

# Kommunikation och användargränssnitt - Projektarbete

*Hitta i Tärpan*

Vera Fristedt Andersson - veran877

Marcus Johansson - marjo561

Alexander Bergman - alebe181

Christophe Hultén - chrhu055

Carl Håkansson - carha719

Utbildning: Civilingenjör i medieteknik

Kurs: TNM040, Kommunikation och användargränssnitt

Examinator: Camilla Forsell

Datum: 2017-12-11

Institutionen för teknik och naturvetenskap, Linköpings Universitet

# Innehållsförteckning

<b>1. Introduktion</b>	<b>3</b>
1.1. Frågeställningar	3
<b>2. Användargränssnitt</b>	<b>4</b>
2.1. Startside	4
2.2. Våningsplan	5
<b>3. Implementation</b>	<b>6</b>
3.1. Verktyg	6
3.2. Navigering	6
3.3. Sökfunktion	6
3.4. Hjälpmedel	7
3.5. Våningsplan	8
3.6. Kartor	8
3.7. Zoom-funktion	9
<b>4. Projektmetod</b>	<b>10</b>
4.1. Tidsplan	10
4.2. Arbetsfördelning	10
<b>5. Diskussion</b>	<b>11</b>
<b>6. Slutsats</b>	<b>13</b>
<b>7. Bilagor</b>	<b>15</b>
<b>8. Källor</b>	<b>16</b>

# 1. Introduktion

Syftet med detta projekt var att skapa ett användarvänligt och funktionellt gränssnitt som är anpassat för en specificerad målgrupp, skrivet i valfritt programmeringsspråk. Vi bestämde oss för att göra en mobilapplikation för studenter vid Linköpings Universitet och beskriver arbetsprocessen från idé till färdig produkt i denna rapport.

Då det kan vara svårt för nya studenter att hitta rätt på Campus Norrköping vid början av sina studier ansåg vi att det finns ett behov av en fullständig och lättläst karta över det. Gruppens medlemmar kom till insikt om att vi själva hade önskat oss en sådan applikation vid skolstarten och det var så idén grundades. För att hinna utforma ett väl fungerande gränssnitt innan deadline bestämde vi oss för att fokusera på byggnaden Täppan. Projektuppgiften som valdes var att utforma en applikation som skulle kunna underlätta för studenter att hitta till klassrum, grupprum och laborationssalar på de olika planen i Täppan. Applikationens syfte var alltså att agera som en karta över varje plan i Täppan, där användaren kan söka efter valfri sal för att få upp vilket plan salen ligger på och var på planet salen ligger.

Projektet påbörjades med att leta efter ett passande programmeringsspråk. Då Javascripts programmeringsbibliotek *React* hade använts tidigare i kursen söktes ett liknande bibliotek, anpassat för mobilapplikationer, och *React-Native* var då det bibliotek som valdes.

## 1.1. Frågeställningar

Det som uppskattades vara den största utmaningen i början av projektet var att vi på egen hand skulle behöva lära oss ett helt nytt programmeringsbibliotek på kort tid. Ett ytterligare orosmoment inför arbetet var att projektet inte skulle hinna bli klart under den angivna tiden. Detta på grund av att ingen i gruppen hade tidigare erfarenhet av att skapa en mobilapplikation och därmed var det okänt hur lång tid allt skulle behöva ta. Ur dessa funderingar uppstod följande frågeställningar:

- Hur svårt är det att skapa en mobilapplikation?
- Hur tidskrävande kommer det bli att lära sig det nya programmeringsspråket?
- Om vi ska kunna genomföra projektet behöver vi en planlösning över täppan, finns det?

- Kommer anpassning till olika telefoner vara ett stort problem?

## 2. Användargränssnitt

### 2.1. Startside

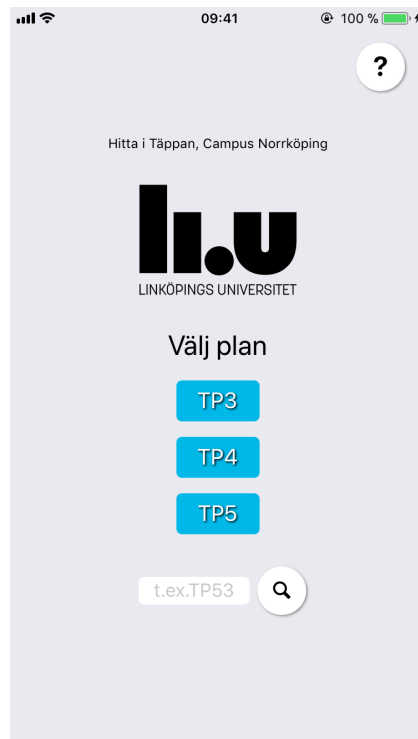


Bild 1. Mobilapplikationens startside, det första man ser när man öppnar appen.

Vid designen av startsidan har stort fokus lagts på bra mappning och synlighet. Tack vare ett stilrent första intryck och få distraktioner, framgår applikationens funktion tydligt (se Bild 1). Det finns en rubrik som kort beskriver användningen, Linköpings Universitets logotyp finns med så att användaren förstår att applikationen är relaterad till universitetet, tre knappar som representerar de olika våningsplanen i Täppan och en sökruta längst ner. De tre knapparna tillsammans med sökrutan agerar som signifier för den affordans som startsidan har. Då användaren klickar på önskat plans knapp, dyker en ny sida upp med en karta över det valda planet och tillhörande salar. Sökrutan tyder på att det finns möjlighet att söka efter önskad sal vilket framgår genom texten i rutan tillsammans med förstoringsglaslet bredvid. Önskas mer detaljerad information finner man det under knappen med frågetecknet som navigerar till en informationssida.

## 2.2. Våningsplan

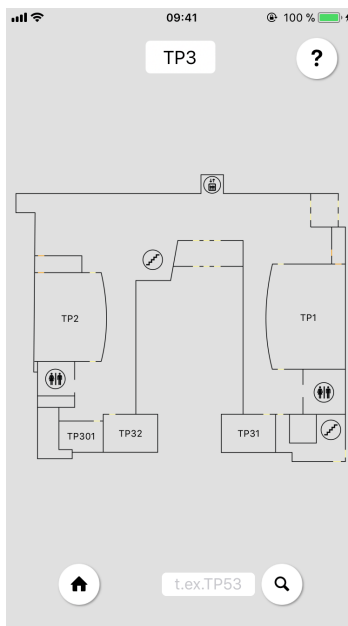


Bild 2. Bild över plan 3 i Täppan.

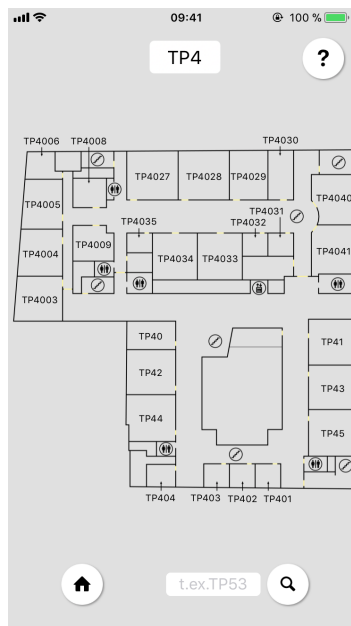


Bild 3. Bild över plan 4 i Täppan

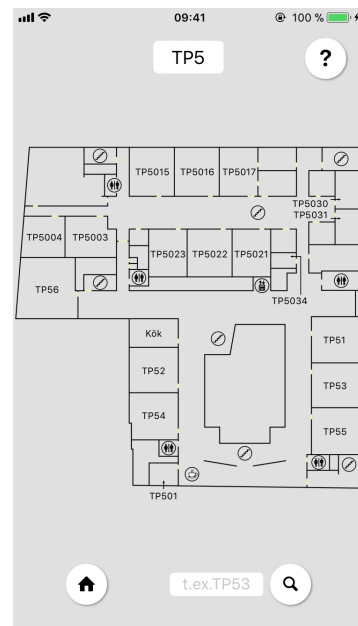


Bild 4. Bild över plan 5 i Täppan.

Precis som med startsidan så har mappning och synlighet spelat en viktig roll vid designen av sidorna med planen. Själva kartan är det som drar den huvudsakliga uppmärksamheten på respektive sida (se Bild 2, 3 & 4). Kartorna tar upp större delen av skärmen och är utformade utan detaljer så att de är enkla och därmed så enkla att förstå som möjligt. Till exempel har bord, stolar och dylikt inte placerats ut till skillnad från de ursprungliga planlösningarna som användes som mall (se Bilaga 1, 2 & 3). Dörrarna har färgkodats för att urskilja handikappdörrar från vanliga dörrar. För att undvika eventuella språkbarriärer har ikonerna för toalett, hiss och trappa universella symboler som kan uppfattas av alla. Eftersom appens målgrupp är studenter togs beslutet att endast skriva ut namnen på de salar som studenter har tillgång till. Om användaren önskar att se mer kartan mer detaljerat finns det möjlighet att zooma in och ut samt dra runt bilden precis som man kan göra med vanliga bilder på de flesta smartphones.

Det finns också en sökfunktion på varje plans sida för att användaren ska slippa behöva gå tillbaka till startsidan varje gång hen vill söka efter en ny lokal. Vill användaren istället navigera vidare till ett annat plan får de först återvända till startsidan via hemknapp för att sedan kunna välja önskat plan. Precis som på startsidan finns det en infoknapp i högra hörnet

som tar upp en ny sida med förklaringar. Alla dessa knappar tillsammans med textrutan har placerats i kanterna av skärmen för att lämna utrymme runt kartan så att den syns ordentligt.

### 3. Implementation

#### 3.1. Verktyg

Program som vi har dragit nytta av under arbetets gång är *Expo* och *GitLab*. *Expo* gör det möjligt att se hur applikationen ser ut i realtid via en simulator på datorn eller telefonen. Det är ett program som automatiskt uppdaterar utseendet på applikationen utefter förändringar i koden och om något blir fel i koden så slutar simulatoren att fungera. *GitLab* är ett versionshanteringssystem som har gjort det möjligt för alla gruppmedlemmar att sitta och koda i samma dokument på flera olika datorer genom att det självant skriver ihop allas förändringar.

#### 3.2. Navigering

I denna app har funktionen *StackNavigator* använts för att kunna navigera mellan startsidan, planen och informationssidan. *StackNavigator* är en inbyggd funktion i *React-Native* där man deklarerar hur många sidor som finns i appen och via knapptryckningar kan man enkelt komma till önskad sida. När man trycker på någon av knapparna markerade "TP3", "TP4" eller "TP5", skickas plannamnet in i *StackNavigator* som väljer att ta upp en ny sida med rätt karta och salar. Väljer man istället att söka efter en särskild sal så skickas salsnamnet in i *StackNavigator* som går igenom listan med alla salar i Täppan och tar fram det plan som salen ligger på.

#### 3.3. Sökfunktion

Sökfunktionen var en av de viktigaste funktionerna att implementera då den behövs för att uppfylla appens huvudsyfte. När användaren söker efter en särskild sal har appen programmerats på ett sätt som gör att den sökta salen markeras med en röd ruta så användaren enkelt kan se var salen ligger i förhållande till planet (se Bild 5).

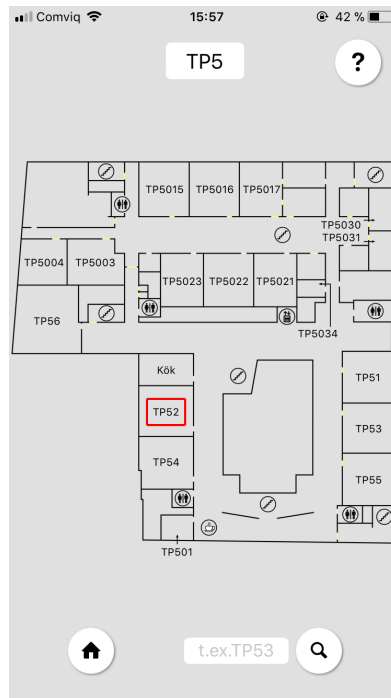


Bild 5. Exempel på hur en sökt sal visas på skärmen.

Sökfunktion tar in det som användaren skriver i sökrutan och analyserar om det finns någon sal med matchande namn. Funktionen har en for-loop som går igenom en .jsonfil som har alla salsnamn i Tappan listade. Innan loopen sätts en boolean-variabel till “false” som fungerar som en check om rummet som söks finns eller inte. Om rummet finns sätts variabeln till “true” och man navigeras till planet med den efterfrågade salen. Efter for-loopen kollar man om värdet fortfarande är “false” vilket innebär att salen inte har hittats. Användaren får då bra återkoppling genom att en pop-up dyker upp på skärmen som meddelar att den sökta salen inte hittades och att ett nytt försök bör göras.

### 3.4. Hjälpmedel

För att underlätta förståelsen hos användaren om hur saker och ting fungerar i appen finns det en del hjälpmedel. I sökrutan finns alltid en platshållare med texten “t.ex. TP53” för att användaren ska kunna förstå hur namnet på den efterfrågade salen ska skrivas. Sökfunktionen är även programmerad på så sätt att det automatiskt skrivs med stora bokstäver för att användaren ska slippa trycka på shift varje gång, något som hjälper både användaren och appen att undvika onödiga felsökningar.

För att användaren ska kunna se vad som skrivs i sökrutan har en funktion vid namn *KeyboardAvoidingView* använts. Den gör så att allt innehåll på sidan förflyttar sig upp för att tangentbordet inte ska täcka sökrutan vid användning (se Bild 6). Det finns dessutom en sida kopplad till frågetecken-knappen som ger användaren ytterligare information om det är någonting som inte framgår (se Bild 7).

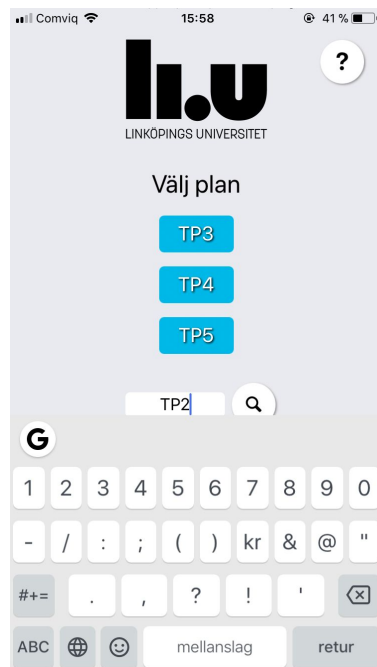


Bild 6. Visar sökning.



Bild 7. Visar informationssidan.

### 3.5. Våningsplan

Sidan med de olika planen har programmerats så att det plannamn som skickas med i *StackNavigator* tar fram tillhörande karta och skriver sedan ut planets salar på dess rätta koordinater. Detta möjliggörs genom att varje plan har en tillhörande lista med alla salsnamn samt vilken x- och y-koordinat varje sal har på kartan. Med hjälp av map-funktionen skrivs alla salsnamn ut på kartbilden och för varje sal ändras stilen på textrutan innan den skrivs ut. Stilen ändras på så sätt att textens placering på bilden anpassar sig efter den x- och y-koordinat som finns med i listan.

### 3.6. Kartor

För att skapa kartorna användes officiella planlösningar hämtade från LiU.se som mall. I Adobe Illustrator markerades alla väggar ut vilket resulterade i en mycket simplare karta som



enbart innehåller rummens konturer. Även ikoner för toalett, hiss, trappa och Studentfikat gjordes i Illustrator. Dörrar markerades med en färg som kontrasterar mot väggarna så att de enkelt kan kunna urskiljas. På TP3 finns handikappingångar vid salarna TP1 och TP2, så dessa dörrar markerades med en annan färg. En förklaring för denna färgkodning återfinns i den ovannämnda informationssidan.

### 3.7. Zoom-funktion

Det var för samtliga i gruppen en självklarhet att användaren skulle kunna zooma in på kartorna för att lättare kunna urskilja hur salarna på varje plan var placerade. Då projektet låg i startgroparna var det emellertid osäkert hur denna funktion skulle implementeras. Tidigt i implementationsprocessen antogs det att en kod från en tredje part skulle behövas för att förverkliga visionen, och arbetet inleddes med att hitta en lämplig s.k. "plugin" (kod som skapats av tredje part och delas öppet med alla som vill använda den för egna syften) för ändamålet. Problem uppstod nästan genast, då dessa plugins var svåra att installera och använda med gruppens begränsade erfarenhet. Det visade sig dock senare att react-native har en zoom-funktion, vilket gjorde att ingen tredjepartslösning behövdes. Funktionen kallas *ScrollView*, och är synnerligen lätt att implementera då den utformats likt en "behållare" vars innehåll går att zooma in på. Det enda som behövde göras var därför att lägga kartorna i behållaren och ange hur långt in respektive ut användaren tillåts zooma. Användaren kan med två fingrar "nypa" för att zooma ut, dra isär fingrarna för att zooma in och dessutom dra med ett finger på skärmen för att flytta runt kartan.

### 3.8. Ikoner

Ikoner användes extensivt i projektet för att kommunicera till användaren vilka delar av skärmen som ska interageras med för att använda olika funktioner. Som exempel kan en ikon användas för att återvända till hemskärmen från sidorna med kartorna. Ikonen är vit med ett hus på, vilket för många symboliserar "hem". Bekanta former som denna används för att göra gränssnitt enklare, mer läsbart och därmed också mer praktiskt vilket gör att affordansen tydligt framgår. Då ikoner används i rätt sammanhang kommunicerar de sin innebörd mer effektivt än textelement (Harley, 2014). Detta gör att vi kan använda mer skärmyta för att visa kartor och annan information, vilket därmed ökar synligheten i appen. För våra sökfält har vi valt att använda oss av en ikon med ett förstöringsglas. Ikonen har lagts precis intill

fältet där användaren skriver för att förbättra mappningen i appen och för att tydliggöra att ikonerna i fråga hänger ihop med sökfältet. Då användaren trycker på en ikon, ändrar den under en kort stund färg från vit till en mörkare grå för att indikera att tryckningen registrerats, därefter utförs kommandot som ikonerna är förknippade med omedelbart. Detta förbättrar ikonernas återkoppling.

## 4. Projektmetod

### 4.1. Tidsplan

Det allra första som utfördes innan programmeringen kom igång var att göra en detaljerad och realistisk tidsplan. Det första steget var att få tag på kartor över Tappan för att sedan kunna göra egna kartor i Adobe Illustrator som stämmer överens med originalen. Något som var beräknat att ta några dagar och därefter kunde programmeringen sätta igång. Efter en vecka var det beräknat att startsidan skulle vara färdig samt att en grund för våningsplanet TP3 skulle vara implementerad. Ytterligare en vecka senare skulle TP3 vara klar tillsammans med sökfunktionen och navigeringen till de olika planen. Under loppet av ytterligare två veckor skulle TP4 och TP5 bli färdiga som därefter skulle följas av finjusteringar och lösning av eventuella problem. Sista veckan innebar till slut rapportskrivning och då skulle det finnas god marginal till att fixa saker som skulle kunna gå snett under arbetsprocessen.

Det som beräknades ta mest tid var att få sökfunktionen och navigeringen att fungera. Att få ett fungerande TP3 förväntades också kräva en del arbete och därför gavs mest tid till att få ett plan så bra som möjligt för att sen enkelt kunna återanvända kod till de andra två planen. För att inte behöva hamna i ett stressigt läge vid slutet av projektet försökte gruppen tänka så realistiskt som möjligt med hur mycket tid som skulle krävas för de olika delarna. I och med att *React-Native* var ett helt främmande språk, insåg vi att mycket av tiden skulle gå till att läsa sig fram till lösningar och försöka förstå de inbyggda funktionerna som språket har.

### 4.2. Arbetsfördelning

På grund av en del sjukdomar så har arbetsfördelningen varierat mellan gruppens medlemmar. Tillsammans har vi lagt cirka 300 timmar på projektet där Vera har arbetat 24% av den totala tiden, Marcus har stått för 22%, Alexander har snittat på 20% medan både

Christophe och Carl har täckt 17%. Alla har bidragit till både programmeringen och rapportskrivningen.

Under projektets gång delades olika uppgifter ut till medlemmarna i gruppen. Oftast arbetade gruppmedlemmarna i par. I början av arbetet var de olika arbetsuppgifterna oberoende av varandra men ju längre vi kom, desto mer inflätade var de olika sidorna i varandra.

Parprogrammeringen blev mindre påtaglig allt eftersom de funktioner som gick att programmera individuellt blev klara och det behövdes ett samarbete mellan alla för att få applikationen att gå ihop. Detta gjorde att arbetsfördelningen blev väldigt dynamisk och istället hjälpte alla i gruppen till där det behövdes; om en person körde fast på en funktion så hjälpte de andra till.

## 5. Diskussion

Redan från början var vi medvetna om vilka mål vi hade och vilket betyg vi satsade på. Disciplinen har alltid varit hög och satte vi tider för när man skulle vara på campus och jobba med projektet så har alla nästan alltid varit i tid, och var man inte det så fick man bjuda på fika till resten av gruppen. En annan styrka har varit att vi har samarbetat bra och alla har respekterat varandra och vi har fört en bra dialog. Indelningen av ansvarsområden inom gruppen så som kontaktansvarig och projektledare skedde naturligt och alla har gjort sin del av det ansvarsområdet.

Ett problem som uppstod direkt när vi skulle börja programmera var att vi märkte att det inte gick att återanvända någon *React* som använts vid laborationerna, utan vi var tvungna att börja om på nytt med *React-Native* istället. Det kanske inte låter som om språken skiljer sig mycket men *React* används till hemsidor och *React-Native* används till mobilapplikationer. Även om det var ett jobbigt moment i början så känner vi alla att det har varit väldigt lärorikt och det har gett oss nya kunskaper. Vi har googlat oss fram till det mesta och laborationsassistenterna har varit till stor hjälp även om de inte kunde just *React-Native*.

Det som tog längst tid att implementera var zoom-funktionen på kartorna. När vi skulle hitta en funktion för att kunna zooma och dra runt bilden provade vi många olika komponenter som hittats online utan att få någon att fungera på det sättet vi ville.

Efter många timmars prövning av olika komponenter hittade vi till slut *ScrollView* som fungerade bäst. Däremot uppstod ytterligare ett problem när vi väl fått zoomen att fungera. Endast halva kartan syntes och vägrade hamna centrerat på skärmen. Flera timmar gick åt att sitta och läsa om *ScrollView* och koden testades att skrivas om flera gånger utan förbättringar. Tack vare Emils hjälp lyckades vi tillslut få kartan att bli centrerad på skärmen, men fick istället ytterligare ett problem. När man kom in på de olika sidorna med planen så var alla kartor lite in-zoomade vilket gjorde att man inte kunde se kanterna. Efter ytterligare många timmars felsökning och test av olika lösningar hittade vi tillslut en som fungerade. Genom att hämta skärmens dimensioner på telefonen via en inbyggd funktion i *React-Native* så kunde vi anpassa storleken på containern som kartan låg i så att den inte överskred skärmens storlek. Affordansen hos zoom-funktionen framgår inte särskilt tydligt i applikation, detta skulle behöva förbättras genom någon slags signifi.

Ytterligare problem som uppstått har varit kring bilderna. Oavsett hur hög upplösning de har exporterats i så blir kvalitén på dem dåliga när man väl lägger in dem i appen. Utöver detta så tar det ett tag för bilderna på varje sida att laddas. All text och ikoner kommer upp så fort användaren navigerar till vald sida men det dröjer innan själva bilden kommer upp (det vill säga kartorna, LiU-loggan och bilden på info-sidan). Vi sökte runt efter lösningar på dessa problem men lyckades inte hitta något användbart innan projektets slut. Om vi hade haft mer tid på oss så hade dessa problem säkert gått att lösa men för närvarande är det ett problem.

Eftersom gruppens medlemmar har olika telefoner så försökte vi till en början designa gränssnittet så att det skulle passa alla telefoner, men insåg rätt snabbt att det var lättare sagt än gjort. Därför togs beslutet att göra appen anpassad efter iOS-telefoner eftersom majoriteten av medlemmarna i gruppen har det märket på sina telefoner. På grund av att textrutorna med salsnamnen har placerats med hjälp av koordinater så hamnar de bara rätt på de iOS-telefoner som har bildförhållandet 16:9 vilket innebär alla iPhones som kom ut efter 2012. Hade det funnit mer tid så hade vi även här kunnat förbättra gränssnittet, så att det är anpassningsbart för alla telefoner och storlekar.

Vid projektets slutfas gjorde vi några användarstudier där vi bad klasskamrater testa applikationen och ge feedback. De nackdelar som vi fick höra var att studentfiktet saknades,

samt att det hade varit bra om det gick att söka på endast salens nummer istället för att behöva skriva "TP" framför. Efter denna feedback gjorde vi en symbol med en kaffekopp för att markera var på kartan studentfikat ligger och la till en förklaring för symbolen på informationssidan. När det kommer till att kunna söka på endast salsnummer så kände vi att tiden inte räckte till för att ändra om sökfunktionen eftersom det var en utav de delar som hade krävt mycket tid. Om användartestet hade gjorts tidigare istället för i slutet så hade det förmodligen gått att lösa. De fördelar som vi fick höra var att ikonerna är snygga och har ett tydligt syfte, även att det är enkelt att förstå användningen av appen och att de olika sidornas mappning är väl genomtänkt.

Angående den ursprungliga tidsplanen så har arbetet inte gått exakt som planerat. Sökfunktionen och navigeringen blev färdigt i tid förutom lite slutgiltiga finjusteringar som var kvar. Det som tog mycket längre tid än planerat var sidorna med planen. Mycket av förseningen berodde på problemen som uppstod i samband med zoom-funktionen av kartorna. Utöver detta så tog undersökningen av vilka koordinater som varje textruta med salsnamnen skulle ha, oerhört mycket tid. För att hitta rätt x- och y-koordinat så fick vi manuellt sitta och knappa in olika siffror på varje textruta för att se till så att de hamnade på exakt rätt ställe. På grund av att man hela tiden hittar saker som kan förbättras i applikationen så började vi med rapporten fyra dagar senare än planerat. Trots detta så känner vi ändå att det inte behövde stressas in i det sista. Vi blev klara med allt och är nöjda med resultatet.

## **6. Slutsats**

Det visade sig i ett tidigt skede av utvecklingen av applikationen att det kanske inte var så svårt som befarats innan arbetet började. Det är mycket enkelt att komma igång med utvecklingen av en mobilapplikation. Stora mängder dokumentation av grundläggande funktioner och principer finns tillgängliga, tillsammans med omfattande vägledning online i både text- och videoform. Emellertid blir det svårare och mindre tydligt vad som ska göras ju längre utvecklingen fortskrider då mer avancerade och inte lika vedertagna koncept implementeras i applikationen. Rent allmänt är problem lättare att lösa ju fler som stött på dem tidigare, vilket gör att det är svårare att få svar på frågor som är mer specifika för applikationen som utvecklas.

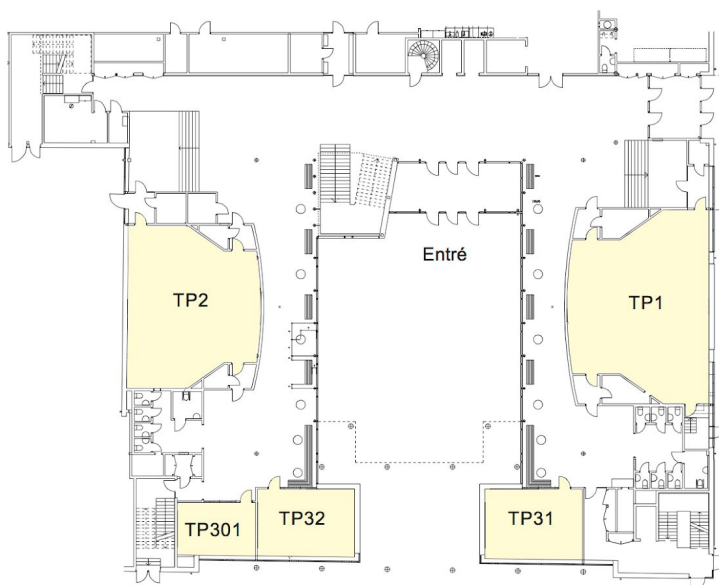
Att lära sig Javascript visade sig vara en utmaning. Språket är inte särskilt likt något gruppen använt tidigare, förutom den mest grundläggande programmerings-syntaxen. Som komplement till de kunskaper gruppen fått genom föreläsningarna i kursen användes också resurser på internet samt vägledningen på handledarpassen.

Planlösning över Täppans våningar fanns tillgängligt på Linköpings Universitets hemsida, och användes som bekant i utvecklingen av applikationen.

Dynamisk skalning till olika skärmar implementerades tyvärr inte i applikationen. Det stod klart hur funktionen skulle implementeras ganska sent i utvecklingen, och en avsevärd del av koden som skrivits hittills skulle behövt skrivas om. Detta ansågs inte vara värt det, då det skulle ta mycket av den kvarvarande utvecklingstiden och sänka kvaliteten av den slutgiltiga applikationen och dess funktioner. Istället för att få applikationen att fungera lite sämre på många enheter valde gruppen att få den att fungera mycket bra på en enhet.

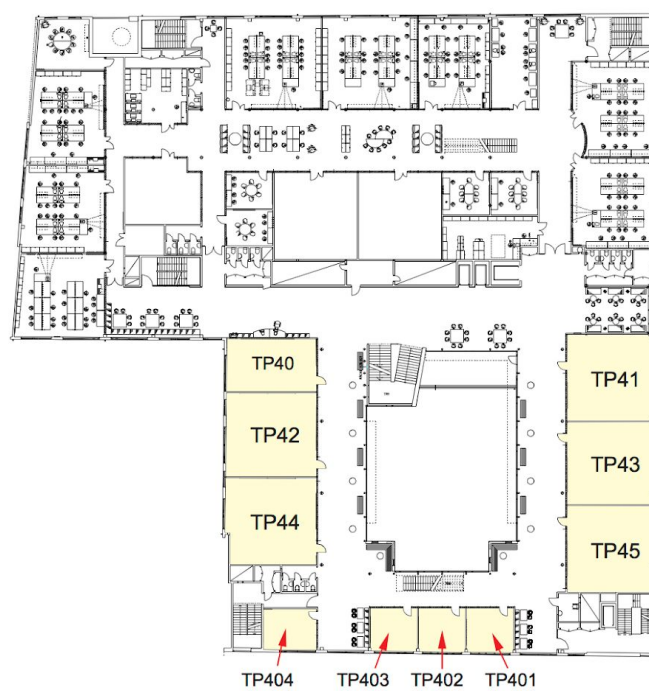
## 7. Bilagor

**Täppan plan 3**



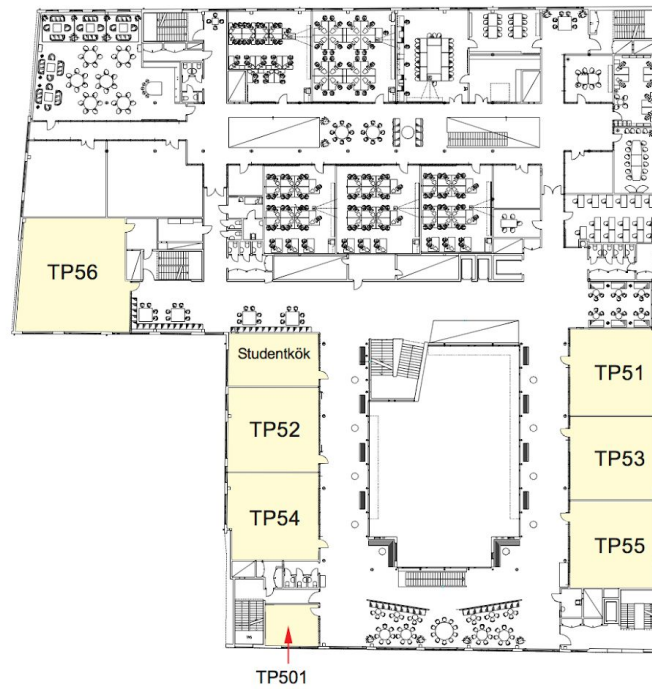
Bilaga 1. Ursprunglig karta över TP3

**Täppan plan 4**



Bilaga 2. Ursprunglig karta över TP4

### Täppan plan 5



Bilaga 3. Ursprunglig karta över TP5

## 8. Källor

Harley, Aurora (2014), *Icon Usability*

**Tillgänglig:** <https://www.nngroup.com/articles/icon-usability/>