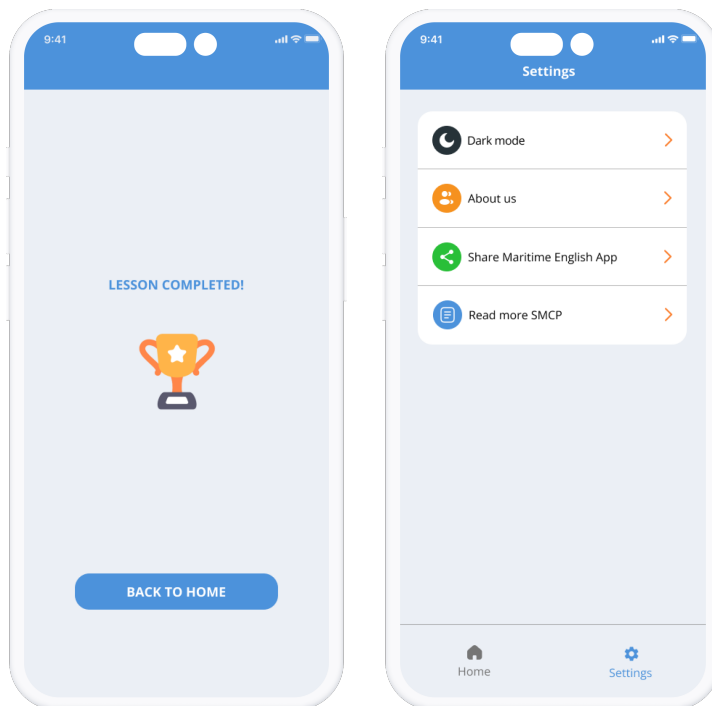
Radio Communication



General phrases

The ship

Pop up



Celebrate

Settings

Figur 1: Applikationsprototyp

#### 4. Genomförande

---

För att skapa en användarvänlig applikation har “Few Hues, Many Values”, “Corner Treatments”, “Prominent Done Button” och “Jump to Item” designmönster implementerats.

Inom designmönstret “Few Hues, Many Values” används en till tre primära färger i gränssnittet. Därefter skapas en färgpalett genom att välja olika ljusa nyanser inom dessa färger för att skapa en visuell hierarki [15]. I denna applikation har blått (hex: 4C92DB) och orange (hex: FB7F2E) valts som huvudfärger. Utifrån den blåa färgen (hex: 4C92DB) har en ljusblå nyans (hex: F5F7FA) valts. Den ljusblåa nyansen har använts som bakgrundsfärg i hela applikationen för att ge gränssnittet en lugnare känsla och skapa en visuell hierarki.

“Corner Treatments” är ett designmönster som tillämpas användning av runda hörn istället för vanliga räta vinklar på gränssnittselement [15]. Detta mönster används konsekvent i hela applikationen för att ge den ett karaktäristiskt utseende och göra gränssnittet mer intressant än med vanliga räta hörn.

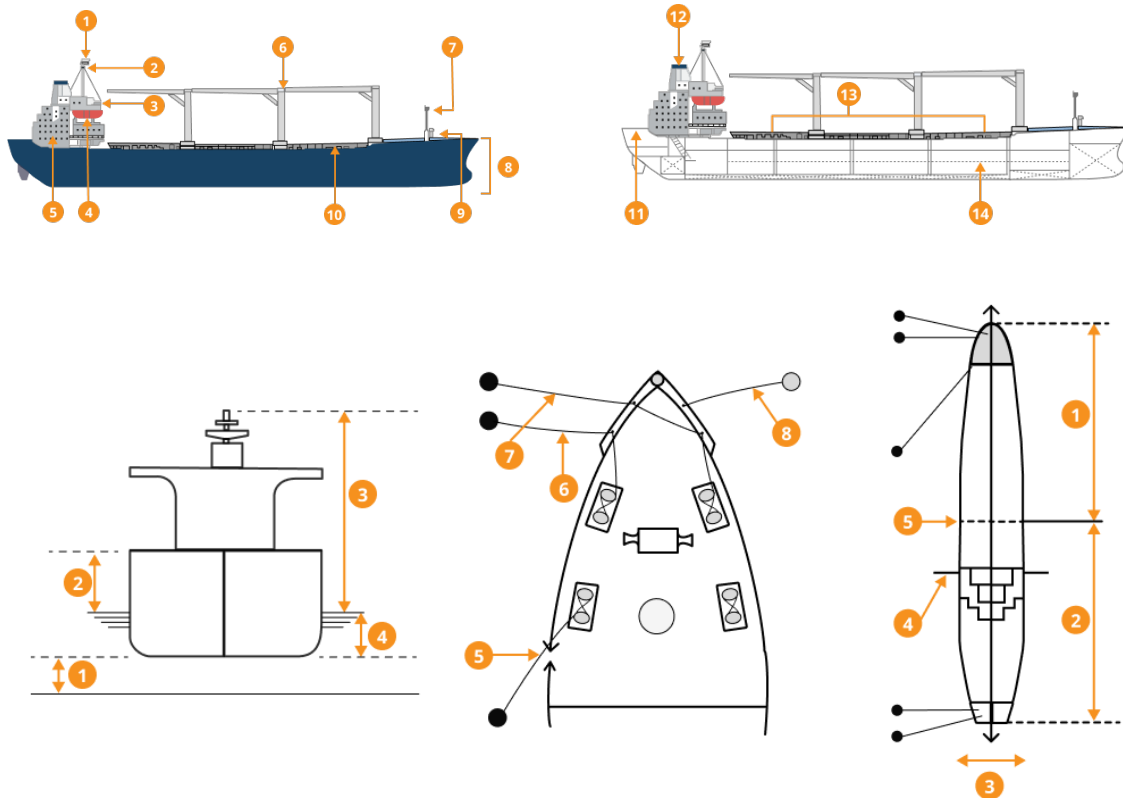
Inom designmönstret “Prominent Done Button” placeras en tydligt markerad knapp i gränssnittet. Syftet är att göra det enkelt för användare att avsluta eller bekräfta en specifik uppgift eller process i applikationen [15]. På “The Ship” sidan har detta designmönster använts för att tydligt markera och underlätta användarens navigering till nästa övningar genom att klicka på knappen “Continue”.

“Jump to Item” är ett designmönster som möjliggör snabb navigering genom att tillåta användare att söka och hoppa direkt till specifika objekt i applikationen [15]. Detta mönster har tillämpats för “General Phrases” sidorna. Det hjälper användare att snabbt skriva in specifika ord de önskar och hoppa direkt till listan med dessa ord.

För att säkerställa applikationsinnehållet är läsbart för alla användare, inklusive användare med synnedsättning, har kontrasten mellan text och dess bakgrund som använts i applikationen kontrollerats innan implementeringen. Applikationen har använts den ljusblåa (hex: F5F7FA) och vita (hex: FFFFFFFF) som bakgrund för svart (hex: 000000) text. Kontrasten mellan dessa färger har kontrollerats med hjälp av WebAIMs verktyg.

### 4.3 Fartygsskisser

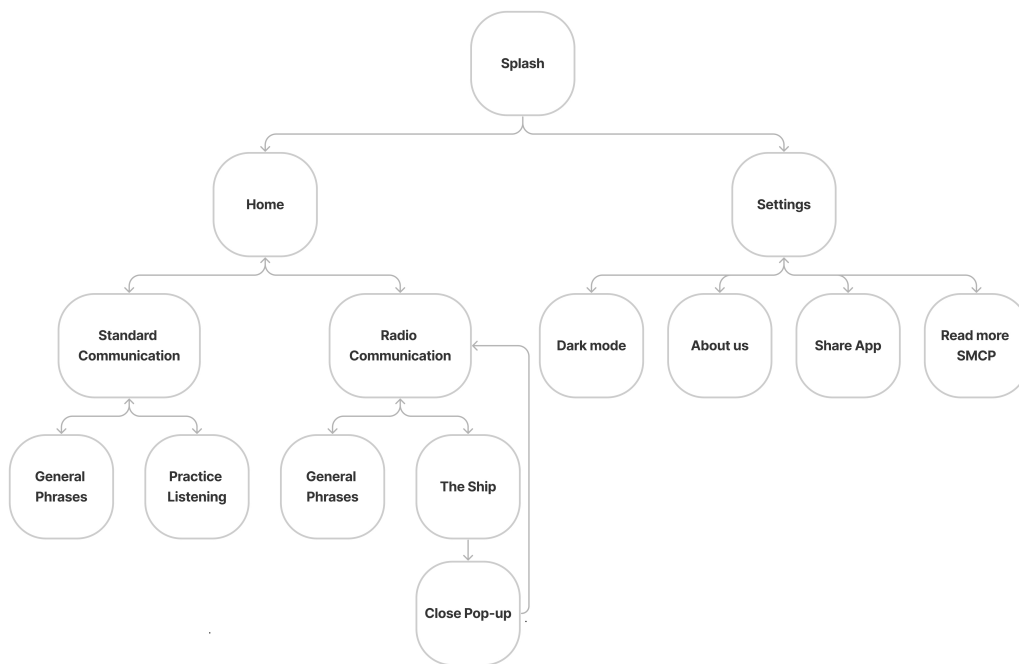
I övningen om fartygsdelar används bilder på fartyg med utsatta pilar till olika delar. Bilderna är baserade på material från IMO [Bilaga A] och MarEng-programmet [Bilaga B]. Bilderna har modifierats för att passa ändamålet i applikationen. Genom att modifiera bilderna kan övningen omstruktureras för att undvika onödiga upprepningar samt uppfylla beställarens förväntningar. I figur 2 visas de fartygsskisser som används i applikationen.



Figur 2: Fartygsskisser

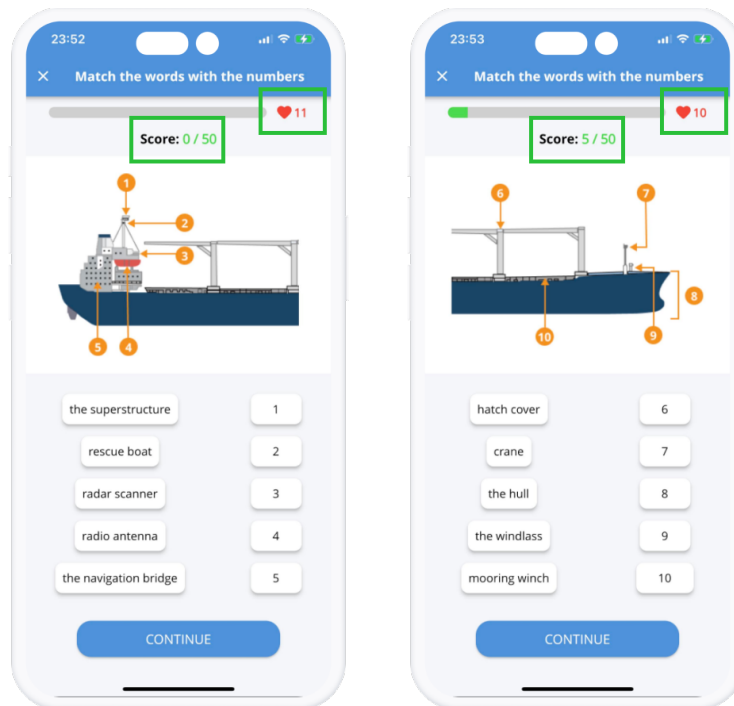
## 4.4 Applikationsflöde

I figur 3 visas applikationsflödet som ger en överblick över navigeringen mellan de olika sidorna i applikationen. Pilarna indikerar hur användaren kan ta sig från eller till en sida.

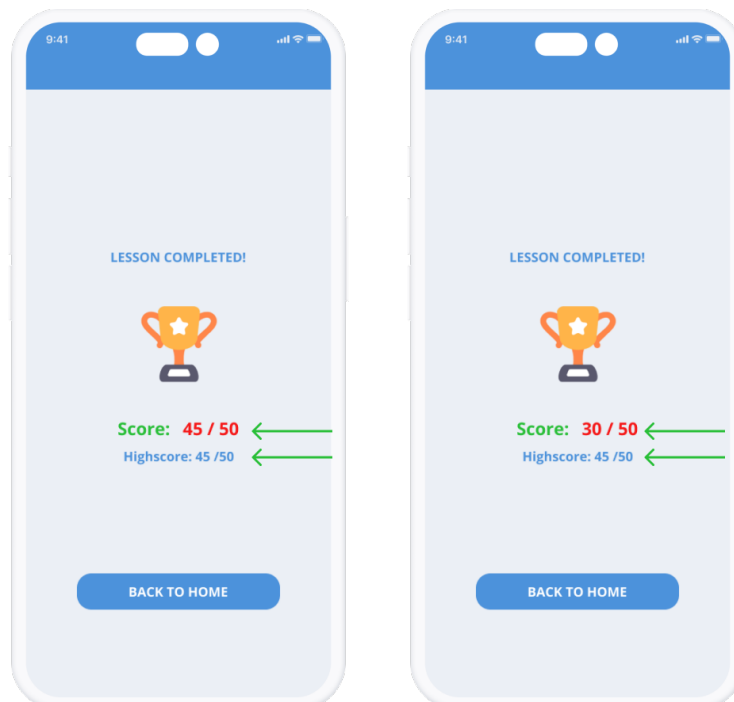


Figur 3: Applikationsflöde

Under implementeringsfasen har nya funktioner lagts till på vissa sidor för att förbättra applikationens funktionalitet och användarupplevelse. Sidorna behåller samma mål som beskrivs i prototypen. För att göra The Ship-övningen mer underhållande har två andra funktioner lagts till. En poängberäkning, där användare får en poäng varje gång de matchar rätt par av ord och nummer. Om användaren matchar fel förlorar de en poäng. Dessutom visar sidan antal övningar som användare behöver genomföra och vilken övning som är aktuell (se figur 4). En annan funktion har implementerats för att spara den högsta poängen som användaren har uppnått. På Celebrate sidan, utöver att visa ett meddelande som gratulerar användaren för att ha slutfört övningen, visas också användarens totala poäng samt deras högsta poäng hittills (se figur 5).



Figur 4: "The ship" sidan med nya funktioner



Figur 5: "The Celebrate" sidan med nya funktioner



som länkar mellan användargränssnittet och applikationens logik genom att förmedla användarens inmatning från vy-klasserna till modell-klasserna. Detta innebär att widgets i vy-klasser som hanterar användarinmatning, använder sig av metoder i kontroller-klasser för att uppdatera modell-klassen. Programmet har sex olika kontroller: NavigationController, PhrasesController, SentencesController, PracticeListenController, SettingsController och ShipExController. Klassernas build-metod returnerar tillhörande vy-klass.

Anledningen till att programmet har flera kontroller-klasser är Flutter's sätt att hantera tillstånd och navigation. Eftersom setState-metoden endast bygger om widgeten som kallas i build-metoden och navigation sker utanför denna metod, behövs en kontroller-klass till varje vy-klass vars information behöver uppdateras med ny data.

Vy-klasserna representerar det grafiska gränssnittet och datan i applikationen. Programmet har en vy-klass för varje ny sida i applikationen. Dessa klasser är: Homepage, Glossarypage, PracticeListen, Phrasescmp, Radiocompage, Radiosentences, Settings, ShipExercise, Celebrate och About.

Data som används i programmet är en ordlista med definition för SMCP-fraser och en lista med meningar för radiokommunikation. Programmet använde även en lista med bilder på fartyg och tillhörande lista med delar på fartyget för ship-övningen.

### 4.6 Använda Dart-paket

För att hantera persistens används Dart-paketet `shared-preferences`. Paketet innehåller kod för att lagra data av typerna: `int`, `double`, `bool`, `String` och `List<String>`. Datan lagras och hämtas med hjälp av 'key-value' par. Paketet är endast ett bra val för data som inte kräver säkerhet eller kan riskeras att förloras. I programmet lagras endast en `bool` för mörkerläge och en `int` för highscore på ship-övningen.

För att hantera ljudinspelning används paketet `flutter-sound`. Inspelningen sparas som en temporär fil på enheten och vid uppspelning används `audioplayer` paketet för att spela upp filen. Programmet efterfrågar tillstånd för inspelning med hjälp av `permission-handler` paketet.

Paketet `flutter-tts` används för att läsa upp SMCP fraser, fartygsdelar och meningar för radiokommunikation. Detta används som en kortsiktig lösning för att kunna testa applikationen. De korrekta inspelningar som är tänkta att användas i applikationen kan inte användas i denna version på grund av licenser.

För att öppna en länk till SMCP hemsidan från applikationen används paketet `url-launcher`.

Andra paket inkluderar `cupertino-icons`, `google-fonts`, `animated-splash-screen`, `lottie` och `flutter-screenutil`. Dessa paket kommer med nya möjligheter för utseende i applikationen.

### 4.7 Användarutvärdering

För att utvärdera applikationen används ett användbarhetstest [8], [16]. Syftet med detta är att identifiera problem som användarna kan stöta på när de använder applikationen, samt att få insikter om hur användarvänligheten kan förbättras [17]. Testet utförs med fyra olika deltagare vid olika tillfällen. Enda kriteriet för deltagare är att de är studenter på högskolenivå.

För utvärderingen har sex olika uppgifter skapats, där deltagaren ombes att utföra uppgifterna i ordning. Under utvärderingens gång observeras och dokumenteras deltagarens interaktion och reaktion för att utvärdera deras framgång i att genomföra uppgifterna. I tabell 1 presenteras uppgifterna, scenario vid avklarad uppgift och den tidsram som uppgiften bör klaras av inom. Tidsramen har bestämts baserat på uppgiftens komplexitet.

Tabell 1: Beskrivning av testuppgifter i den ordning som användare kommer utföra

Uppgift	Uppgiftsbeskrivning	Scenario vid avklarad uppgift	Tidsram (sekunder)
1	Låt oss säga, idag när du öppnar applikationen vill du lära dig fraser inom standard kommunikation. Kan du hitta de fraserna och dess definitioner?	Steg 1: Vid homepage, väljer "standard communication" Steg 2: Väljer "general phrase" Steg 3: Hittar fraser och dess definition	60
2	När du har hittat alla standardfraser, kan du söka efter en lista av fraser som börjar med bokstaven C?	Steg 1: Klickar på sökrutan Steg 2: Skriver in bokstaven C	60
3	Antag att du har lärt dig definitioner av alla fraser inom standard kommunikation. Nu vill du träna på både uttal och hörförståelse. Kan du hitta var du kan träna på det? Kan du hitta var du kan lyssna på modersmålstalare? Vad du kan göra när du vill spela in ditt uttal och sedan lyssna på inspelningen?	Steg 1: Från hemsidan, klickar på "standard kommunikation" Steg 2: Väljer "practice listening" Steg 3: Klickar på "volume icon" för att lyssna på modersmålstalare Steg 4: Klickar på "micro icon" för spela in Steg 5: Klickar på "play icon" för lyssna på uppspelningen	180
4	Den här gången vill du lära dig fraser inom radiokommunikation med ett specifikt ämne, nämligen "radar". Kan du hitta en lista över dessa fraser?	Steg 1: Från hemsidan, klickar på "radiokommunikation" Steg 2: Väljer "general phrases" Steg 3: Klickar på sökrutan Steg 4: Skriver in "radar"	60
5	Nu vill du lära dig om olika delar av fartyget. Kan du hitta var du kan göra övningen och hur du ska utföra den? Kan du ta reda på att du har genomfört övningen korrekt eller inte? Vad kan du göra om du vill ta en paus (dvs stänger av the ship sidan) efter du har gjort några övningar?	Steg 1: Från hemsidan, klickar på "radiokommunikation" Steg 2: Väljer "the ship" Steg 3: Drag ordet med matchande nummer på bilden Steg 4: Användare förklarar hur de kan ta reda på om de har genomfört övningen korrekt eller inte (genom score som visas på skärmen) Steg 5: Användare klickar på "close icon"	360

#### 4. Genomförande

---

6	Du vill nu sätta på mörkerläge, kan du hitta vart du sätter på det?	Steg 1: Klickar på “back arrow icon” för att komma tillbaka homepage Steg 2: Klickar på “settings” Steg 3: Sätter på mörkerläge	90
---	---	---	----

Efter att användare har slutfört samtliga uppgifter ombeds de att fylla i en enkät. Enkäten består av 12 frågor som innehåller 10 frågor om användares åsikter om applikationen samt två frågor om eventuella förbättringsförslag. I tabellen 2 visas enkätfrågorna. Svaren bedöms på en skala från 1 till 5, där 1 innebär att de helt inte håller med och 5 innebär att de helt håller med.

Tabell 2: Beskrivning av enkätfrågor

Fråga Nummer	Fråga
1	Jag skulle vilja använda den här applikationen ofta.
2	Jag tyckte att applikationen var onödigt komplicerad.
3	Jag tyckte att applikationen var lätt att använda.
4	Jag skulle behöva stöd av en teknisk person för att kunna använda den här applikationen.
5	Jag tyckte att de olika funktionerna i applikationen var väl implementerade.
6	Jag tyckte att det var för mycket inkonsekvens i den här applikationen.
7	Jag tror att de flesta skulle lära sig att använda den här applikationen väldigt snabbt.
8	Jag tyckte att applikationen var väldigt krånglig att använda.
9	Jag kände mig väldigt säker på att använda applikationen.
10	Jag behövde lära mig många saker innan jag kunde komma igång med applikationen.
11	Vad var din favoritdel av applikationen? Minst favorit?
12	Finns det något som du skulle vilja ändra eller förbättra i applikationen?

# 5

## Resultat

Produktutvecklingen har skett genom en användar-centrerad design process. Denna process innebär att projektet börjar med att samla in krav för systemet. Eftersom beställaren hade en kravbild färdig inför projektet anses detta steget vara avklarat. Nästa steg i processen var att utveckla en prototyp. Prototypen har skapats med hjälp av online-verktyget Figma och har sedan godkänts av beställaren. Därefter påbörjades implementeringen. Detta har skett genom agil mjukvaruutveckling. Syftet med denna modell är att organisera arbetet utefter prioriterade uppgifter.

Resultatet är en fungerande applikation som kan köras i både iOS och Android samt har utvecklats baserat på den designade prototypen (se figur 1 och figur 4). Applikationen uppfyller beställarens krav, vilket inkluderar definitioner av SMCP-fraser, möjlighet att lyssna på uttalet av SMCP-fraser, möjlighet att spela in och jämföra eget uttal av SMCP-fraser samt övning för olika delar av fartyg.

Nästa steg i processen innebär användarutvärdering. För att utvärdera applikationens användbarhet genomfördes ett usability-test med studenter på högskolenivå. Testet innefattar sex olika uppgifter samt ett frågeformulär. Syftet med testet var både att utvärdera ifall applikationen betedde sig som utvecklare väntat, men också att hitta eventuella problem och möjligheter till förbättring.

Tabell 3: Tiden som deltagarna spenderar på varje uppgift och dess genomsnittliga tid

Fråga	1(60s)	2(60s)	3(180s)	4(60s)	5(360s)	6(90s)
Deltagare						
1	19	6	50	20	160	10
2	20	15	30	40	130	6
3	17	25	32	25	190	15
4	5	4	15	22	60	5
Genomsnitt	12.25	12.5	31.75	26.75	135	9

I tabell 3 visar tidsramen som varje uppgift förväntas vara avklarad inom, den tid som deltagarna har spenderat på uppgiften samt den genomsnittliga tiden för uppgiften bland alla deltagarna. Dessa resultat används sedan för att beräkna andelen avklarade uppgifter (se tabell 4).

Tabell 4: Resultat från användbarhetstest

Fråga	1	2	3	4	5	6
Deltagare						
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	1
Andelen avklarade uppgifter	100%	100%	100%	100%	100%	100%

I tabell 4 visas resultatet efter att deltagarna utfört uppgifterna. Där 1 betyder att deltagaren har lyckats att slutföra uppgifter inom tidsramen och 0 betyder att deltagaren misslyckats. Därefter beräknas andelen avklarade för varje uppgift. Resultatet visar att andelen avklarade uppgifter är 100%. Detta tyder på en hög framgångsgrad där alla deltagarna har klarat testuppgifterna inom tidsramen.

Tabell 5: Resultaten från System Usability Scale (SUS)

Fråga	1	2	3	4	5	6	7	8	9	10	Poäng
Deltagare											
1	5	1	1	1	5	1	5	1	5	1	
2	4	1	5	1	5	1	5	1	5	1	
3	5	1	5	1	5	1	5	1	5	1	
4	5	1	5	1	5	1	5	1	5	1	
Genomsnitt	3.75	4	3	4	4	4	4	4	4	4	38.75 * 2.5 = 96.88

Efter att testet har genomförts, skickas enkäten (se tabell 2) till deltagarna. Där får deltagarna besvara frågorna på en skala från 1 till 5, där 1 betyder att de inte alls håller med och 5 betyder att de håller helt med. I tabell 5 visas svaren från deltagarna. Utifrån dessa svar kan användbarhet poängen beräknas. Det slutliga resultatet är 96.88.

Deltagarna ger även nya perspektiv kring förbättring av applikationen genom svar till de två sista frågorna på enkäten. En åsikt som framkommit från samtliga deltagarna är att det behövs fler övningar. Det bör finnas ett bättre sätt att träna på SMCP-fraserna. En annan åsikt är att övningen för delar på fartyg kunde vara lättare att förstå. Till exempel försökte deltagare klicka på ord och siffra för att matcha, istället för att dra. En ytterligare åsikt som framkommit är möjligheten att lyssna på fraserna från verkliga personer istället för text-to-speech.

# 6

## Diskussion

### Slutsats

Resultatet från utvärderingen gav att applikationen inte är komplicerad, eftersom deltagarna klarade uppgifterna snabbt och utan onödiga klick. Denna positiva feedback bekräftade att applikationen fungerar som förväntat när det gäller användbarheten. Poängen från SUS enkäten indikerar att applikationen har hög användbarhet och att användarna uppskattar applikationen. De två sista enkätfrågorna gav flera möjligheter för vidareutveckling och förbättring.

### Kritisk Diskussion

Genom att följa en användar-centrerad design process har vi kunnat uppfylla de krav som är nödvändiga för applikationen. Eftersom vi bestämde att kraven från beställaren var tillräcklig startpunkt för projektet användes ingen föreslagen metodik för kravinsamling. Detta ledde till krav som inte följde en viss standard. Konsekvenserna av detta blev krav som var svårare att implementera eftersom utformningen var tvetydig. Hade vi använt en metodik för framställning av kraven hade implementeringsprocessen kunnat vara mer lätthanterlig och gått snabbare.

Genom att använda ett agilt arbetssätt i implementeringsprocessen hade vi bra koll på vilka uppgifter som skulle genomföras härnäst. Detta ledde till att vi kunde visa resultat snabbt. En konstant kontakt med olika parter i projektet gjorde att förändringar kunde påpekas snabbt vilket ledde till att det mesta av arbetet inte behövdes tas bort eller göras om. En metod inom agilt arbete som vi inte tog användning av, men som kunde hjälpa projektet var korrekt utformning av user stories. Uppgifterna som lades till scrum-brädan hade inga kriterier för att anses som avklarade, utan märktes som avklarad när utvecklaren kände att uppgiften var det. Detta ledde till att vissa funktioner behövdes revideras, dock var det inga stora förändringar som behövdes och det tog inte mycket tid av projektet.

Genom att skapa applikationen med Flutter har vi uppnått ett användargränssnitt som är responsivt och attraktivt, utan att behöva koda dessa aspekter själva. Användning av Flutter gjorde även att utvecklingen gick snabbt. Framst på grund av likheten mellan syntax i Dart och Java, då båda gruppmedlemmar hade erfarenhet inom Java, innan projektet påbörjades. Flutter har även en funktion som gör det möjligt att uppdatera simulatoren med ny kod, utan att starta om programmet. Detta gjorde också att implementeringen skedde snabbt.

Något som förvärrade implementeringsprocessen var att koden inte följde MVC-

mönstret långt in i projektet. Detta ledde till en omstrukturering som tog mycket tid. Något som kan ha påverkat programoden är att funktionerna inte var enhets-testade genom projektets gång. Att koden ska vara enhetstestad bör ha varit ett kriterium för färdigimplementerade krav. Detta skulle gett oss ökad trygghet att applikationen fungerar som förväntat.

Användarutvärderingen hade helst gjorts med deltagare från sjöfartsutbildningar. Men på grund av svårigheter att hitta deltagare samt tidsbrist, nöjde vi oss med att göra utvärderingen med andra studenter. Konsekvenserna av detta blev att vi inte fick många tips på nya funktioner, eftersom deltagarna inte vet vad mer som används inom utbildningen. Något som kan ha påverkat enkätens svar är att deltagarna var bekanta, vilket kan ha påverkat deras svar positivt. Detta hade kunnat undvikas om enkäten fylldes i anonymt eller om deltagarna var främlingar. Dessa aspekter anser vi vara ett mindre problem eftersom applikationen kommer att vidareutvecklas i framtiden, med nya möjligheter för utvärdering. Vi tycker att utvärderingen var tillräcklig för att hitta vissa oklarheter inom applikationen.

För uppgifterna till användbarhetstestet hade en tidsgräns satts utefter att deltagarna inte var studenter från sjöfartsutbildningar. Tanken var att de skulle behöva ställa frågor om de olika funktionerna. Det visade sig att tidsgränserna var för höga vilket skulle kunna påverka slutsatserna som drogs från resultatet. Något som påvisar att applikationen inte är komplicerad är att ingen testuppgift hade stora variationer i genomförd tid bland deltagare. Anledningen till variation på fråga 5 var att testet utfördes på lite olika sätt. I sista testet utfördes endast ett försök att matcha ord med siffra i bild, medans i de föregående gjordes flera försök. Denna avvikelser hade kunnat undvikas om testuppgiften var förklarad i mer detalj.

Vår version är open-source vilket betyder att de inspelningar som fanns tillgängliga inte kunde användas, då de inte var open-source. Ansvar för korrekt uttal vid uppspelning av fraser ligger hos utvecklarna för den version som vidareutvecklas för användning inom utbildningen.

### **Samhälleliga, etiska och miljöaspekter**

Applikationen kommer vara tillgänglig på både Android och iOS vilket ger studenter möjlighet att direkt få tillgång till läromaterial och övningar på sina mobiltelefoner. Detta har en positiv inverkan på miljö genom att eliminera behovet av att skriva ut läromaterial på papper och därmed minskar användningen av papper. Projektet har en positiv inverkan på samhället genom att främja utbildning och hjälpa studenter att förbättra sin kunskap inom maritima engelska. Detta möjliggör för studenter att utveckla nödvändiga färdigheter för att kunna arbeta inom den globala sjöfartsindustrin. Projektet kommer att vara tillgängligt under en open-source licens. Detta möjliggör för både framtida studenter och andra intressenter att vidareutveckla vilket främjar samhällsutvecklingen inom utbildning.

Projektet är avgränsat att inte ta ställning till de pedagogiska aspekterna av språkinlärning, eftersom projektet fokuserar på att utveckla en teknisk applikation med

syfte att komplettera undervisningen på Chalmers utbildningar. SMCP-fraserna som visas i applikationen följer standarder som IMO har infört för maritim engelska. Detta gör att studenter får tillgång till korrekt och relevant material för sin utbildning.

Vidare har projektet följt Sveriges Civilingenjörsköringsförbunds hederskodex genom att endast använda material med open-source licenser under produktutvecklingen.

### **Vidareutveckling**

Med mer tid för projektet hade vi gärna implementerat fler övningar i applikationen. I nuläget kan användaren endast öva på hörförståelse och uttal av SMCP-fraserna samt delar av fartyg. I en slutgiltig version bör det finnas övningar för definition av SMCP-fraserna. En annan funktion som vi hade i backlog men inte hann med var att implementera så användaren kan repetera sina misstag. Detta hade sparat tid för användare som bara gör några få misstag, då de inte behöver göra om hela övningar för att lära sig allt. Eftersom applikationen använder text-to-speech för att spela upp uttal av fraser är det några uttal som blir felaktiga. Den slutgiltiga versionen av applikationen bör innehålla inspelningar med korrekt uttal.

Projektet kommer publiceras under en öppen licens eftersom detta gör det enkelt att vidareutveckla, med mer eller mindre engagemang från olika parter inom projektet. Eftersom syftet med applikationen är att främja utbildning var detta valet självklart för oss som utvecklare, då efterkommande projektgrupper har möjlighet att vidareutveckla projektet, vilket leder till större sannolikhet att applikationen kommer släppas som en färdig produkt. Valet av MIT licensen hindrar heller inte någons möjlighet att licensera en vidareutveckling på andra sätt.

### **Länk till projekt**

Projektet är tillgängligt via: <https://github.com/eliascalzone/maritimeEnglish-Flutter>

# Litteraturförteckning

- [1] C.Cole, P.Trenkner, “The yardstick for Maritime English STCW assessment purposes,” IAMU Journal 6(1), 13-28, 2009.
- [2] “Standard Marine Communication Phrases,” International Maritime Organization. [Online] Tillgänglig:  
<https://www.imo.org/en/OurWork/Safety/Pages/StandardMarineCommunicationPhrases.aspx>
- [3] “What is a Lottie?” lottiefiles.com, 2023. [Online] Tillgänglig:  
[https://lottiefiles.com/what-is-lottie?utm\\_medium=web&utm\\_source=navigation-what-is-lottie](https://lottiefiles.com/what-is-lottie?utm_medium=web&utm_source=navigation-what-is-lottie)
- [4] “unDraw” undraw.co, 2017. [Online] Tillgänglig:  
<https://undraw.co/>
- [5] “Contrast and Color Accessibility,” WebAIM web accessibility in mind, Jan. 09, 2021. [Online] Tillgänglig: <https://webaim.org/articles/contrast/>
- [6] A. Abrahamsson, O. Salo, J.Ronkainen och J. Warsta, “Agile Software Development Methods: Review and Analysis.” Finland, 2002 [Online] Tillgänglig:  
<https://arxiv.org/pdf/1709.08439.pdf>
- [7] “User centered Design,” Interaction Design Foundation [Online] Tillgänglig:  
<https://www.interaction-design.org/literature/topics/user-centered-design>
- [8] S.Dumas, J.Redish. A practical guide to usability testing. Intellect books, 1999.
- [9] B. John, SUS: A quick and dirty usability scale. Usability Eval. Ind.. 189, 1995.
- [10] Flutter, Version 2.2, [Software], 2017. Tillgänglig: <https://flutter.dev/>
- [11] W. Wu, “React Native vs Flutter, cross-platform mobile application frameworks,”. B.S. thesis, Dept. IT. Eng., Uni. Metropolia, Finland, 2018. [Online] Tillgänglig: <https://www.theseus.fi/bitstream/handle/10024/146232/thesis.pdf?sequence=1>
- [12] J. Bucanek, Model-view-controller pattern. Learn Objective-C for Java Developers, 353-402, 2009.
- [13] “Dark VS Javascript benchmarks,” Programming Language and compiler Benchmarks, May. 04, 2023. [Online] Tillgänglig:  
<https://programming-language-benchmarks.vercel.app/dart-vs-javascript>
- [14] “pub.dev policy,” pub.dev, 2023. [Online] Tillgänglig:  
<https://pub.dev/policy>
- [15] J. Tidwell, Designing Interfaces, 2nd ed., O’Reilly, 2011.
- [16] “Usability Testing 101,” NN/g Nielsen Norman Group, Dec. 01, 2019. [Online]. Tillgänglig: <https://www.nngroup.com/articles/usability-testing-101/>

- [17] J.Preece, Y.Rogers, H.Sharp, Interaction Design: Beyond Human-Computer Interaction, 2019.
- [18] “Contrast Checker,” WebAIM web accessibility in mind, [Online] Tillgänglig: <https://webaim.org/resources/contrastchecker/>
- [19] Flutter, Version 2.2, [Software], Google, 2017. Tillgänglig: <https://flutter.dev/>
- [20] Figma, [Software], Figma, Inc, 2016. [Online] Tillgänglig: <https://www.figma.com>
- [21] MarEng, [Software] Tillgänglig: <https://mkkdok.utu.fi/mat/mareng-old/index.html>

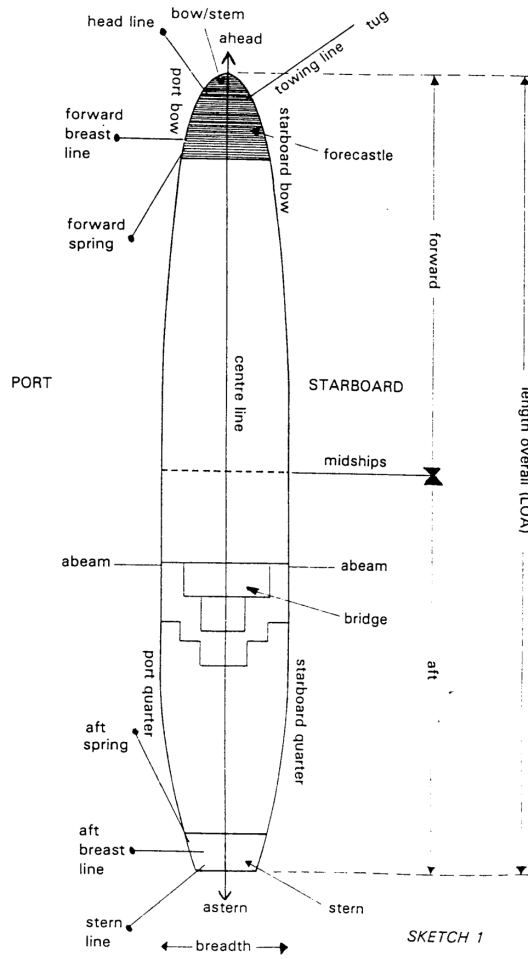
# A

## Appendix A

### Skisser från PDF-material

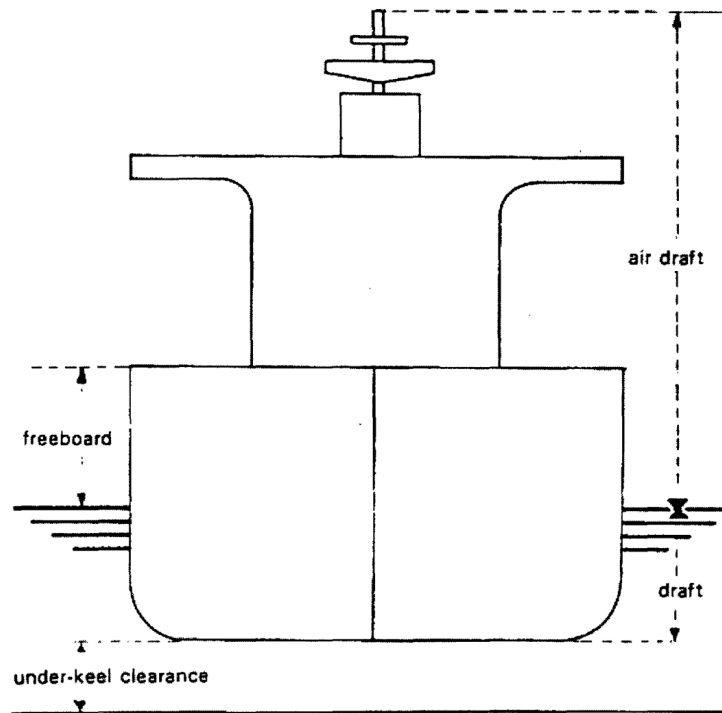
Resolution A.918(22)  
Adopted on 29 November 2001 (Agenda item 9)  
IMO STANDARD MARINE COMMUNICATION PHRASES  
- 27 -

A 22/Res.918

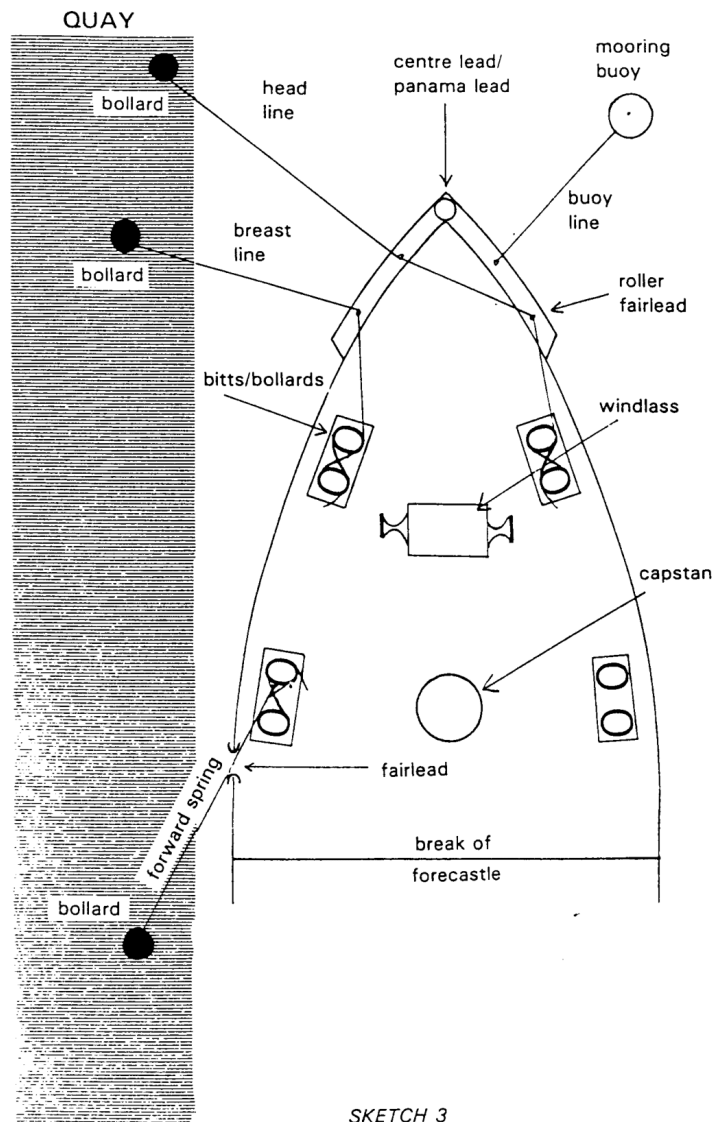


Resolution A.918(22)  
Adopted on 29 November 2001 (Agenda item 9)  
IMO STANDARD MARINE COMMUNICATION PHRASES  
- 28 -

A 22/Res.918



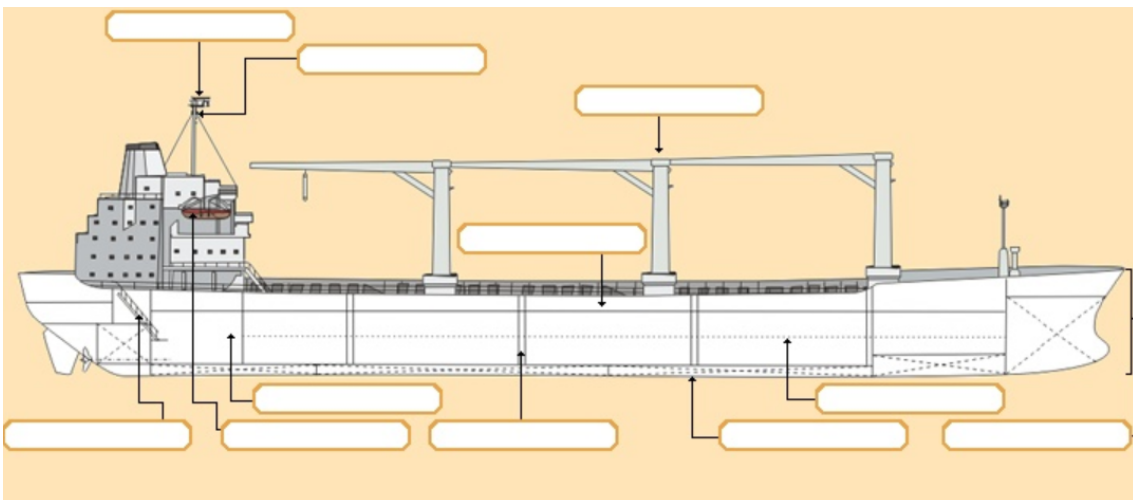
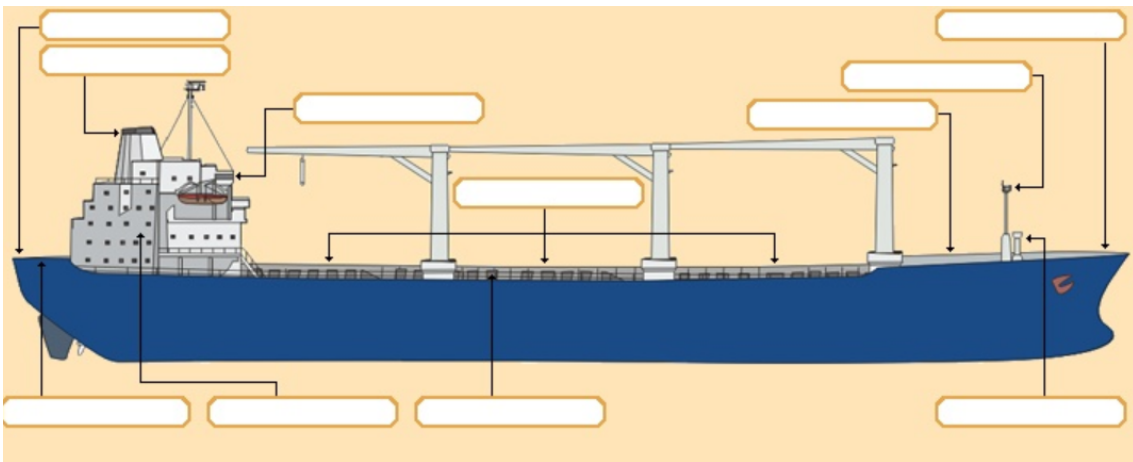
SKETCH 2



# B

## Appendix B

Skisser från MarEng



INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK  
CHALMERS TEKNISKA HÖGSKOLA  
Göteborg, Sverige  
[www.chalmers.se](http://www.chalmers.se)



GÖTEBORGS  
UNIVERSITET

---



**CHALMERS**