

BACHELOR'S THESIS 2021

Ett koncept för framtida övervakningssystem för åtkomstkontrollerade miljöer

Ferit Bölezek, Dominik Gashi

Elektroteknik
Datateknik

ISSN 1651-2197

LU-CS/HBG-EX: 2021-02

DEPARTMENT OF COMPUTER SCIENCE

LTH | LUND UNIVERSITY



Ett koncept för framtida övervakningssystem för åtkomstkontrollerade miljöer

LU-CS/HBG-EX 2021-02



LUNDS UNIVERSITET
Campus Helsingborg

LTH Ingenjörshögskolan vid Campus Helsingborg
Institutionen för datavetenskap

Examensarbete:
Ferit Bölezek
Dominik Gashi

© Copyright Ferit Bölezek, Dominik Gashi

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige
Lunds universitet
Lund 2021

Sammanfattning

Detta examensarbetets syfte var att i samarbete med Axis Communications utveckla en applikation för att övervaka flera ut- och ingångar samt hantera åtkomstkontroll. Examensarbetet har i huvudsak resulterat i en interaktiv prototyp som erbjuder funktionaliteten beskriven i föregående mening. För att uppnå detta resultat har omfattande arbete genomförts med informationsinhämtning samt utveckling.

Projektmodellen Scrum användes för att strukturera samt följa examensarbetets faser som bestod av:

1. Elicitering och Informationsinsamling
2. UI/UX-prototyp
3. Utveckling av en interaktiv prototyp
4. Utvärdering av den interaktiva prototypen.
5. Avslutning

Examensarbetets tidigare stadier bestod mestadels av elicitering och inläring, där olika tekniker utnyttjades för att säkerställa en god grund för prototyp-utvecklingen i senare stadier. Exempel på dessa tekniker är intervjuer av anställda på Axis Communications, läsning av litteratur, informations-uppsökning om liknande existerande lösningar samt inläring av relevant information genom videoklipp online.

Efter första fasen började utvecklingen av UI/UX-prototypen baserad på informationen som hämtats in samt önskemål från Axis Communications. UI/UX-prototypen utgjorde en grund för senare utveckling av den interaktiva prototypen under tredje fasen.

Den interaktiva prototypen som blev examensarbetets slutgiltiga resultat, består i huvudsak av en interaktiv planritning där operatören kan reglera dörrarnas tillstånd (låsa, låsa upp etc.), se var personer befinner sig baserat på kortet som användes på kortläsare för att begära åtkomst, samt se historik över passage för samtliga dörrar på planritningen. Utöver detta kan operatören även se status och information om komponenter (exempelvis dörrar, kortläsare, kameror etc.).

Under examensarbetets slutskede låg stort fokus på utvärdering av den interaktiva prototypen, där informella tester utfördes på prototypen för att upptäcka eventuella fel.

Nyckelord: Åtkomstkontroll,
Övervakningssystem,
Prototyp,
Interaktiv planritning,
React,
SVG,
Websocket

Abstract

The purpose of this thesis was to develop an application in collaboration with Axis Communications to monitor several outputs and inputs, as well as manage access control. The degree project has mainly resulted in an interactive prototype that offers the functionality described in the previous sentence. To achieve this result, extensive work has been carried out with information gathering and development.

The Scrum project model was used to structure and follow the degree project phases which consisted of:

1. Elicitation and Information Collection
2. UI/UX-prototype
3. Development of an interactive prototype
4. Evaluation of the interactive prototype.
5. Completion

The earlier stages of the degree project consisted mostly of elicitation and learning, where different techniques were used to ensure a good basis for prototype development in later stages. Examples of these techniques are interviews with employees at Axis Communications, reading literature, seeking information about similar, existing solutions and learning relevant information through online videos.

After the first phase, the development of the UI/UX-prototype began based on the information gathered and requests from Axis Communications. The UI/UX-prototype formed the basis for later development of the interactive prototype during the third phase.

The interactive prototype was the final result of the degree project, it consists mainly of an interactive floor plan where the operator can regulate the conditions of the doors (lock, unlock, etc.), see where people are based on the card used on the card reader to request access, as well as see the history of passage for all doors on the floor plan. In addition to this, the operator can also see the status and information about components (e.g. doors, card readers, cameras, etc.).

During the final phase of the degree project, there was a strong focus on evaluation of the interactive prototype, where informal tests were carried out to detect any errors.

Keywords: Access control,
Surveillance System,
Prototype,
Interactive floorplan,
React,
SVG,
Websocket

Förord

Vi, Ferit Bölezek och Dominik Gashi, skulle vilja tacka Axis Communications för ett gott samarbete och för ett trevligt bemötande. Axis Communications har agerat som en enastående stödpelare åt oss under examensarbetets gång. Vi känner att vi fått det stöd och den hjälp vi önskade för att fullborda detta examensarbete. Vi är även glada över möjligheten att ta del av arbetslivet som högskoleingenjör inom datateknik, även om det mestadels varit på distans på grund av Covid 19.

Vi vill även tacka handledaren och examinatoren på Lunds Universitet för konstant återkoppling, hjälp och stöd genom hela examensarbetet. Vi vill även tacka för den konstruktiva kritiken vid varje handledningstillfälle som bidragit till att examensrapporten ständigt förbättras för varje version.

Vi hoppas att du som tar del av denna rapport tycker den är informativ och intresseväckande.

Innehållsförteckning

Inledning	2
Bakgrund	2
<i>Exempel på dagens åtkomstkontrollerade miljöer</i>	3
<i>Möjlig lösning</i>	3
<i>Företagets efterfrågan</i>	4
<i>Syfte</i>	6
<i>Målformulering</i>	6
<i>Problemformulering</i>	6
<i>Motivering av examensarbetet</i>	6
<i>Avgränsningar</i>	7
Teknisk bakgrund	8
<i>React</i>	8
<i>Material Design</i>	11
<i>Scalable Vector Graphics (SVG)</i>	12
<i>Node.js</i>	13
<i>Websocket</i>	13
<i>Material-UI</i>	14
<i>Trello</i>	14
<i>AXIS A1601 Network Door Controller</i>	14
Metod	16
<i>Arbetsättet</i>	17
<i>Projektmodellen</i>	21
<i>Kommunikation</i>	23
<i>Källkritik</i>	24
Analys	30
<i>Fas 1: Elicitering och informationsinsamling</i>	30
<i>Fas 2: UI/UX-prototyp</i>	31
<i>Fas 3: Utveckling av en interaktiv prototyp</i>	32
<i>Fas 4: Utvärdering av systemet</i>	40
<i>Fas 5: Avslutning</i>	41
<i>Analys av arbetet med Scrum</i>	41
Resultat	44
Slutsats	50
<i>Problemformulering</i>	50
<i>Reflektion över etiska aspekter</i>	51
<i>Framtida utvecklingsmöjligheter</i>	51

Terminologi	54
Källförteckning	56
Appendix	58
<i>UI/UX Prototyp</i>	58
<i>Intervju med Axis UX-designers</i>	67
<i>Scrumboard</i>	69

1. Inledning

Detta är ett examensarbete som gick ut på att undersöka nuvarande lösningar som finns för att övervaka åtkomstkontrollerade miljöer. Ett exempel på en åtkomstkontrollerad miljö är Lunds Universitet Campus Helsingborg då det krävs passerkort för att få åtkomst till en del av universitetets lokaler. Utifrån undersökningarna som gjordes skulle en exekverbar prototyp utvecklas. Prototypen skulle inkludera funktionalitet som innebar övervakning av flertal dörrar samt hanteringen av dessa, hantering innebar bland annat möjligheten att låsa och låsa upp dörrarna.

I detta kapitel presenteras bakgrunden, syftet och målformuleringar för examensarbetet. Utöver det beskrivs motiveringen av examensarbetet, vilka problem som undersöks samt vilka avgränsningar som finns.

1.1 Bakgrund

Åtkomstkontroll handlar om förhindrandet av åtkomst till information eller till en plats för obehöriga. För att förhindra eller tillåta åtkomsten till en specifik plats såsom en byggnad behöver man hålla reda på vilka som är behöriga för tillgång till platsen [1]. Ett exempel på detta kan vara passerkort som tillåter inträde till utvalda lokaler beroende på vem passerkortet tillhör.

Traditionella lösningar som kort/kortläsare brukar användas i samband med åtkomstkontroll, dessa kan exempelvis användas för att bevilja eller neka åtkomst till en byggnad. Byggnader som använder dessa lösningar kan även ha ett övervakningssystem som ger en konstant överblick över åtkomstkontrollerade miljöer [1].

Problemet inom området är att dessa övervakningssystem kan vara gamla, svåra att förstå samt svåra att använda. Därför krävs det en effektivisering av dessa gamla system, även om de anses fylla sitt syfte. Ett effektivare system innebär att de som använder sig av systemet kan använda det snabbare, med färre och mindre komplexa steg för att utföra liknande handlingar, till exempel neka åtkomst eller övervaka flera ingångar samtidigt. Detta bidrar både till en större användarvänlighet i form av lättare tillgång och till att den nya, mer effektiva, teknologin som finns på marknaden utnyttjas.

Idag ser förutsättningarna annorlunda ut i jämförelse med tiden då de allra första åtkomstkontrollerade miljöerna togs fram. Eftersom ny teknologi blir mer lättillgängligt med åren, inte bara för företag utan även för privatpersoner, vill b.l.a. Axis finna ett sätt att integrera den nya teknologin med den äldre, redan existerande, teknologin. Syftet med denna form av integrering mellan ny och gammal teknologi, handlar inte bara om effektivisering som tidigare nämnts, utan även återvinning av komponenter vilket innebär besparingar både

ekonomiskt och miljömässigt, något som skulle vara svårare att uppnå om återvinningen inte görs och systemet byggs på nytt från grunden.

1.1.1 Exempel på dagens åtkomstkontrollerade miljöer

Idag är det vanligt med hårdvara som kortläsare, knappsats, passerkort och kameror för att skapa en åtkomstkontrollerad miljö. Ett möjligt användningsscenario för detta är:

Användningsfall: Individ försöker ta sig in i en åtkomstkontrollerad byggnad

Förutsättning: Individen har ett passerkort

1. Individen närmar sig knappsatsen/kortläsaren.
2. Individen tar fram sitt passerkort och placerar den på kortläsaren och knappar in sin kod.
3. Kortläsaren bekräftar att individen har rätt att ta sig in i byggnaden.
4. Individen öppnar dörren och tar sig in i byggnaden.

Alternativa fall:

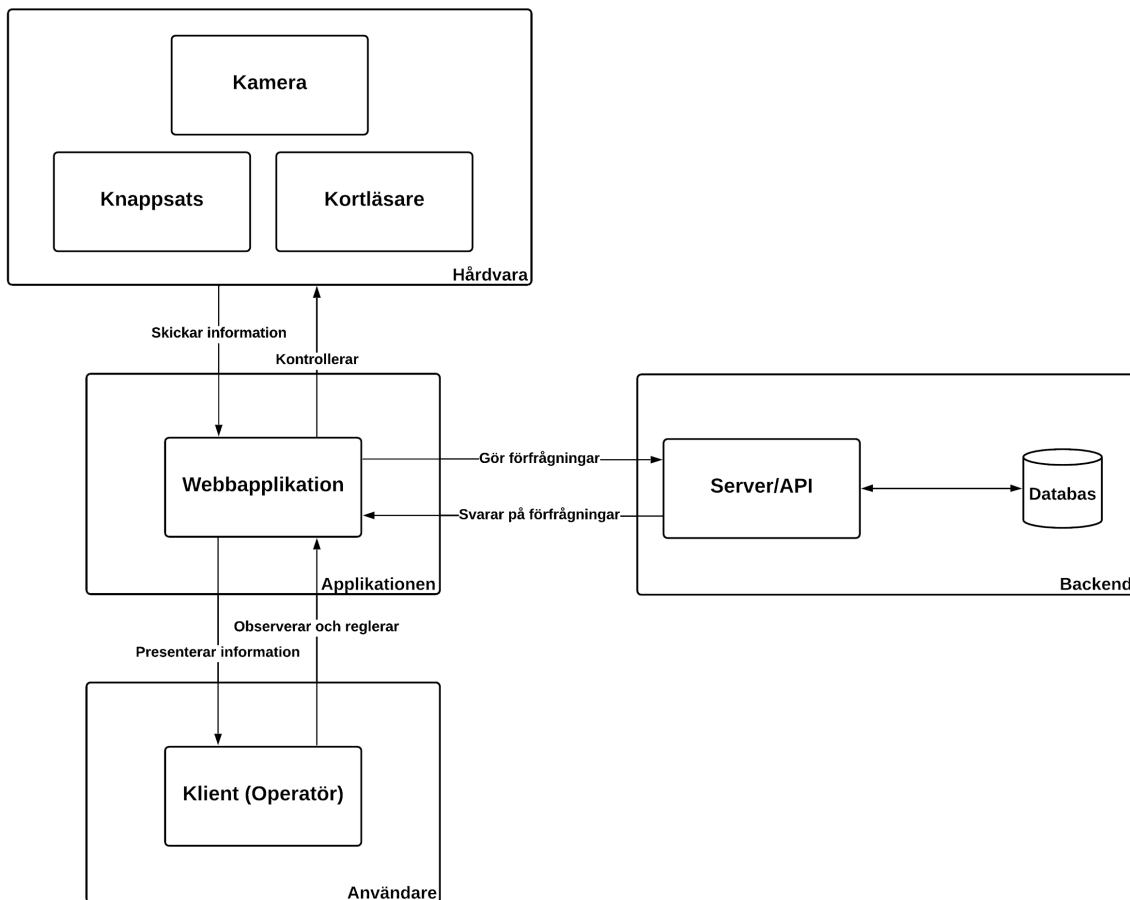
Steg 2 och 3. Individen anger fel kod vilket leder till att åtkomst till byggnaden nekas.

Steg 3. Kortläsaren känner inte igen passerkortet och därav nekar åtkomst till byggnaden.

Steg 3. Individen har inte rätt till att komma in i byggnaden, exempelvis individen tillhör inte gruppen som har åtkomst till byggnaden men däremot tillhör gruppen som har åtkomst till en annan byggnad.

1.1.2 Möjlig lösning

En möjlig lösning är att använda sig av ett utvecklingsverktyg för att bygga en webbapplikation som kan ge en överblick över den åtkomstkontrollerade miljön genom en webbläsare (se kapitel 6). Detta innebär att man ska kunna ha en överblick över både den åtkomstkontrollerade miljön som helhet och även enskilda rum och dörrar.



Figur 1. Kontextdiagram för det tänkta systemet.

För att uppnå detta kan designstandarder inom UI/UX användas för att skapa en webbapplikation som är enkel att förstå samt enkel att använda. Se figur 1 för ett kontextdiagram av det tänkta systemet.

1.1.3 Företagets efterfrågan

Examensarbetet görs i samarbete med Axis Communications som är ett svenskt företag som specialiserar sig inom fysisk säkerhet och videoövervakning [2]. Som tidigare nämnts arbetar Axis ständigt med att effektivisera sina system för att bland annat göra besparingar ekonomiskt, miljömässigt samt tidsmässigt.

Axis efterfrågar ett system som möjliggör övervakningen av en åtkomstkontrollerad miljö där man exempelvis kan övervaka hela byggnader, enskilda rum eller enskilda dörrar. Systemet ska kontrolleras av en operatör som på ett översiktligt sätt kan se alla individer som besöker ett rum genom att använda passerkort för att få åtkomst. Operatören ska också ges möjlighet

att reglera åtkomst på individnivå genom att få åtkomst till passerkortets information genom systemet.

Systemet skall också ge operatören möjligheten att låsa och låsa upp dörrar. Genom användning av ansiktsgenkänning ska systemet hjälpa operatören att försäkra att individen som tillträder det åtkomstkontrollerade miljön är behörig att befinna sig i byggnaden. I teorin skulle detta åstadkommas genom en jämförelse av passerkortets bild på ägarens ansikte med ansiktet på personen som tillträder byggnaden.

Detta system skulle innebära en utökning av Axis nuvarande system. Idag har operatören möjlighet till övervakning av en passage åt gången vilket innebär en låg effektivitet och hög tidsåtgång.

Ett exempel på hur åtkomst ges till en person med passerkort genom att utföra olika handlingar visas i diagrammet nedan (Diagram 1). Handlingarna kan bl.a. innebära att kortet används, lösenordet matas in eller att operatören ger åtkomst till personen med passerkort manuellt. Beroende på vad handlingarna har för resultat, exempelvis om rätt eller fel kod matas in, kan personen med passerkort få åtkomst eller nekad åtkomst.

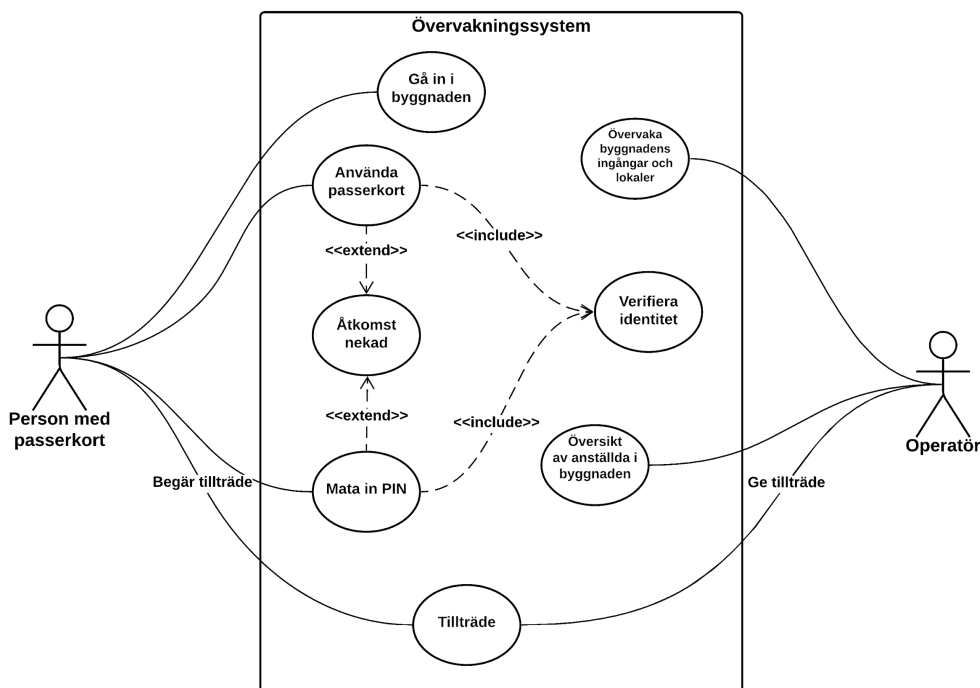


Diagram 1. Användarfallsdiagram av systemet. (UML-notation)

1.2 Syfte

Syftet med examensarbetet är att undersöka existerande lösningar och utifrån detta ska en prototyp för ett effektivare och i större utsträckning mer tidsbesparande övervakningssystem skapas som framtida privatpersoner, andra företag och organisationer eller kunder hos Axis skulle kunna använda för att övervaka åtkomstkontrollerade miljöer.

1.3 Målformulering

Ett av målen för detta examensarbete är att göra en analys som kommer vara baserad på en UI/UX-prototyp och simuleringar (se kapitel 5). Även undersökningar av hur system kan tänkas hantera åtkomstkontroll och övervaka åtkomstkontrollerade miljöer kommer att göras.

Slutligen ska en interaktiv prototyp för ett övervakningssystem baserat på undersökningarna, utvecklas, testas och anses vara enkel att använda samt enkel att förstå enligt efterfrågan, behov och tekniska kunskaper från Axis kunder som tidigare specificerats.

Ett annat mål för detta examensarbete är att analysera nuvarande system, identifiera deras styrkor och svagheter, samla in erfarenheter från användarna av det nuvarande systemet och utifrån det utöka Axis nuvarande system.

1.4 Problemformulering

Detta examensarbete kommer att besvara följande frågor:

1. Vilka nuvarande lösningar finns för åtkomstkontroll?
2. Vilka åtgärder kan införas för att bidra till effektivisering av en redan implementerad övervakning av en åtkomstkontrollerad miljö?
3. Vad krävs för att den föreslagna lösningen skall komma att användas i framtiden?

Motivering av examensarbetet

Genom att undersöka nuvarande övervakningssystem ska deras svagheter identifieras. Därefter ska en prototyp skapas som ska gå att utöka och underhållas. Detta möjliggörs genom användandet av ett välkänt programmeringsspråk samt genom att följa välkända UI/UX standarder. Detta system ska i slutändan gynna användaren av övervakningssystemet samt framtida utvecklare som potentiellt ska underhålla samt utöka källkoden. Som tidigare nämnt gynnar detta Axis då det innebär en effektivisering, återvinning och modernisering av deras nuvarande system.

Valet av ämne gjordes dels då intresset för säkerhet både mjukvarumässigt samt hårdvarumässigt finns. Arbetets ämne är koncentrerat kring säkerhet men knyter väl an till

andra områden såsom mjukvaruutveckling samt hårdvaruutveckling. Valet av ämne var redan bestämt i ansökan som gjordes till Axis.

1.5 Avgränsningar

Prototypen kommer endast köras och testas på webbläsaren Google Chrome. Utvecklingen kommer att ske på hårdvara som Axis har tillgängligt för examensarbetarna. Detta inkluderar bland annat en laptop, en AXIS A1601 Network Door Controller [37] och en kortläsare. Laptopsen som tillhandahålls av Axis och som kommer att användas för utvecklingen kör operativsystemet Linux, detta innebär att prototypen kommer endast testas på en Linux-miljö. Dessa avgränsningarna har gjorts för att underlätta testandet för den begränsade tidsram som examensarbetarna har för detta arbetet. Detta innebär även att mer tid kan avsättas för utvecklingen som är en viktig del av detta examensarbetets syfte, och mindre tid på tester som inte är relevant för detta examensarbetets syfte.

2. Teknisk bakgrund

Utvecklandet av det tänkta systemet kommer att bland annat ske med hjälp av JavaScript-biblioteket React, plattformen node.js och designstandarden Material Design. Detta kapitel kommer att introducera de ovan nämnda teknikerna samt ge en introduktion för vektorgrafikformatet SVG som kommer att ha en central roll i systemet.

2.1 React

React är ett JavaScript-bibliotek utvecklat av Facebook. Biblioteket har öppen källkod och används för att skapa interaktiva användargränssnitt för webbsidor [3].

Frontend-utveckling i React är komponentbaserad där varje komponent kan representera en bit användargränssnitt på skärmen, detta kan exempelvis vara en knapp, ett textfält, en behållare för andra UI-komponenter mm. Varje komponent har sitt eget tillstånd, är anpassningsbar och återanvändbar [3].

För att skapa komponenter i React används vanligtvis JSX som är en utökning av JavaScript-syntaxen som gör det möjligt att skapa användargränssnitt med en XML/HTML-liknande syntax [4]. I Figur 2 visas hur JSX-kod kan struktureras i React, koden körs sedan genom kompilatorn "Babel" som omvandlar koden till det som visas i Figur 3 [4][5]. Eftersom JSX-koden i slutändan omvandlas till ren JavaScript-kod så är det möjligt att använda React utan Babel eller JSX, detta innebär dock att man förlorar läsbarheten som JSX erbjuder [4].

```
<Grid container spacing={2} direction='column'>
  <Grid item>
    <NotificationList title="Critical" />
  </Grid>
  <Grid item>
    <NotificationList title="Warnings" />
  </Grid>
  <Grid item>
    <NotificationList title="Other" />
  </Grid>
</Grid>
```

Figur 2. Exempel på hur JSX kod struktureras i React.

```

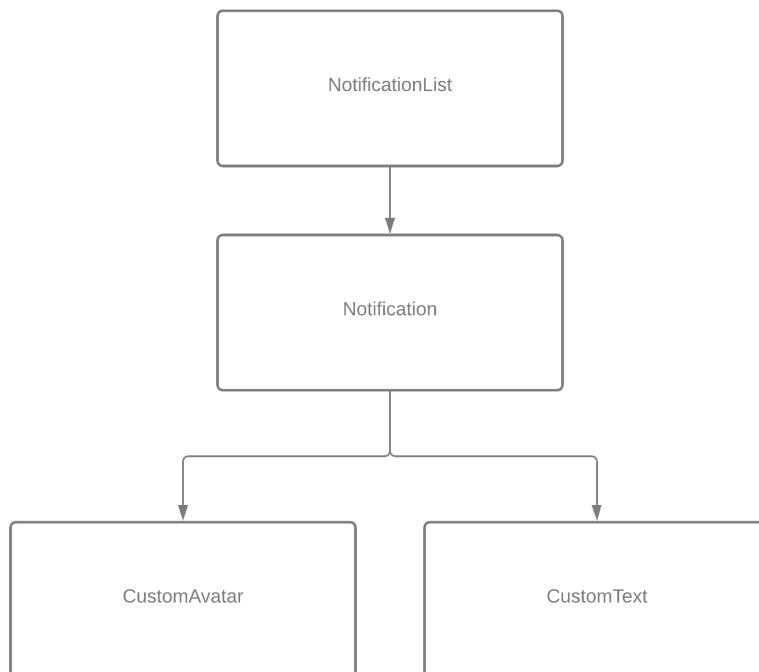
React.createElement(Grid, {
  container: true,
  spacing: 2,
  direction: "column"
}), React.createElement(Grid, {
  item: true
}), React.createElement(NotificationList, {
  title: "Critical"
})), React.createElement(Grid, {
  item: true
}), React.createElement(NotificationList, {
  title: "Warnings"
})), React.createElement(Grid, {
  item: true
}), React.createElement(NotificationList, {
  title: "Other"
})))));

```

Figur 3. Exempel på hur koden kan struktureras utan JSX.

Reactkomponenternas tillstånd kan sparas och användas genom användningen av state-systemet. Detta gör det möjligt att veta och uppdatera komponentens tillstånd i React utan att behöva hämta ut det från Document Object Model (DOM).[3] Ett exempel på detta är en komponent som representerar ett textfält i webbläsaren. Genom att bevara textfältets text i state-systemet kan man vid alla tidpunkter veta vad som befinner sig i textfältet, utan användning av state-systemet hade man varit tvungen att hämta ut denna information från DOM.[6]

Eftersom Reactkomponenter kan vara behållare för andra Reactkomponenter så är det vanligt med trädstrukturer för komponenterna. Figur 4 visar ett exempel på hur en trädstruktur av komponenter i React kan se ut, i detta fallet är "NotificationList" en komponent som kommer att representera en lista och "Notification" är en komponent som kommer att representera ett element i listan. Varje Notification kommer att innehålla komponenterna "CustomAvatar" och "CustomText" som kommer att vara ansvariga för att visa text och bild i Notification.



Figur 4. Exempel på ett trädstruktur där NotificationList och Notification är behållare för andra komponenter.

I dessa trädstrukturer är det vanligt att man vill passera data från en komponent till en annan. För data som flödar nedåt (från förälder till barn) kan props-systemet användas. Datan passeras ner till barnet som parameter i en JSX-tag, vilken kan hämtas ut av barnet via props-systemet.[7]

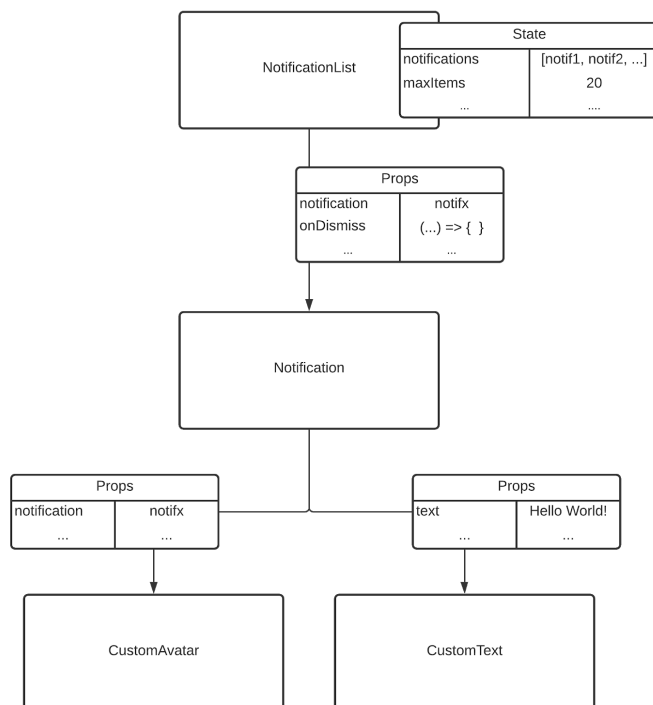
I Figur 5 visas hur data kan skickas från förälder till barn. En komponent NotificationList initieras och får datan "Critical". NotificationList kan sedan hämta ut datan genom att läsa title-variabeln som har fått data i props-systemet. I detta exemplet har "title" valts som variabelnamn, namnet på variabeln måste inte sättas till detta utan programmeraren kan själv välja ett passande variabelnamn.

```
<NotificationList title="Critical" />
```

Figur 5. Exempel på passering av data från förälder till barn.

Det är även möjligt att kommunicera uppåt i trädstrukturen genom att passera en funktion från föräldern ner till barnet. Barnet kan då anropa funktionen i föräldern för att skicka någon sorts information till föräldern. Figur 6 visar ett exempel på samspelet mellan komponenter, state-systemet och props-systemet, NotificationList använder sig av State för att hålla reda på alla notifikationer som finns i listan samt en variabel som anger max antal notifikationer som får finnas i listan. NotificationList samt Notification använder sig av props-systemet för att kommunicera med sina barn. NotificationList passerar även onDismiss som en funktion via

props-systemet till Notification, vilket gör det möjligt för Notification att anropa denna funktionen vid något tillfälle och passera information till föräldern.[8]



Figur 6. Samspelet mellan komponenter, state-systemet och props-systemet.

Genom att använda JSX, komponenter, state-systemet och props-systemet kan man konstruera interaktiva användargränssnitt med React för webben.

React har använts i detta examensarbete för att utveckla användargränssnittet för den interaktiva prototypen. En fördel med React är att det är ett välkänt och moget JavaScript-bibliotek. Detta innebär att det är enkelt att hitta information på nätet angående biblioteket och hitta information online vid felsökning, informationskällor som använts beskrivs i kapitel 3. Nackdelen med React är att den kan ha en brant inlärningskurva, utöver inläringen som krävs för bibliotekets funktionaliteter så krävs även tidigare kunskaper inom HTML, JavaScript och CSS.

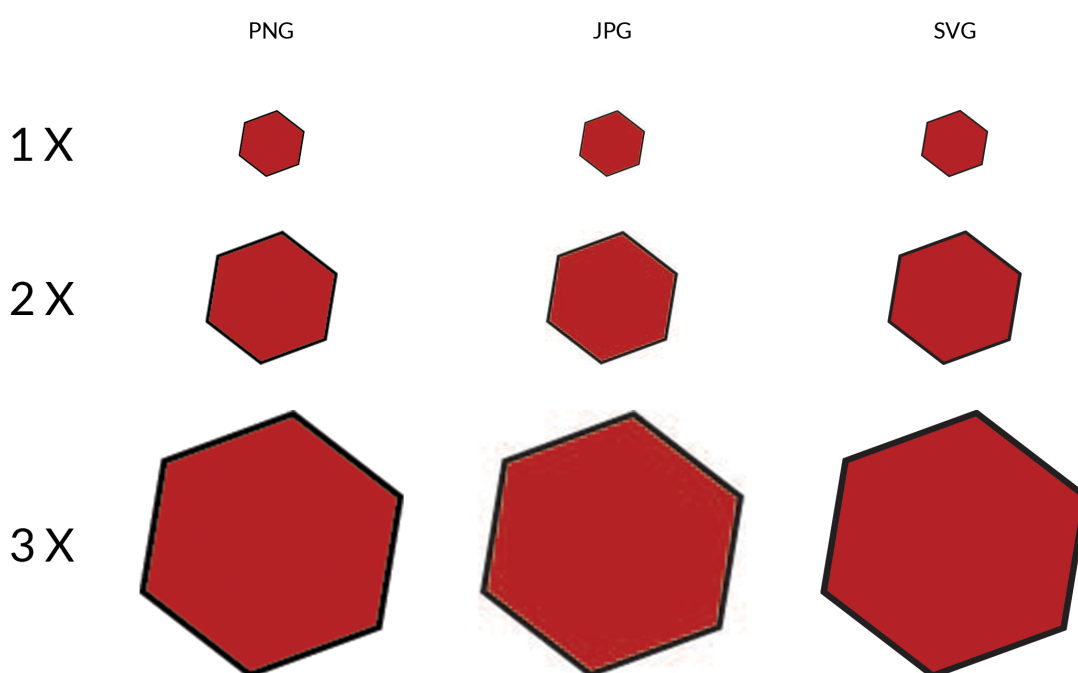
2.2 Material Design

Material Design är en designstandard utvecklat av Google. Material Design skapar en standard för hur användargränssnitt borde se ut på bland annat iOS, Android och webb [14]. En fördel med användningen av Material Design är att den bidrar till en enhetlighet genom att delar av designen återanvänds i flera sammanhang i applikationen. Däremot kan

användningen av Material Design innebära ett viss hinder för examensarbetarna i kreativitet och variation, då användargränssnittet har fördefinierade riktlinjer.

2.3 Scalable Vector Graphics (SVG)

SVG är ett XML-baserad vektorgrafik-format. Bilder kan skapas och editeras i SVG format genom XML-kod. Formatet skiljer sig från andra bildformater som JPG, PNG etc. Skillnaderna är bland annat att SVG kan skalas upp utan att tappa bildkvalité (se Figur 7), bilderna kan animeras och att formatet gör det möjligt att skapa interaktivitet mellan användare och bilder [15].



Figur 7. Skillnaden på bildkvalité när en bild förstoras. Jämförelsen görs mellan tre populära val på bildformat på webben (PNG, JPG och SVG).

Interaktivitet för SVG är möjligt eftersom formatet har ett DOM (Document Object Model) och events (som onclick, onmouseenter, onmouseleave, ondrag, etc.) som man kommer åt från JavaScript. Detta innebär att JavaScript-kod kan köras vid specifika händelser exempelvis när en användare klickar på bilden och att JavaScript-kod kan redigera DOM-strukturen för SVG filen genom att ändra befintlig information i SVG filen, lägga till ny information eller ta bort information [16][17].

Interaktiviteten som SVG möjliggör har använts i den interaktiva prototypen, hur detta gjorts beskrivs i kapitel 4.

2.4 Node.js

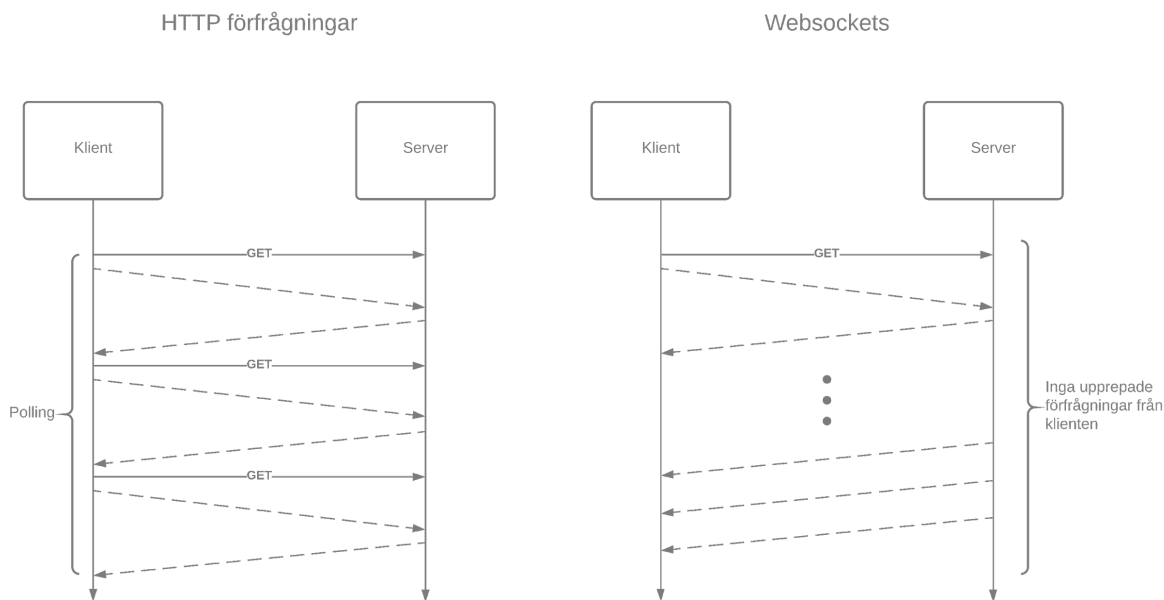
Node.js är en plattform med öppen källkod som används för att köra Javascriptkod i backendservrar. Node.js använder sig av asynkron programmering vilket innebär att servrar som hanterar förfrågningar med användning av Node.js kan arbeta effektivare genom parallell hantering av flera anslutningar [24][25].

Node.js har även en standard pakethanterare "NPM" [31] som möjliggör användning av node paket i JavaScript-applikationer, dessa paket är bibliotek med kodfiler som kan importeras till applikationer. Detta är jämförbart med importering av kodfiler med Gradle och Maven i Java.

Node.js anses vara lämplig för användning i detta examensarbete för att köra den interaktiva prototypen på en lokal server samt för hämtning av node paket. Användning av NPM för att köra React applikationer rekommenderas även av React [30].

2.5 Websocket

Websocket är ett protokoll som används för tvåvägskommunikation mellan en klient och en server. Detta innebär att en Websocket-anslutning mellan en klient och server gör det möjligt för servern att skicka meddelanden till klienten även om servern inte fått någon förfrågan från klienten [26].



Figur 8. Skillnaden på förfrågningar mellan protokollen HTTP och Websocket.

Figur 8 visar fördelen som Websockets har över HTTP i de sammanhang då en klient ofta behöver ta emot data från en server. Efter en initial anslutning mellan klienten och servern behöver klienten inte skicka fler förfrågningar till servern. Användning av HTTP skulle

innebära upprepade förfrågningar till servern för att potentiellt få data tillbaka, detta beror på om servern har data att skicka vid tillfället då en förfrågan tas emot från klienten. Om servern inte har någon data att skicka tillbaka är förfrågningarna från klienten en onödig belastning för servern [26].

På grund av fördelen som nämns ovan är det möjligt att skapa applikationer som kan ta emot data i realtid. Exempel på dessa kan vara aktieapplikationer som kan i realtid uppdatera priset av aktier eller applikationer som tillåter flera användare att redigera text samtidigt. Denna egenskap används i detta examensarbete [26].

2.6 Material-UI

Material-UI är ett ramverk som förenklar utvecklingen av webbapplikationer genom att erbjuda färdigbyggda Reactkomponenter som följer standarden för Material Design. Dessa komponenter kan användas för att undvika att själv behöva skapa komponenter som följer Material Design standarden, vilket bidrar till fortare och enklare utveckling [27]. Nackdelen med användningen av Material-UI komponenter är, likt nackdelen med Material Design, att examensarbetarna begränsas i kreativitet och variation då redan implementerade komponenter återanvänds i flera delar av samma applikation.

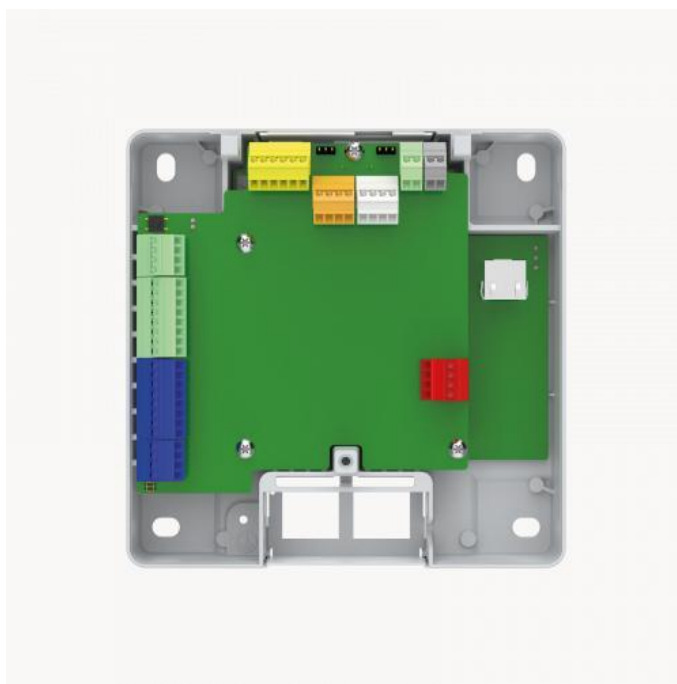
Material-UI's komponenter har använts för utvecklingen av den interaktiva prototypen och ingår i slutresultatet.

2.7 Trello

Trello är ett verktyg för att bland annat organisera projekt [20]. Med Trello kan ett projekt delas in i delmoment som antecknas på kort, likt virtuella post-it lappar. Korten organiseras i olika "boards" där betydelsen för varje "board" är upp till varje användare att bestämma. Korten kan sedan flyttas omkring för att visualisera projektets process samt hur långt man kommit i projektet. Med andra ord kan användandet av Trello medföra att projekt blir mer strukturerade, då man bland annat kan se vad som ska göras, vem som gör vad och hur långt man kommit i processen med ett delmoment eller flera.

2.8 AXIS A1601 Network Door Controller

AXIS A1601 är en enhet skapad av Axis communications som används för åtkomstkontroll ändamål [37]. Enheten drivs av ström via en ethernetkabel (Power Over Ethernet) och får också sin internetanslutning via denna kabel. Enheten kan kopplas upp till ett flertal enheter för att kontrollera dessa, detta kan exempelvis vara dörrar, kortläsare/knappsatser och knappar. Insidan av en A1601 illustreras i figur 9.

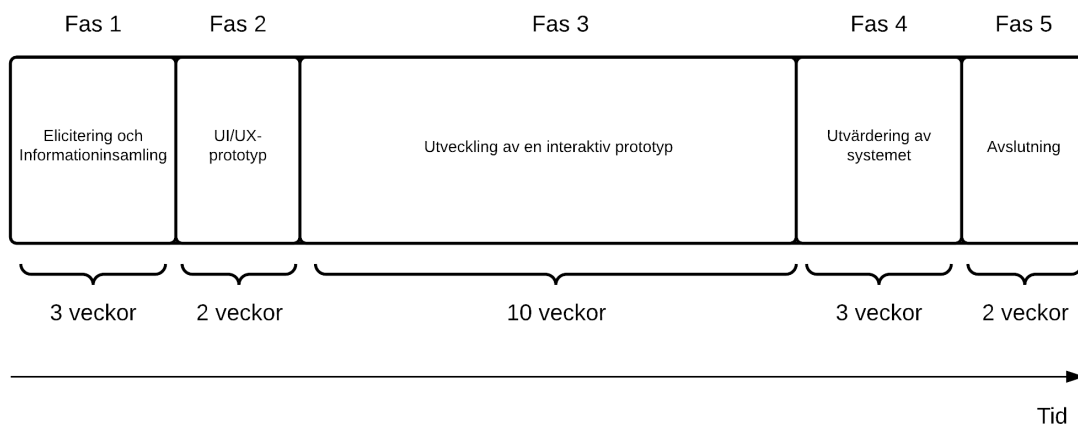


Figur 9. Insidan av en AXIS A1601 Network Door Controller [37].

3. Metod

I detta kapitel beskrivs tillvägagångssättet för detta examensarbete. Hur utvecklingen har skett samt vilka standarder, modeller och processer som följts för att uppnå resultatet med detta arbetet. Detta kapitel kommer även ta upp hur en del problem löstes och vilka åtgärder som krävdes för att förebygga att hinder uppstår.

Detta examensarbete har delats upp i fem olika faser där examensarbetarna arbetat sig stegvis igenom varje fas. Arbetandet genom faserna har skett i ordningen som illustreras i figur 10.



Figur 10. Uppdelningen av faserna.

Detta har varit en arbetsprocess lik vattenfallsmodellen där varje fas har avslutats innan påbörjandet av ett annat. Fördelen med detta har varit att examensarbetarna hade redan från början en plan på hur arbetet skulle utföras. Utöver det kunde examensarbetarna vid varje fas fokusera på vad som var relevant för just den fasen, detta innebar exempelvis att vid första fasen sattes enbart fokus på elicitering och informationsinsamling. Detta brukar vara en vanlig nackdel i modellen, återgående till en fas efter att det avslutas är inte något som brukar ske i vattenfallsmodellen. Detta identifierades som ett potentiellt problem tidigt i arbetet eftersom examensarbetarna inte var kunniga inom åtkomstkontroll. För att undvika problemet gjordes anpassningar till modellen vilket innebar att elicitering och informationsinsamling skedde under hela arbetets gång. Däremot skedde majoriteten av eliciteringen och informationsinsamlingen under första fasen.

Detta arbetssätt har främst använts för att framställa rapporten. Varje fas har resulterat i dokumentation som införts i denna rapporten. I faserna “UI/UX-prototyp” och “Utveckling av en interaktiv prototyp” som i tid utgjorde majoriteten av detta examensarbete, har modellen Scrum använts.

Scrum är ett välkänt projektmodell och används ofta för att framställa IT-system. Examensarbetarna har tidigare erfarenheter inom Scrum vilket var en viktig anledning till varför projektmodellen valdes. Projektmodellens fokus på “fungerande programvara framför

omfattande dokumentation” [38] var en annan viktig anledning till valet av denna projektmodellen. Detta innebar att fokus kunde sättas på att utveckla en interaktiv prototyp som ansågs vara det viktigaste för detta examensarbetets syfte. En annan fördel med Scrum har varit möjligheten för en kontinuerlig leverans. Funktionalitet som utvecklades under en Sprint kunde i slutet av Sprinten presenteras för handledaren. Hur Scrum har använts samt anpassningar som gjorts för modellen presenteras i kapitel 3.2.

I resterande faser (fas 4 och 5) spenderades mest tid på att framställa rapporten, resultatet av utvecklingsfasen dokumenterades i kapitel 4 och 5. Slutsatser framställdes i rapporten utifrån vad som efterfrågades av Axis för denna prototypen, prototypen som framställdes och intervjuerna som skedde med handledaren samt UX-designers på Axis.

3.1 Arbetssättet

Detta kapitel beskriver de fem faserna under detta examensarbete samt ger ett djupare inblick i hur resultatet av en fas hanterades.

3.1.1 Fas 1: Elicitering och informationsinsamling

Första fasen bestod av elicitering och informationsinsamling för prototypen som skulle utvecklas. Eliciteringen skedde främst genom en ostrukturerad intervju med handledaren samt “Engineering Manager” för Access Control-teamet på Axis Communications. Informationen som samlades in angående kraven för systemet dokumenterades i form av Scrum tasks på ett Scrumboard.

Ett krav gällande systemet var att Axis föredrog att detta utvecklades i ramverket “React”. Informationen för utveckling i React samlades in online, främst från Reacts egna webbsida men även från informativa videoklipp på Youtube.

Informationsinsamlingen har även bestått av diverse litteratur som fanns tillgängligt på internet. Litteraturen var allt från faktabaserade skrifter till kortare vetenskapliga artiklar eller informativa videoklipp på streaming-plattformar som bland annat Youtube. Inhämtningen av information med detta tillvägagångssättet har bidragit till den ökade kunskapen i ämnen inom områdena examensarbetet berör.

Arbetet med elicitering och informationsinsamling har varit en aktiv process under hela examensarbetet. Efter varje möte med Axis erhöles ny information och nya synvinklar, tankar, åsikter och ideér vilket innebar att nya ändringar fick göras som i sin tur påverkade andra faser i arbetets gång.

Intervjuer

För elicitering av information skedde bland annat ostrukturerade intervjuer främst med handledaren på Axis, syftet med dessa intervjuerna var att samla in mer information kring vad som förväntas av Axis, vilka önskemål de har för applikationen samt vilka krav som ska sättas på applikationen. Denna typ av intervju ansågs lämpligast då åtkomstkontroll var ett nytt

ämne för examensarbetarna som de inte var vana vid. Då fanns möjligheten att vara mer flexibel i diskussionen genom att kunna ställa frågor som uppstod i samband med pågående samtalsämne. På så sätt kunde mer information hämtas ut från intervjun enligt examensarbetarna. Dessa intervjuerna varade i ca. 30 minuter och skedde online via Microsoft Teams [21].

Eftersom dessa intervjuer skedde spontant i samband med möten dokumenterades inte frågorna eller svaren, däremot användes svaren under examensarbetet för vidare utveckling av den interaktiva prototypen samt för att uppnå målen med examensarbetet.

Utöver det skedde en intervju med två UX-designers hos Axis Communications där syftet var att samla in generell information kring olika målgrupper som finns för åtkomstkontroll-applikationer, råd kring hur man skapar en bra användarupplevelse, om planen att använda en interaktiv planritning (förklaras i kapitel 4) mm. Intervjun var semistrukturerad och utöver frågorna som ställdes så användes en del av tiden för att visa den interaktiva prototypen som hade utvecklats och få konstruktiv kritik.

Intervjuerna som skedde med UX-designers skedde online via Microsoft Teams [21]. Dessa varade i ca. 30 minuter och gick till på följande sätt att examensarbetarna till en början introducerade sig själva, arbetet som utförts samt annan relevant bakgrundsinformation. Därefter lästes frågorna upp av examensarbetarna och UX-designernas svar sammanfattades och antecknades. Anledningen till varför examensarbetarna valde att sammanfatta svaren istället för att ordagrant dokumentera svaren berodde på tidspress. Möjligheten att spela in ljudet var inte möjligt på grund av tekniska komplikationer och examensarbetarna fick därmed skriva en sammanfattning av varje svar. Frågorna som ställdes till UX-designers hos Axis Communications samt svaren på dessa kan läsas i Appendix (9.2).

Svaren på frågorna användes därefter under utvecklingens gång där delar av svaren utgjorde grunden för hur olika delar av applikationen skulle implementeras. Examensarbetarna fick även en ha dessa svaren i åtanke vid utveckling av andra relaterade delar av applikationen, särskilt svaret angående divergering från Material Design. Detta användes för att undvika så mycket som möjligt att avvika från Material Design då det standardiserade designen underlättar användningen av applikationen, se Appendix (9.2).

Litteratur

Böcker och skrifter som använts under examensarbetet listas i källförteckningen (kapitel 8). Syftet med litteraturstudierna var att ge ett fördjupat förståelse i områden som säkerhet, användarupplevelse (UX) och intervjuteknik.

Informationen som samlades in angående säkerhet var från boken [1]. Detta användes för att skriva en bakgrund om åtkomstkontroll i denna rapporten. Information angående användarupplevelse (UX) samlades främst in från Googles webbsida för Material Design [14]. Informationen användes vid utvecklandet av UI/UX-prototypen (fas 2), informationen var bland annat vilka färgkombinationer som är föredragna, hur layouten av en webbsida kan struktureras och vad som bör undvikas vid skapandet av en webbsidas användargränssnitt.

Studier om intervjueteknik kommer från boken [23]. Examensarbetarna undersökte olika intervjuetekniker som finns tillgängligt och vilken av dessa som kan vara lämpligt för intervjuerna som skedde.

Andra Informationskällor

Utöver information som samlats in från intervjuer och litteraturer så har andra informationskällor som Medium och StackOverflow undersökts och använts. Dessa källor användes under “Fas 3: Utveckling av en interaktiv prototyp” för bättre förståelse och debugging av React, Node.js och JavaScript. Eftersom all text och kod i Medium och StackOverflow kan fritt skrivas av vem som helst så sågs det till att all information som samlades in från dessa webbsidorna verifierades, detta innebar bland annat all kod som presenterades på dessa sidorna testades innan det implementerades i applikationen.

3.1.2 Fas 2: UI/UX-prototyp

Utifrån kraven på systemet konstruerades en UI/UX-prototyp i UI/UX-designverktyget Adobe XD [28]. För prototypen sattes fokus på att konstruera ett användargränssnitt som i huvudsak skulle följa Axis önskemål genom att presentera relevant åtkomstkontroll för information till den tänkta användaren och målgruppen som diskuterades med Axis, se Appendix (9.2).

UI/UX-prototypen var grunden för vidare utveckling av en interaktiv prototyp samt en möjlighet för validering och verifiering av Axis Communications. Prototypens design följdes inte strikt för att undvika begränsningar i kreativitet under utvecklingen av den interaktiva prototypen. Detta innebär att det fortfarande fanns möjligheter samt utrymme för förbättringar och förändringar i UI/UX-designen under utvecklingens gång.

Även om den framtida utvecklingen skulle avvika från den ursprungliga UI/UX-prototypen i avseende på det visuella användargränssnittet, sågs denna tidiga prototyp som en viktig grund för hela arbetet. Syftet med en prototyp som följer Axis önskemål var att i ett så tidigt skede som möjligt sätta upp en struktur för vilka de fundamentala delarna i detta arbetet skulle bli, det vill säga vilka delar som var viktiga för Axis och som måste finnas kvar oavsett hur användargränssnittet utformas i senare utveckling. Detta knyter an till den aktiva eliciteringsfasen (fas 1) som nämndes ovan, där återkopplingen från Axis i samband med presentation av arbetets faser är essentiell för examensarbetet beträffande informationsinhämtningen och utformningen av slutprodukten.

3.1.3 Fas 3: Utveckling av en interaktiv prototyp

För att utveckla den interaktiva prototypen användes Javascript-biblioteket React. Prototypen utvecklades i en Linux-miljö på datorer som tillhandahölls av Axis. Källkoden för mjukvaran skrevs i Visual Studio Code [29]. Källkoden versionhanterades med Git på GitLab [32].

Funktionalitet i applikationen utvecklades enligt prioriteringsordningen som sattes på de olika tasks som fanns på scrumboarden. Först sattes fokus på att sätta upp en fungerande

React-miljö. För att initialisera React-projektet användes Create-React-App [30] och NPM [31]. Detta kan man uppnå genom att mata in följande rader kommandon i kommandotolken:

```
npx create-react-app name-of-app
```

Där “name-of-app” är namnet som väljs för React-projektet. Detta initialiserar ett nytt React-projekt.

```
cd name-of-app
```

Sedan måste man navigera till React-projekt-mappen i kommandotolken. Mappen för React-projektet finns på den plats där den initialiserades med föregående kommandorad.

```
npm start
```

Sedan körs kommandoraden ovan för att starta en lokal server som applikationen körs på. Denna server kan komma åt från andra enheter som är uppkopplade till samma nätverk. Alla ändringar syns på applikationer när sidan uppdateras.

Därefter skapades komponenterna som utgjorde UI för den interaktiva prototypen. Majoriteten av komponenterna som knappar, text och former importerades från Material-UI ramverket och sedan skraddarsyddes dessa för att uppfylla UI/UX-prototypens design som gjordes i föregående fas.

När den initiala UI/UX-designen var på plats så skapades SVG modellen av en imaginär planritning som gjordes interaktiv genom SVG-manipulationer och injektioner.

Därefter skapades komponenter med mer funktionalitet som exempelvis att visa information (se Appendix A.7 - A.8) eller styra andra delar av applikationen. Informationen som visades var temporära platshållare som ersattes senare under utvecklingen.

Sedan skapades en Node.js server för att testa Websockets-uppkopplingen från applikationen till servern. Vid detta steg testades även HTTP-anrop från applikationen. Dessa uppkopplingar omdirigerades sedan till Axis interna API.

HTTP-anrop till Axis API användes för att hämta ut information om kortinnehavare, zoner och enheter. Websockets anslutningen användes för att få data i realtid när en kortinnehavare interagerar med kortläsare.

3.1.4 Fas 4: Utvärdering av systemet

Utvärderingen av systemet bestod av generella användartester där examensarbetarna testade om utvecklad funktionalitet fungerade enligt motsvarande task på scrumboarden som utfördes av examensarbetarna och en utvärdering av applikationen genomfördes av handledaren på Axis. En mer omfattande analys av arbetet som helhet, där applikationen samt processen för hela examensarbetet reflekteras kring, beskrivs i kapitel 4.

Efter testerna och utvärderingarna åtgärdades eventuella fel som uppstod innan applikationen återigen utvärderades. Dessa fel kunde vara allt från små buggar i applikationen till hela funktioner som inte fungerade som tänkt. Detta skedde iterativt tills en omgång tester och utvärderingar inte genererade några fel.

3.1.5 Fas 5: Avslutning

Utvecklingen har under hela examensarbetets gång funnits på Axis GitLab-sida där den som tidigare nämnt versionshanterats. Handledaren på Axis hade tillgång till senaste versionen av applikationen genom GitLab och kunde därför mottaga den där. En slutgiltig version av examensrapporten färdigställdes och skickades in till handledaren samt examinator på LTH.

3.2 Projektmodellen

Under detta examensarbete följdes projektmodellen Scrum som bygger på att arbetet utförs enligt strukturerade faser [18].

3.2.1 Scrum

Scrum är en agil projektmodell som implementerar ett heuristiskt tillvägagångssätt för att genomföra ett projekt. Systematiken för scrum består av en s.k. “*Backlog*” som innehåller alla delar av examensarbetet som ska utföras för att uppnå det önskade resultatet. Utvecklingsteamet bestämmer sedan längden på intervallet för projektets “*Sprintar*”. Sedan plockas delar av innehållet från Backloggen ut och sätts in i “*Sprintlog*”. Det som står i Sprintloggen ska sedan utföras under en sprint. Längden för sprinten, vem som gör vad samt när möten ska hållas är exempel på sådant som utvecklingsgruppen själva utser och anpassar till examensarbetet [18].

3.2.2 Scrum Board

Under utvecklingens gång användes Trello som Scrum Board. Arbetsuppgifter delades i mindre delar och för varje arbetsuppgift skapades ett kort. Kortet innehöll en beskrivning av arbetsuppgiften, dess prioritet (låg, medel, hög) och en som var ansvarig för arbetsuppgiften. Se Appendix (figur A.10.1) för ett exempel på hur scrumboarden såg ut.

Ett kort kunde placeras under en av flera listor, nedan listas dessa:

1. **Backlog:** Funktioner som var tänkta att implementeras i framtiden placerades i backloggen. Nya funktioner lades till i listan närsom dock kunde en funktion i backloggen inte flyttas in till Sprint Backloggen under en sprint utan gjordes endast i början av en ny sprint.

2. **Sprint Backlog:** Listan innehöll funktioner som skulle implementeras under pågående sprint.
3. **In Progress:** Listan innehöll funktioner som höll på att implementeras i applikationen.
4. **Testing:** Funktioner som har implementerats flyttas till “Testing” för att testas av båda examensarbetarna.
5. **Sprint Done:** Funktioner som har passerat testandet under pågående sprinten och anses fungera som tänkt flyttades till denna listan.
6. **Done:** Påbörjandet av en ny sprint innebar att alla kort som fanns under “Sprint Done” flyttades till “Done”. Denna listan fungerade som ett arkiv för alla funktioner som implementerats i applikationen.

3.2.3 Anpassningar till projektmodellen

För att modellen ska kunna följas för detta examensarbete, krävs det en del anpassningar. Detta eftersom examensarbetarna har saker att ta hänsyn till som exempelvis Covid-19 restriktioner, handledaren på Axis arbetstider och obligatoriska moment i andra kurser som sker parallellt med examensarbete. Dessa anpassningar innebär bland annat:

- **Roller:**
Arbetsuppgifter som Scrum-master, produktägare och utvecklingsteam ansvarade för hanterades i stor grad av examensarbetarna och tilldelades inte till någon specifik person. Dessa arbetsuppgifter bestod bland annat av hanteringen av backlog, ledandet av Scrum möten, hantering av Scrum Board mm. Detta var en nödvändig anpassning då resurserna samt förutsättningarna för en Scrum-master, produktägare och utvecklingsteam inte fanns för detta examensarbetet. Dessutom ansågs det, enligt examensarbetarna, vara gynnsamt att båda arbetarna antog alla roller, eftersom det bidrar till att fler perspektiv inkluderas vilket i sin tur innebär en större nyansering, framförallt i slutresultatet.
- **Sprintar:**
Varje Sprint var 14 dagar lång, detta ansågs av examensarbetarna att vara en bra tidsram för en relativt kort utvecklingstid som fanns för prototypen. Då möjligheten att arbeta med prototypen varje vecka inte var sannolikt ansågs 7 dagar långa Sprints vara för korta. En Sprint längre än 14 dagar ansågs vara för långa på grund av korta tidsramen som fanns för detta examensarbete. En 14 dagar lång Sprint innebar även att i slutet av en Sprint kunde resultatet av Sprinten visas för handledaren på Axis, vilket var en lagom tid för att visa en uppdatering av vad som utvecklats i applikationen. Varje möte som skedde under en Sprint skedde via kommunikationsplattformen Discord [19]. Möten hölls online främst på grund av Covid-19 restriktionerna som fanns.

- **Backlog och Sprintlog:**
Backlog och Sprintlog fördes i Trello [20], en applikation som låter användare skapa anslagstavlor för att bland annat styra upp och strukturera projekt genom att skapa kort med delmoment som kan hanteras på olika sätt för att visualisera projektets utveckling (se Appendix A.10.1). Då utvecklingen skedde online pga Covid-19 restriktioner innebar det att Backloggen och Sprintloggen fick representeras digitalt.
- **Sprint retrospektiv:**
Retrospektiv-möte hölls med jämna mellanrum, där huvudfokuset var att analysera huruvida projektmodellen följdes, vilka brister det fanns samt vad som fungerade bra. Detta ansågs vara en viktig del av projektmodellen och för detta examensarbete då nya perspektiv lyftes fram och diskuterades. Från dessa återkopplingar skedde förbättringar i den interaktiva prototypen. Utöver det fanns det alltid ny funktionalitet som hade implementerats i applikationen i slutet av en Sprint, detta innebar att en kontinuerlig leverans av nya versioner av applikationen kunde visas för handledaren på Axis.
- **Prioriteringar:**
Alla tasks på scrumboarden fick en prioritering baserad på hur viktig de ansågs vara inför slutversionen av prototypen. De tasks som fick en "hög" prioritet var funktionalitet som måste finnas med i slutversionen av prototypen. De tasks som fick en "medel" prioritet ansågs vara viktiga och borde vara med i slutversionen av prototypen. De tasks som fick en "låg" prioritet bestod av funktionaliteter som inte ansågs vara viktiga för den slutliga versionen av prototypen och kunde utvecklas i mån av tid. Denna prioriteringsmetoden (kategorisering) användes för dess enkelhet. Prioriteringen skedde relativt fort och fokus kunde sättas på att utveckla applikationen.

Syftet med anpassningarna var att göra arbetssättet optimalt för detta examensarbete. Detta innebär att modellen måste kunna följas även om hinder uppstår som exempelvis tentamen, omtentamen, obligatoriska moment i andra kurser etc. Dessutom måste arbetssättet även anpassas till Axis arbetssätt, vilket medför att arbetet kan ske parallellt och synkroniserat. Alla dessa åtgärder innebär i sin tur att samarbetet och utvecklingen effektiviseras.

3.3 Kommunikation

I detta delkapitel beskrivs hur kommunikationen sköts, dels mellan medlemmarna inom examensarbetet och dels mellan examensarbetarna och de olika aktörerna.

3.3.1 Kommunikation mellan examensarbetarna

På grund av kontakt-restriktioner till följd av Covid-19 pandemin sköttes kommunikationen mellan examensarbetarna till största del på distans. Detta eftersom det bland annat rädde brist på tillgängliga mötesplatser och även för att undvika smittspridning i så stor grad som möjligt.

Med detta i åtanke skedde fysiska möten endast vid behov, där bedömningen av behovet gjordes beroende på omständigheter i förhållande till examensarbetets utveckling.

Kommunikationen mellan examensarbetarna skedde via internet på olika applikationer för kommunikation. Dessa applikationer var bland annat Discord och Microsoft Teams [21]. Utöver dessa applikationer sköttes kommunikationen tidvis via SMS. Mest förekommande formen av kommunikation var kortare, mer frekventa och mer informerande textmeddelanden. Även röstsamtal förekom som kommunikation vid större diskussioner och mer omfattande delmoment.

Eftersom examensarbetet var på heltid var kommunikationen relativt frekvent. Någon form av statusuppdatering för det pågående arbetet med delmomenten hölls i genomsnitt 4 gånger per vecka.

3.3.2 Kommunikation med Lunds Universitet

Kommunikation med handledare på Lunds Universitet skedde på distans genom användning av kommunikationsplattformen Zoom [22]. Avstämningar bestod bland annat av feedback på rapporten samt eventuella frågor som examensarbeterna hade.

3.3.3 Kommunikation med Axis

Tidig kommunikation med företaget skedde fysiskt i Axis huvudkontor i Lund. På grund av Covid-19 restriktioner övergick all kommunikation och arbete med företaget till distanslösningar. För att kommunicera med handledare och andra arbetare på Axis användes kommunikationsplattformen Microsoft Teams.

Enklare funderingar diskuterades via text medan större diskussioner och avstämningar skedde via röst- eller videosamtal. Avstämningar med företaget skedde oftast 1 gång om veckan med undantag av tentamensperioder. Dessa avstämningar bestod bland annat av redovisningar för vad för sorts arbete som hade genomförts under veckan, problem som stöttes på under arbetet, vad som var planen för kommande veckan och andra eventuella tankar.

Kommunikation med andra anställda hos Axis, bland annat UX ingenjörer skedde även via Microsoft Teams, detta främst för intervjuer samt feedback på arbete.

3.4 Källkritik

3.4.1 Kurslitteratur

Kurslitteratur som använts under detta examensarbete består av böckerna [1, 23]. Kursböckerna har använts under studier på Lunds Tekniska Högskola. Informationen i böckerna anses vara pålitliga då Pearson anses vara en betrodd utgivare samt att

böckerna är skrivna av författare med en relevant koppling till området böckerna täcker.

3.4.2 Axis

Information som samlats in angående företaget Axis Communication består av webbsidorna [2, 37]. Eftersom informationen kommer från Axis egna webbsida anses denna källan ha hög trovärdighet.

3.4.3 React

Informationsinsamling om React har skett från webbsidorna [3, 4, 6-8, 34]. Informationen från dessa källor kommer från Reacts egna webbsida. Därför anses dessa vara pålitliga.

3.4.4 Babel

Informationsinsamling om Babel har skett från webbsidan [5]. Informationen från källan kommer från Babels egna webbsida. Därför anses den vara pålitlig.

3.4.5 Material Design & Material UI

Informationen som använts för Material Design och Material UI kommer från webbsidorna [14, 27]. Informationen anses vara pålitlig då det kommer från Material Design respektive Material UIs egna webbsidor. För Material Design är det Google som ligger bakom standarden, vilket förstärker informationens trovärdighet.

3.4.6 Mozilla

Information som samlats in från Mozilla angående SVG och DOM kommer från webbsidorna [15-17]. Källan anses vara trovärdig eftersom Mozilla är väl ansedd i IT-branschen.

3.4.7 Scrum

Informationen som samlats in för förklaringen av Scrum kommer delvis från webbsidan [18] och delvis från tidigare erfarenheter. Källa [18] skapades bland annat av Ken Schwaber som är en av Scrums grundare, därför anses denna källa vara väldigt trovärdig.

3.4.8 Discord

Programmet som främst använts för kommunikation mellan examensarbetarna finns tillgänglig för nedladdning från webbsidan [19]. Från denna källan har ingen information utan syftet har enbart varit för hämtningen av programvaran. I fråga om dess pålitlighet kan det anses pålitligt i förhållande till säkerhetsnivån för

meddelandena som skickas via Discord. Detta eftersom Discord endast erbjuder standard kryptering och inte end-to-end kryptering av något slag [19].

3.4.9 Trello

Webbapplikationen som använts för backlog och sprintlog under planering i syfte att följa Scrum finns tillgänglig online från webbsidan [20]. Ingen information hämtas från denna källa, huvudsakliga syftet med användandet av källan har varit för tillgång till webbapplikationen. Med tanke på att Trello använder SSL kryptering för all trafik kan källan anses pålitlig i förhållande till säkerhetsnivån på informationen som förs via Trello.

3.4.10 Microsoft Teams

Programmet som främst använts för kommunikation mellan examensarbetarna och Axis Communications finns tillgänglig för nedladdning från webbsidan [21]. Från denna källa har ingen information hämtats utan syftet har enbart varit för att hämta programvaran. I fråga om dess pålitlighet kan det anses pålitligt med tanke på att Axis endast använder detta program för kommunikation med examensarbetarna. Däremot erbjuder Microsoft Teams inte end-to-end kryptering av något slag utan endast standard kryptering [21].

3.4.11 Zoom

Programmet som främst använts för kommunikation mellan examensarbetarna och Lunds Universitet finns tillgänglig för nedladdning från webbsidan [22]. Från denna källa har ingen information utan syftet har enbart varit för att hämta programvaran. I fråga om dess pålitlighet kan det anses pålitligt med tanke på att Zoom erbjuder TLS 1.2 kryptering med AES (Advanced Encryption Standard) 256 bitars algoritm för meddelanden som skickas via Zoom [22].

3.4.12 Node.js

Informationsinsamling om Node.js har skett från webbsidan [24]. Informationen från källan kommer från Node.js egna webbsida. Därför anses den vara pålitlig.

3.4.13 W3Schools

Informationsinsamling om Node.js och JavaScript har skett från webbsidan [25]. W3Schools är välkänd och en respekterad källa bland utvecklare runt om i världen. Utöver det har all kod-material som samlats in från denna källa testats innan det använts i den interaktiva prototypen.

3.4.14 IETF

Informationsinsamling om WebSocket har skett från IETFs webbsida [26]. Internet Engineering Task Force (IETF) är en organisation som skapar internet-standarder. Denna organisationen står bakom många välkända protokoll som WebSocket och TLS, därför anses denna källan att vara pålitlig.

3.4.15 Adobe XD

Informationsinsamling om Adobe XD har skett från webbsidan [28]. Informationen från källan kommer från Adobe XD egna webbsida. Därför anses den vara pålitlig.

3.4.16 Visual Studio Code

Informationsinsamling om Visual Studio Code har skett från webbsidan [29]. Informationen från källan kommer från Visual Studio Codes egna webbsida. Därför anses den vara pålitlig.

3.4.17 NPM

Informationsinsamling om NPM har skett från webbsidan [31]. Informationen från källan kommer från NPMs egna webbsida. Därför anses den vara pålitlig.

3.4.18 GitLab

Informationsinsamling om GitLab har skett från webbsidan [32]. Informationen från källan kommer från GitLabs egna webbsida. Därför anses den vara pålitlig.

3.4.19 Pierre Carbonnelle GitHub

Källan för Pierre Carbonnelle Github är webbsidan [33]. Informationen i källan använder sig av data som kommer från Google Trends. Google anses vara en pålitlig källa och det är möjligt att bekräfta att datan som presenteras i sidan är sann genom att jämföra datan som presenteras i sidan med datan som finns på Google Trends.

3.4.20 Adobe Illustrator

Källan för Adobe Illustrator är webbsidan [35]. Informationen från källan kommer från Adobes egna webbsida. Därför anses den vara pålitlig.

3.4.21 Hugo Zapata GitHub

Källan för Hugo Zapata Github är webbsidan [36]. Informationen som presenteras på denna sida är för node paketet som kan importeras till projektet. Informationen ansågs vara pålitlig då paketet importerades och testades.

3.4.22 Manifesto for Agile Software Development

Källan för Manifesto for Agile Software Development är webbsidan [38]. Informationen som presenteras i webbsidan är skrivet av skaparna av manifestet. Därför anses den vara pålitlig.

4. Analys

Detta kapitel beskriver de beslut som tagits utifrån informationsinsamling och elicitering som skedde under arbetets gång, kapitlet behandlar även de problem som stöttes på under arbetets olika faser samt deras lösningar.

4.1 Fas 1: Elicitering och informationsinsamling

Under första fasen fick examensarbetarna information angående prototypen som eftersöktes och ytterligare information om prototypens krav samt Axis Communications önskemål för systemet samlades in genom ostrukturerade intervjuer. Företaget hade en befintlig lösning för övervakning av endast en dörr och var då intresserade av ett koncept som möjliggör övervakning av flertal dörrar samtidigt.

Problemet under denna fas var främst att identifiera en användarvänlig lösning för att möjliggöra övervakning av flertal dörrar samtidigt, detta innebär att användaren ska med maximalt 1-2 klick lyckas öppna eller stänga en dörr, och ha en översikt av alla dörrar på en sida. Kraven för övervakningen innehöll bland annat möjligheten att kunna få ut information som dörrens nuvarande status (öppen, stängd, ej fungerande etc.), vem som interagerat med dörrens kortläsare, hur länge dörren varit öppen (om den är öppen), se historik över vem som interagerat med dörren över en viss tidsram mm.

Den föreslagna lösningen av examensarbetarna var att använda en interaktiv planritning som kan visuellt hjälpa en operatör att exempelvis identifiera icke-fungerande dörrar, personer som interagerar med en viss dörr och även kontrollera dörrar via planritningen som att låsa eller låsa upp dörrarna.

Lösningen presenterades för handledaren på Axis Communications och därefter diskuterades lösningen. Handledaren tyckte att lösningen var adekvat och kunde förenkla arbetsprocessen för operatörer. Handledaren presenterade även ett exempel där användningen av en interaktiv planritning kan förenkla arbetsprocessen för operatörer som övervakar järnvägar där det ibland kan vara svårt att identifiera var en icke-funktionell enhet är på järnvägen.

För att utveckla prototypen bestämdes det att React skulle användas med Hewlett-Packard laptops som hade operativsystemet Linux förinstallerat, detta var ett önskemål från Axis. På grund av omständigheterna gällande Covid-19 var det vid det här stadiet känt att utvecklingen av prototypen inte kunde ske på plats på Axis huvudkontor. Då examensarbetarna behövde komma åt Axis-tjänster som är endast åtkomliga inom Axis egna nätverk så gavs examensarbetarna användarkonton hos Axis vilket gjorde det möjligt att komma åt Axis-tjänster via VPN.

För att skapa en UI/UX-upplevelse som de flesta användare känner igen så planerades användningen av designstandarden Material Design. Detta är en designstandard som är välkänd och är en standard som används även inom Axis. Förslaget att använda Material

Design möttes av positiva reaktioner från Axis sida och därför togs beslutet att Material Design skulle utgöra grunden för UI/UX-upplevelsen i applikationen.

4.2 Fas 2: UI/UX-prototyp

Fokuset under den andra fasen var att konkretisera all information samt alla önskemål som samlats in, i form av en UI/UX-prototyp. Denna prototyp gjordes i Adobe XD eftersom detta program erbjuder funktioner som examensarbetarna efterfrågade. Dessa funktioner var bland annat möjligheten att lägga till komponenter som kunde ändras och anpassas efter examensarbetarnas behov. Utöver detta är Adobe XD kostnadsfritt, som var en avgörande faktor i valet av designverktyg för UI/UX-prototypen.

Med tanke på att UI/UX-prototypen skulle utgöra en grund för användargränssnittet av den interaktiva prototypen, delades arbetet av UI/UX-prototypen upp i två inofficiella block. Första blocket handlade om att skapa utseendet för prototypen samtidigt som den skulle vara lätt att använda och inte avvika för mycket från vad användarna är vana vid.. Andra blocket syftade till att integrera de önskade funktionerna i användargränssnittet på ett naturligt sätt, det vill säga på ett sätt som är logiskt ordnat eller strukturerat.

Utseendet samt integreringen av funktionerna är sammanhängande och examensarbetarna var tvungna att finna en balans mellan att integrera så många av de önskade funktionerna som möjligt utan att det ska vara på användarvänlighetens bekostnad. Samtidigt ska inte kreativiteten begränsas som följd av att vissa funktioner skulle integreras.

UI/UX-prototypens utseende tänktes igenom väl. Det handlade bland annat om färger, knappars utseende, komponenters utseende, textstil och navigering. Som grund utgick examensarbetarna från Axis Communications logotyp i val av färgschema. Logotypens orangea del inspirerade bland annat den orangea färgen på prototypen (Appendix A.1-A.9). Gällande textstil, knapparnas- och komponenternas utseende föredrogs mjukare (rundade) kanter över hårda kanter, detta då mjuka kanter är modernt och vanligt förekommande i nya applikationer eller mjukvaror som görs. Knapparnas symboler skulle även spegla funktionen den har, exempelvis ska en kugghjul-knapp ta användaren till inställningar.

Examensarbetarna fick överväga vilka funktioner som ansågs essentiella för examensarbetet och vilka funktioner som kunde uteslutas. När grunden för UI/UX-prototypens utseende fastslagits, skulle examensarbetarna integrera funktionerna och anpassa dessa till gränssnittet. Resultatet av detta innebar en modern, användarvänlig och välbalanserad prototyp med alla fundamentala funktioner integrerade (Appendix A.1-A.9).

När den slutgiltiga UI/UX-prototypen hade utvecklats, skulle den utvärderas av handledaren på Axis Communications. Utvärderingen möttes av positiv återkoppling och konstruktiv kritik. Kritiken handlade i stora drag om att en del funktioner i UI/UX-prototypen inte behövdes i den interaktiva prototypen (funktioner som inte var nödvändiga hittas i Appendix A.1-A.3). Åtgärder gjordes för UI/UX-prototypen och examensarbetarna hade en bra grund för att påbörja arbetet med utvecklingen av den interaktiva prototypen.

4.3 Fas 3: Utveckling av en interaktiv prototyp

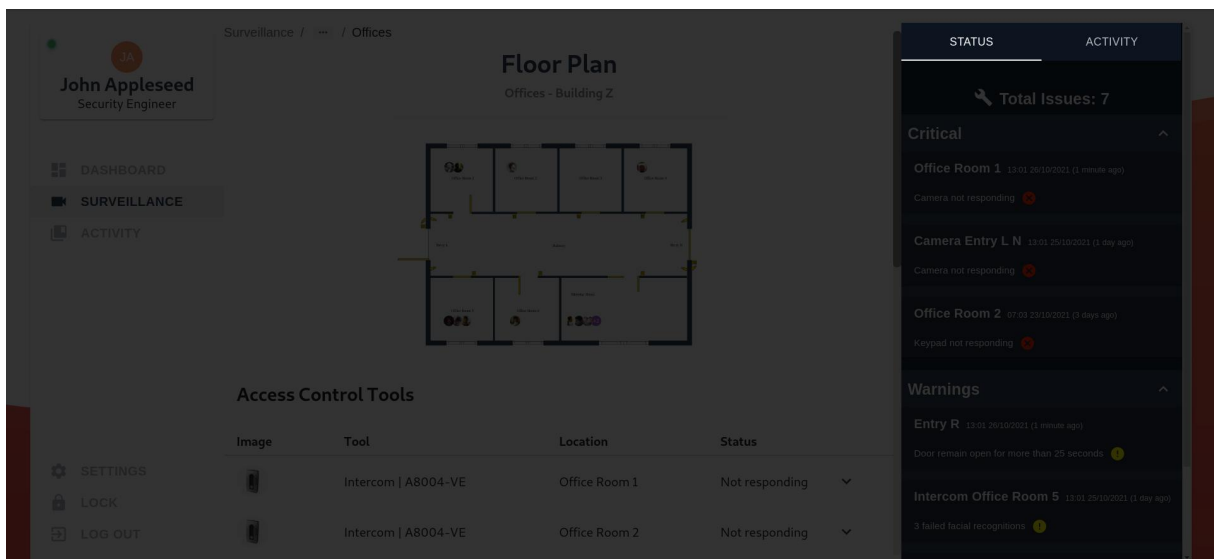
Utvecklingen av prototypen skedde i Visual Studio Code och användningen av denna utvecklingsmiljön var examensarbetarnas val. I skrivandes stund är Visual Studio Code en av de mest populära utvecklingsmiljöerna [33] och är den utvecklingsmiljö som examensarbetarna var bekväma med sen tidigare.

För att skapa en ny React-projekt användes Create React App. Integration av React i ett projekt kan ske på flera sätt, React rekommenderar att Create React App används för nya React-användare [34]. Eftersom React var ett bibliotek som examensarbetarna inte arbetat med tidigare togs beslutet att följa Reacts rekommendation och använda sig av Create React App.

Det första som utvecklades för applikationen var dess UI. Då det var bestämt att Material Design skulle utgöra grunden för applikationens UI/UX rekommenderades Material-UI av handledaren på Axis. Eftersom Material-UI har färdigbyggda komponenter som följer Material Design standarden så var det tänkt att detta skulle underlätta arbetet genom mindre utvecklingstid spenderad på funktionaliteter som knappar, menyer och typografi. Ramverket inkluderas i applikationen genom pakethanteraren NPM och användes under hela utvecklingsfasen.

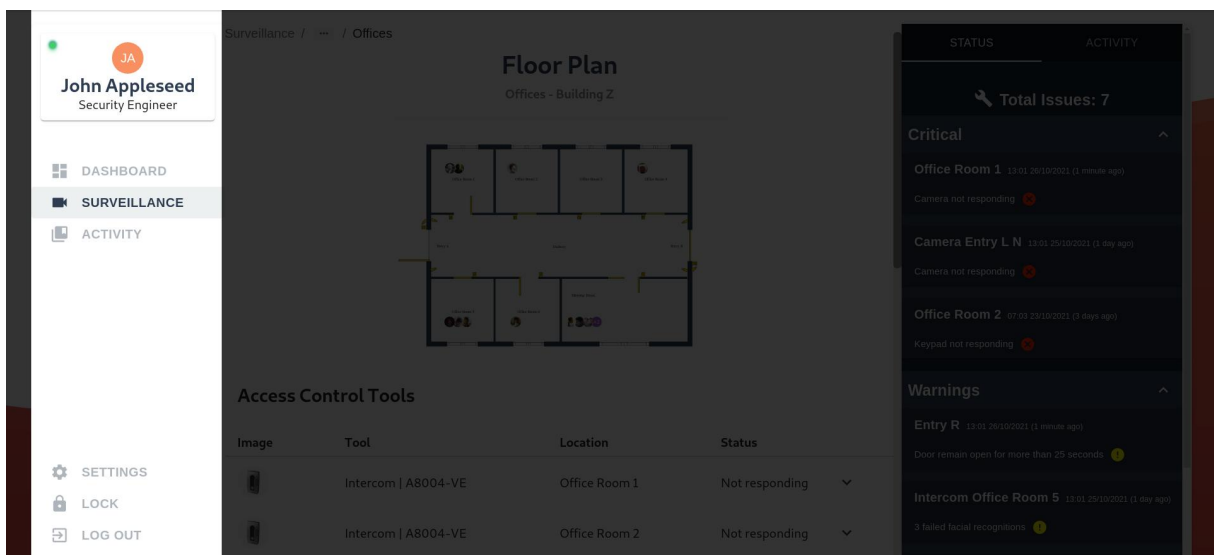
Ett exempel av den planerade strukturen för en specifik sida kan ses i Appendix (A.7). Eftersom UI/UX-prototypen inte följer Material Design-standardens innebar detta att under utvecklingen ersattes delar av UI/UX-prototypen med komponenter från Material-UI i den interaktiva-prototypen. Detta ledde till problem som inte förutsågs av examensarbetarna under den andra fasen. Integrationen av vissa komponenter från Material UI-ramverket fungerade som tänkt och passade in UI/UX-mässigt dock fanns det vissa komponenter som förändrade den tänkta UI/UX-upplevelsen som examensarbetarna designade under den andra fasen.

Figur 11 visar i det belysta området en Material-UI komponent i den interaktiva prototypen som har ersatt designen från UI/UX-prototypens variant som kan ses i Appendix (A.7).



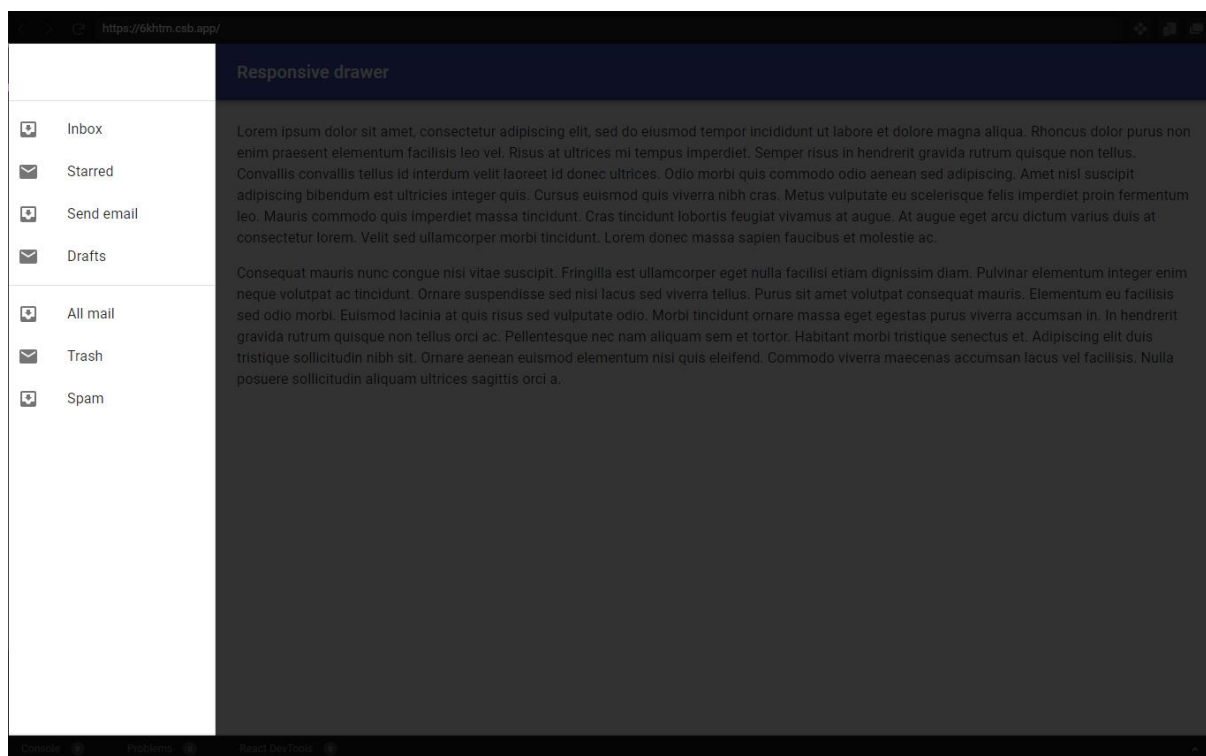
Figur 11. Det belysta området (upp till höger) visar en Material-UI komponent.

En komponent som inte kunde integreras utan att strida mot den planerade UI/UX-designen som visas bland annat i Appendix (A.7) var navigationsmenyn. Figur 12 visar de minimala UI-förändringar som skett med integrationen av Material-UI komponenter som menyknapparna och typografin. Alltså beslutet som togs var att behålla det eget designade menyn från UI/UX-prototypen med vissa ändringar som ökar användarvänligheten genom att använda Material Design typografi, ikoner och knappar.



Figur 12. Det belysta området visar en blandning av Material-UI design och skräddarsydd design.

Figur 13 visar hur en navigationsmeny från Material-UI ser ut UI-mässigt. Navigationsmenyn har fästs till vänster av webbläsaren och består av en lista med navigationslänkar. Valet att divergera från Material Design för att behålla alla komponenternas koherens i sidan bemöttes med förståelse från handledaren och UX-ingenjören.



Figur 13. Det belysta området visar navigationsmenyn (drawer) som finns tillgänglig som komponent i Material-UI.

När den fundamentala UI-designen för sidan var färdigt kunde planeringen och utvecklingen av en interaktiv-planritning utföras. Initialt fanns det ingen kännedom om hur detta kunde utföras på ett genomförbart sätt. Lösningen krävde att applikationen kunde visa upp en bild som kan uppdateras i realtid. Efter flera undersökningar, bland annat i Mozillas webbsida hittades en beskrivning om SVG-vektorgrafik som verkade uppfylla lösningens krav.

Framtagandet av en planritning skedde i vektorgrafikprogrammet Adobe Illustrator [35]. För första versionen av planritningen ritades dörrar, kortläsare/knappsatser och kameror. Dörrarna som visas i figur 14 innehåller två lägen för en dörr, ett läge som är då dörren är stängd och ett läge då dörren är öppen. Detta gjordes för att undvika rotering och förflyttning av en dörr i programkoden. Detta blev en avvägning mellan svårare programkod och större filstorlek, eftersom filstorleken mellan dörrar med två lägen (99.3 kB) och dörrar med ett läge (90.6 kB) skiljde sig åt med endast 8.7 kB ansågs det av examensarbetarna vara bättre att underlätta utvecklingen genom att använda ritningen med dörrar som har två lägen. Detta innebär att en dörr kan via programkod visas som stängd eller öppet genom att gömma synligheten för ett av de två lägen som dörrarna kan ha.



Figur 14. Första versionen av planritningen.

I planritningen måste alla dörrar, kameror och kortläsare/knappsatser ha unika identifierare som gör det möjligt i React-programkoden att identifiera och manipulera dessa element. Ett stort problem med detta var att examensarbetarna inte lyckades hitta någon metod att ge elementen unika identifierare direkt i Adobe Illustrator, följden av detta var att elementens identifierare fick sättas manuellt i efterhand genom att redigera i SVG-filens kod.

För att underlätta programkoden och möjliggöra integration av nya planritningar i applikationen konstruerades en metod för att namnge elementen i planritningen. Alla enheters unika identifierare använder sig av Snake case och innehåller namnet av rummet som den tillhör. Ett exempel är dörren som leder till "Office Room 1" har en unik identifierare som blir:

1. För läget då dörren är stängd:

- Office_Room_1_Door_Closed

2. För läget då dörren är öppen:

- Office_Room_1_Door_Open

För kameror och knappsatser behövs inte två lägen för att representera dess nuvarande läge, dessa enheter kommer att vara stationära vilket innebär att endast ett läge är tillräckligt för att

representera enheten på planritningen. Dessa enheter fick en uppsättning för deras unika identifierare som kan se ut så här:

3. Läget för kameror:

- Entry_L_Camera

4. Läget för knappsatser/kortläsare:

- Entry_L_Intercom

Generellt från dessa kan man se att identifieraren innehåller 3 delar som är:

1. Namnet på rummet som enheten tillhör.
2. Vilken sorts enhet det är.
3. Enhetens nuvarande läge (om enheten har mer än ett läge).

Genom att följa denna metod för att namnge alla enheter som finns på en planritning skulle detta teoretiskt sätt möjliggöra integrationen av nya planritningar i applikationen utan behöva manuellt stödja dem i programkoden

När planritningen var färdigritad och all tilldelning av unika identifierare till enheterna var klart så importerades SVG-filen till React-applikationen. Nästa steg var att manipulera SVG-filen för att möjliggöra interaktivitet. Interaktivitet som söktes av examensarbetarna var bland annat att kunna klicka på dörrar för att få upp en popup-ruta med information om dörren och kunna utföra kommandon som att låsa eller låsa upp en dörr, andra former av interaktivitet som söktes var att markera enheter som är trasiga och visa personers nuvarande position på planritningen.

För att genomföra det ovannämnda användes ett npm-paket (react-svgmt) som underlättar manipulation av SVG-filer [36]. En alternativ metod för att manipulera SVG-filer i React är att använda sig av "refs" som gör det möjligt att modifiera delar av DOM i webbsidan. Utöver Reacts rekommendation att generellt undvika att överanvända refs så upplevdes refs att vara komplicerad för att manipulera stora delar av SVG-filen. Detta var den primära anledningen till användningen av react-svgmt.

Genom att använda react-svgmt kunde varje elements attribut uppdateras. Detta inkluderar bland annat elementens synlighet, färg och storlek. Ett exempel på hur detta sker i applikationen visas i figur 15, i detta exempel så sätts den stängda dörrens synlighet till att vara gömd och den öppna dörrens läge sätts till att vara synlig.

```
<SVGProxy selector="#Office_Room_1_Door_Closed" visibility="hidden" />
<SVGProxy selector="#Office_Room_1_Door_Open" visibility="visible" />
```

Figur 15. Exempel på hur SVG-manipulation sker i applikationen med react-svgmt.

När enheterna kunde manipuleras via programkoden fortsatte utvecklingen med SVG-filen genom att implementera "avatars" på planritningen, dessa visar var på planritningen en specifik person befinner sig. Detta var ett stort problem som tog en stor del av utvecklingstiden. Den första metoden som testades var att försöka kalkylera positionen på bildskärmen där en avatar borde placeras och rita bilden på denna plats. Denna metod misslyckades på grund av att examensarbetarna inte tog hänsyn till svårigheten att behöva kalkylera om positionen för varje avatar då webbfönstrets storlek förändrades. Detta ledde till att många omritningar skedde då webbfönstrets storlek ändrades och utöver ompositioneringen av en avatar krävde detta nya kalkyleringar av varje avatars storlek för att ha en relativ storlek som ökar då webbfönstrets storlek ökas och minskas då webbfönstrets storlek minskas. Därför skrotades denna metoden för en alternativ metod.

För att undvika ständiga ompositioneringar och storleksförändringar användes Adobe Illustrator för att modifiera planritningen som visas på figur 14. Figur 15 visar inkluderingen av cirklar, dessa cirkelarnas bakgrund kan ändras via SVG-manipulation i programkoden till att visa en bild som specificeras av programmeraren.

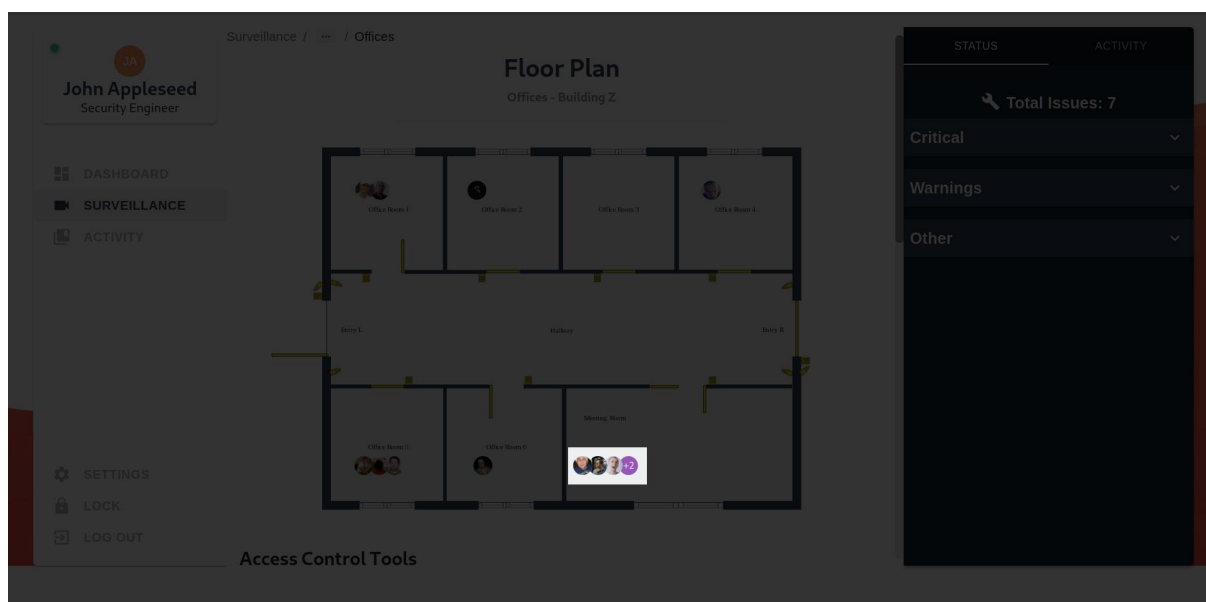


Figur 15. Andra versionen av planritningen som inkluderar avatars.

Inkluderingen av cirkelarna i planritningen har ett par nackdelar som att det ökar SVG-filen storlek, kan endast visa 3 + 1 avatars (den fjärde förklaras senare i kapitlet) och måste ritas in manuellt i planritningarna. Gällande filstorleken blev det återigen inga stor marginaler mellan planritningen som visas på figur 13 (99.3 kB) och den som visas på figur 15 (124 kB), alltså skillnaden blev 24.7 kB. Möjligheten att visa mer än en avatar för ett visst rum var ett svårt

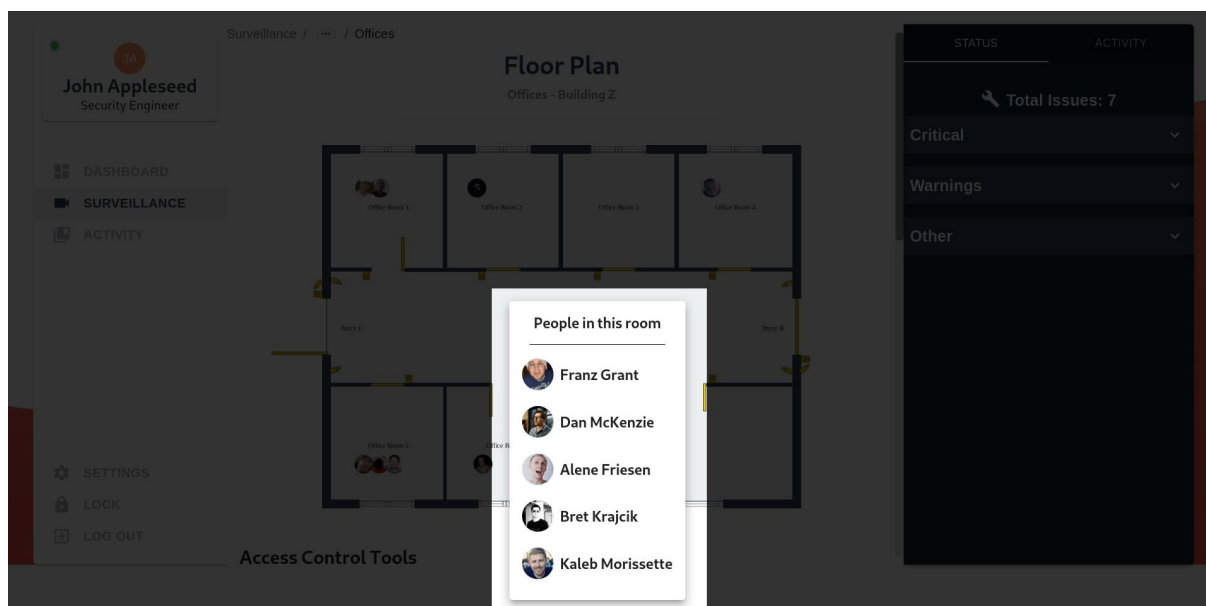
problem att lösa, eftersom cirklarna ritas in manuellt i planritningen så har man vid det här stadiet en möjligheten att bestämma hur många cirklar som ska visas per rum, examensarbetarna valde att maximalt 3 stycken per rum var tillräckligt för att gå vidare med utvecklingen dock kan detta givetvis variera för olika planritningar när applikationen är i bruk.

I planritningen som visas på figur 15 kan 3 tomma cirklar visas som tidigare förklarar kan innehålla en bild som dess bakgrund. Utöver dessa 3 finns en fjärde cirkel med beteckningen "+x". Detta är en metod som används i Material Design för att indikera att det finns fler avatars än det maximala som visas. Oftast är x en siffra som indikerar just hur många fler avatars som finns. I Applikationen implementerades detta av examensarbetarna samt möjligheten att klicka på dessa för att se hela listan med personer som finns i rummet. Figur 16 visar en grupp av avatars som överstiger det maximala.



Figur 16. En grupp av avatars som innehåller 5 avatars totalt.

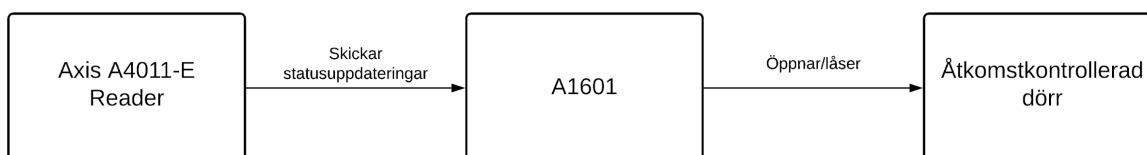
Resultatet av att klicka på en avatar som överstiger det maximala visas på figur 17. Detta är metoden som valts för att lösa problemet med att ett maximalt antal cirklar som kan visas väljs i förväg då själva planritningen ritas. Nackdelen med att i förväg bestämma max antal cirklar som ska visas är dock svår att komma undan, i figur 16 kan det ses att de resterande 2 avatarerna som inte ritas ut kan egentligen ritas ut, då det finns utrymme för det.



Figur 17. Popup-ruta som visas vid klick på avatar gruppen.

När möjligheten att manipulera alla enheter och avatrar via programkoden var på plats gavs examensarbetarna tillgång till Axis Communications interna API för att testa applikationen med verklig data, utöver det fick examensarbetarna en "AXIS A1601 Network Door Controller", en knappsats och en knapp för att simulera ett verkligt scenario.

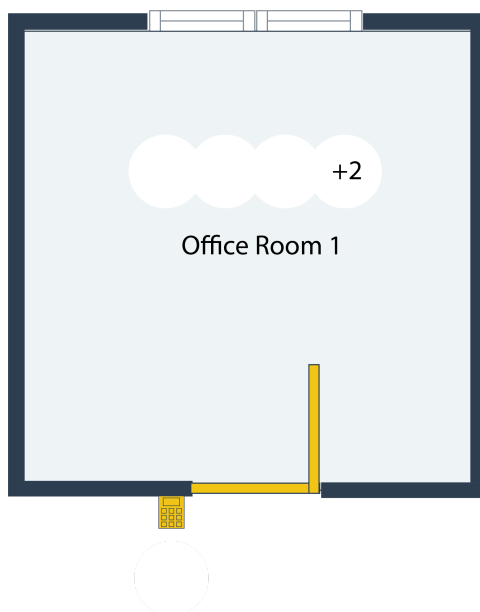
Genom användningen av Axis interna API kunde en anslutning till A1601 göras via programkoden i React-applikationen. Detta innebar att information gällande vilka personer som har interagerat med kortläsarna/knappsatserna kunde hämtas ut, utöver det kunde kommandon skickas till A1601 för att exempelvis låsa/låsa upp dörren. Exempel på ett scenario där en A1601 används i kombination med en kortläsare visas i figur 18. En Axis kortläsare (AXIS A4011-E Reader) används i detta scenariot där blippandet av en användares kort samt inmatningen av PIN-kod genererar statusuppdateringar som skickas till A1601. Via React-applikationen kan dessa uppdateringar hämtas från A1601 och evalueras. Evalueringen som görs i applikationen resulterar i att åtkomst beviljas och att dörren låses upp eller att åtkomst nekas och dörren förblir låst.



Figur 18. Exempel på hur samspelet mellan en kortläsare, en A1601 och en åtkomstkontrollerad dörr sker.

För att underlätta testningen av hårdvaran med applikationen skalades planritningens storlek ned. Den nedskalade versionen av planritningen visas i figur 19. Den nedskalade versionen

visar endast 1 rum. Nedskalning rekommenderades av handledaren för att underlätta testandet av A1601.



Figur 19. Nedskalad planritning som endast visar 1 rum.

Vid korrekt inmatning av PIN-kod måste resultatet av detta visas i applikationen, för att undvika HTTP-polling implementerades Websockets i applikationen. Detta innebar att då nya statusuppdateringar togs emot från kortläsaren kunde A1601 notifiera applikationen. Detta var sista steget i applikationen som väckte planritningen till liv. Genom att använda informationen som skickades från A1601 kunde examensarbetarna implementera funktionalitet som visar en användares position på planritningen, olika dörrars status (öppen/stängd) samt om åtkomst beviljades eller nekades för en specifik dörr.

4.4 Fas 4: Utvärdering av systemet

Under examensarbetets gång skedde det utvärderingar av alla faser i arbetet kontinuerligt. För examensarbetarna var det viktigt att varje del av arbetet fyllde ett syfte samt gav ett resultat. Analysering- samt utvärderingsprocessen har skett både informellt och formellt, beroende på vad som analyserades och vilken påverkan den hade för examensarbetets huvudmål och önskade resultat. Två av delarna som analyserades mer djupgående var UI/UX-prototypen samt den interaktiva prototypen.

UI/UX-prototypen analyserades informellt av framförallt examensarbetarna. Eftersom detta skedde tidigt under examensarbetet gång lades fokuset på att ge prototypen ett logiskt flöde samtidigt som den innehåller alla tänkta funktioner. Denna tidiga analyseringsprocess ansågs viktig för att upptäcka potentiella fel innan den interaktiva prototypen utvecklades, och på så sätt minska felkorrigering i senare stadier som kunde innebära stora kostnader i tid. Under denna analyseringsprocess var det viktigt för examensarbetarna att handledaren på Axis Communications gjorde en utvärdering. Även detta gjordes informellt genom en demo via

Microsoft Teams videosamtal. Handledarens återkoppling betraktades som värdefull information för arbetets fortlöpning och vidareutveckling eftersom Axis Communications är en nyckelintressent för examensarbetet.

När den interaktiva prototypen utvecklats analyserades samt utvärderades den både formellt och informellt. Den informella analysen utfördes främst av examensarbetarna och skedde parallellt med den interaktiva prototypens utveckling. Varje ny funktion som implementerades genomgick informella tester av examensarbetarna för att konstatera att funktionen uppfyllde dess tänkta syfte. Informella tester gjordes även i takt med att fler funktioner implementerades för att analysera hur funktionerna uppför sig i relation till varandra, detta för att tidigt i utvecklingen upptäcka om funktioner inte fungerade i kombination med varandra och då orsaken till detta.

I samband med att den interaktiva prototypen kopplades samman med hårdvaran genomfördes ett antal informella tester av examensarbetarna för att säkerställa att korrekta resultat genererades samt att komponenterna uppfyllde dess tänkta syfte.

I utvecklingen av den interaktiva prototypens sista stadie utfördes informella tester av hela applikationen. Detta skedde genom att användartesta varje funktionalitet som representerades som en task på scrumboarden.

4.5 Fas 5: Avslutning

Examensarbetet avslutades med en överlämning av den interaktiva prototypen och av inlämning av examensrapporten. Vid överlämningen av applikationen i slutet av examensarbetet var det viktigt att alla delar av applikationen följde med, samt att allt material som lånades av Axis Communications återlämnades (verktyg, komponenter etc.). Eftersom applikationen fanns tillgänglig på Axis Communications GitLab-sida, kunde företaget ta emot applikationen genom att ladda ner en klon av applikationens filer. Ett officiellt överlämnings-möte hölls för att säkerställa att överlämningen gjordes på korrekt sätt samt med alla parter kännedom och godkännande.

Utöver överlämningen till Axis Communications bestod avslutningen av inlämning av examensrapporten till handledare samt examinator på Lunds Universitet. Därefter förberedde examensarbetarna en presentation av arbetet samt för en opponering av arbetet.

4.6 Analys av arbetet med Scrum

Under examensarbetets gång har projektmodellen Scrum tillämpats för att strukturera samt följa examensarbetets faser. Som tidigare nämnt användes programmet Trello för att konstruera en Scrumboard som bland annat består av en backlog och en sprintlog. Arbetet har gått till på följande sätt att Tasks flyttats från backlogen och placerats i sprintlogen inför varje inledning av en sprint. Sedan tilldelas examensarbetarna tasks från sprintlogen och de implementeras i den ordning de är prioriterade. Huvudmålet med varje sprint har varit att utföra alla tasks innan sprinten avslutas. Om detta målet inte uppfylldes, det vill säga att alla tasks inte utfördes i tid, placerades de i nästkommande sprint's sprintlog.

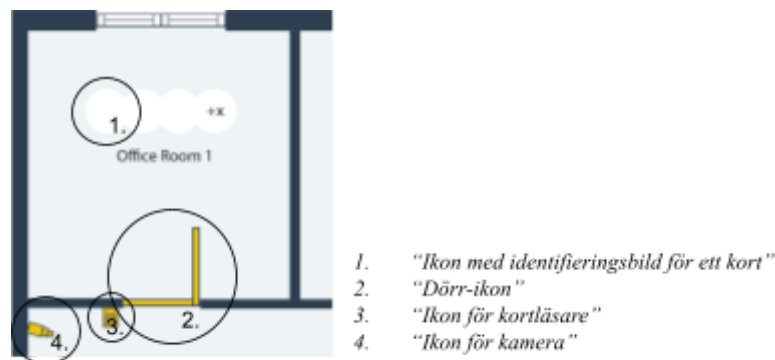
Slutligen kan det fastställas att samtliga tasks med prioriteringsnivå "hög prioritering" i backloggen utfördes samt genomgick granskning (främst i form av användartester) inom examensarbetets tidsbegränsning, det vill säga innan avslutningen samt inlämning av examensarbetet. Tasks med prioriteringsnivå "medel prioritering" utfördes samt genomgick granskning i stor utsträckning, med endast ett fåtal tasks utförda. Utöver detta utfördes en övervägande del av tasks med prioriteringsnivå "låg prioritering" där granskning gjordes i mån av tid.

5. Resultat

Detta examensarbete resulterade i en applikation som Axis Communications kan använda för åtkomstkontroll samt övervakning av flera ut- och ingångar. Denna slutprodukt är en interaktiv prototyp som anses vara en utökning av Axis Communications nuvarande applikation som övervakar endast en ut- och ingång.

I denna applikation finns möjligheten att, från en planritning som uppdateras kontinuerligt, välja en komponent (kortläsare, kamera, dörr etc.) och utföra handlingar baserat på vilken komponent som valdes. Exempel på dessa handlingar är bland annat att låsa och låsa upp dörrar, se status för en kamera eller se information om kortinnehavare. Komponenten väljs genom en vänsterklick på ikonerna på den interaktiva planritningen. För alla komponenter kan man se status samt annan information.

All information gällande komponenterna samt personerna som begär åtkomst kommer från Axis interna API. Det är möjligt att kontinuerligt uppdatera planritningen genom den aktiva Websockets anslutningen som finns när applikationen är i bruk.



Figur 20. Nyckelpunkter i den interaktiva planritningen.

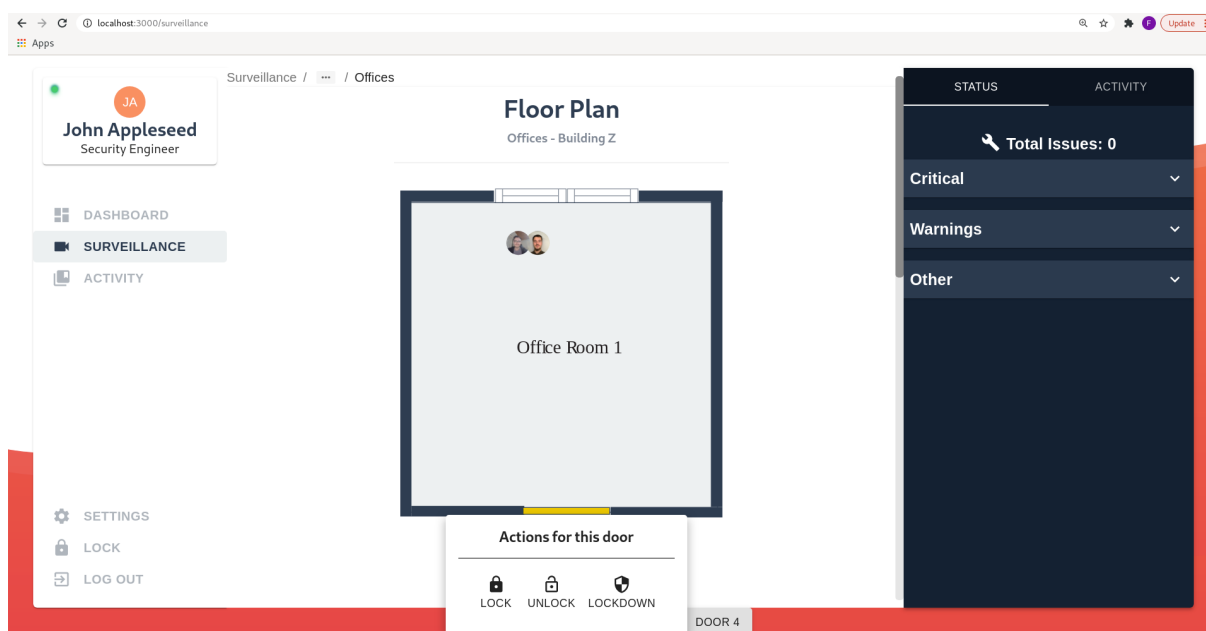
Det går även att granska de senaste korten som erhållit åtkomst till alla ut- och ingångar genom att använda kortläsaren. Informationen som visas i samband med detta avslöjar om kortet fått åtkomst eller ej, samt genom vilken metod kortet har fått åtkomst. Åtkomst kan erhållas på olika sätt, exempelvis kan åtkomst erhållas genom giltigt kort och giltigt kort samt korrekt tillhörande kod. Applikationen visar även om användaren stött på problem i samband med försök till att få åtkomst, som exempelvis om felaktig kod matats in i tidigare försök. Korten som precis erhållit åtkomst till ett rum visas i applikationen som en rund ikon med kortets identifierings-bild, placerad i rummet på planritningen (se figur 20).

Genom en vänsterklick på dörr-ikonerna på den interaktiva planritningen finns valmöjligheten att göra ett av följande:

1. Låsa dörren - I detta läge kan endast personer med kort öppna dörren genom att använda kortläsaren placerad närmast dörren.

2. Låsa upp dörren - I detta läge kan alla med eller utan kort öppna dörren.
3. Lockdown - I detta läge kan ingen med eller utan kort öppna dörren. Dörren i detta fallet förblir låst till dess att dörren låses upp igen via applikationen.

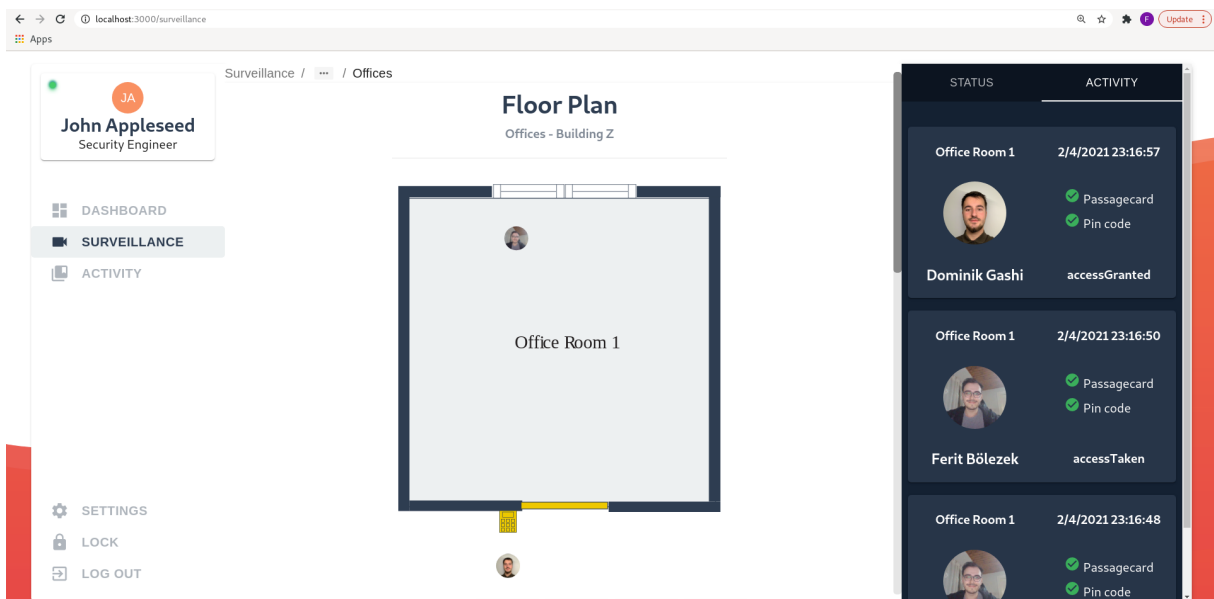
Figur 21 visar ett exempel på hur detta kan se ut i applikationen.



Figur 21. Dialogruta med "lock", "unlock" och "lockdown" presenteras när användaren klickar på dörren.

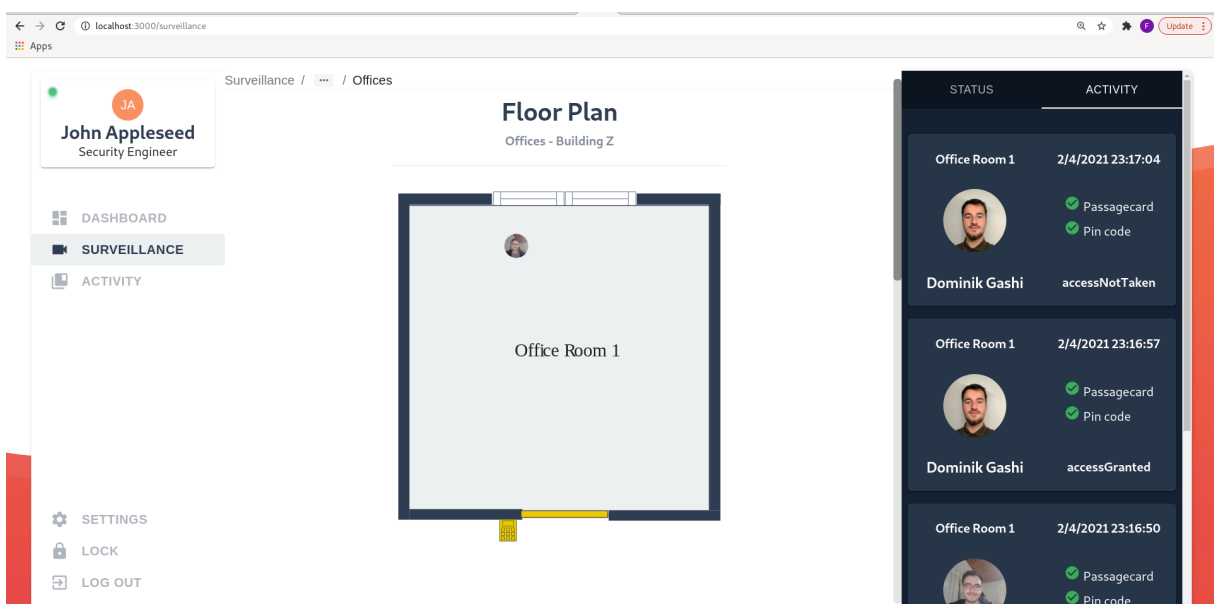
På högra sidan av applikationens fönster finns en tvådelad vy. Ena vyn visar status för hela planritningen och andra vyn visar aktivitet för hela planritningen. I aktivitetsvyn kan information om vilka kort som nyligen erhållit åtkomst till vilken ut- och ingång som helst granskas i ordning från senast erhållen åtkomst till tidigast erhållen åtkomst. Även information om hur åtkomsten erhållits samt vilka eventuella problem som uppkommit visas som beskrivet ovan.

Figur 22 illustrerar hur applikationen visar nyligen beviljade åtkomster till "Office Room 1". I figuren befinner sig en individ utanför dörren och har interagerat med kortläsaren samt fått sin begäran för åtkomst beviljat. Detta visas på högra sidan av applikationen.



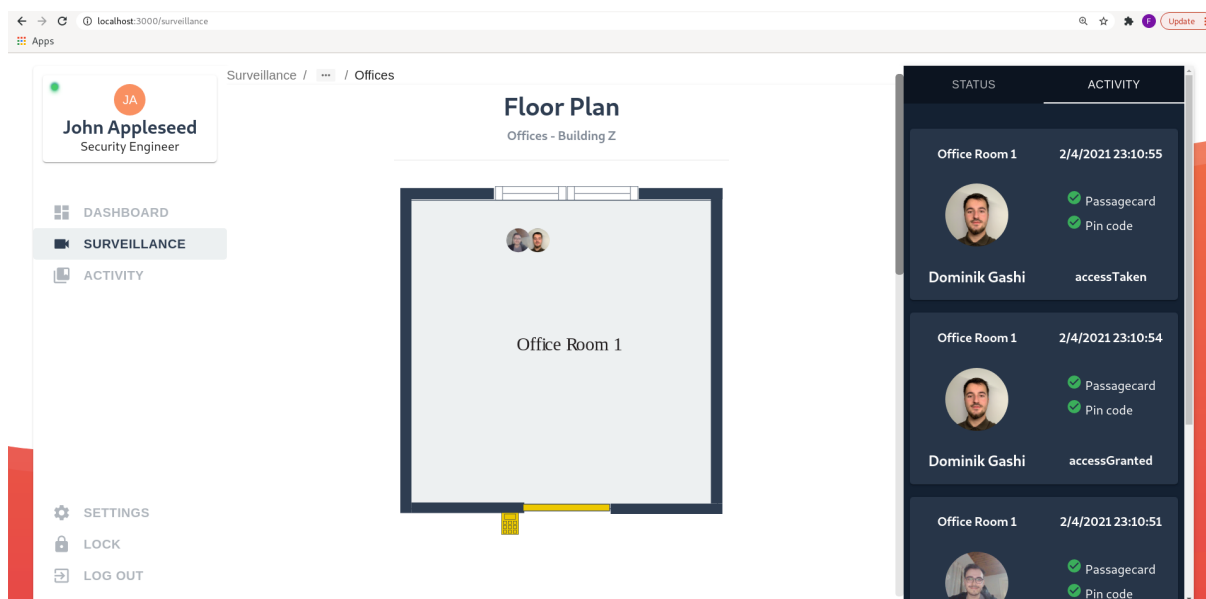
Figur 22. "Dominik Gashi" har fått åtkomst till rummet men har inte gått in i rummet ännu.

Figur 23 visar följden av figur 22, vilket är att individen som fått sin åtkomst beviljat har valt att inte öppna dörren.



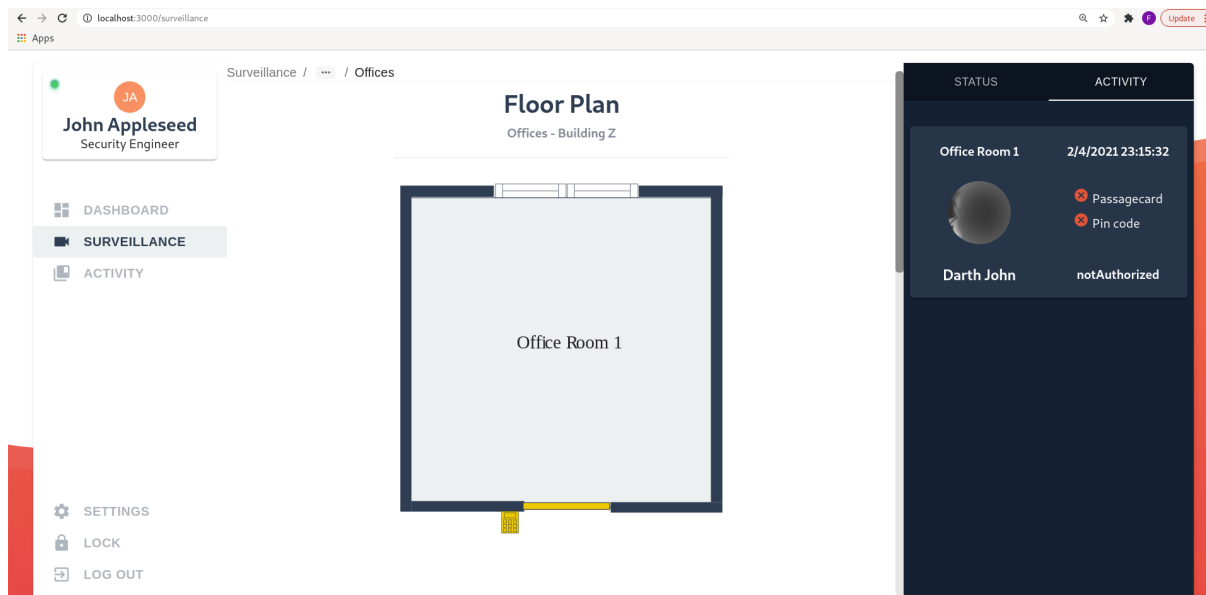
Figur 23. "Dominik Gashi" har fått åtkomst till rummet men gick aldrig in i rummet.

Figur 24 visar ett alternativ till figur 23, och är en annan följd av figur 22. I detta fall har individen valt att öppna dörren och kliva in i rummet. Individens bild visas nu i rummet (figur 24).



Figur 24. "Dominik Gashi" har fått åtkomst till rummet och har gått in i rummet.

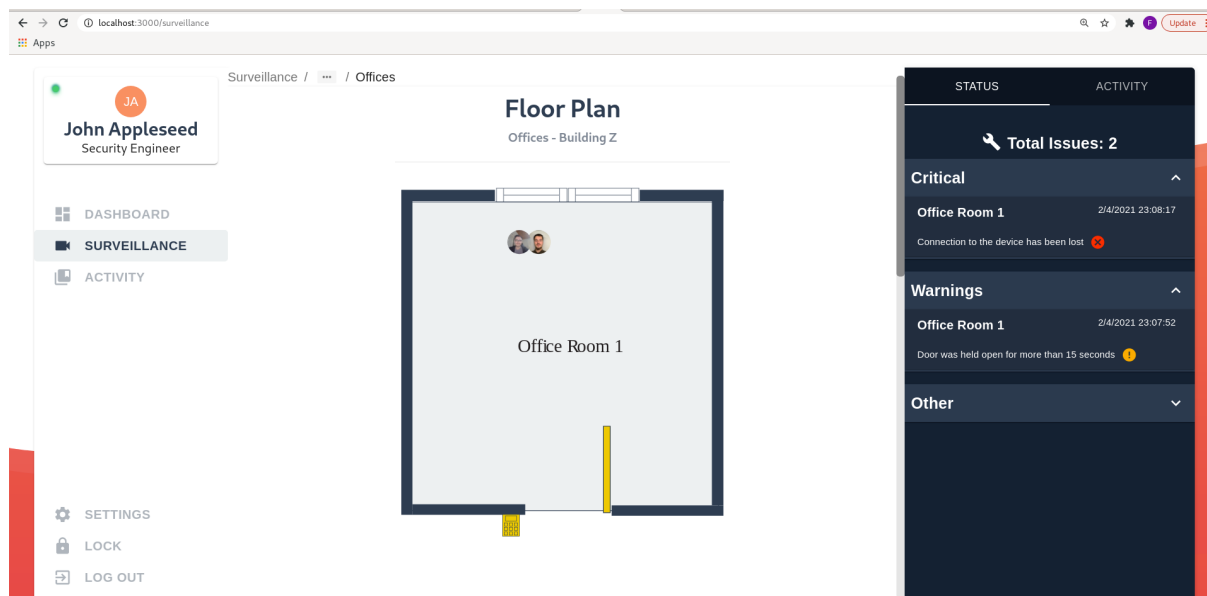
Figur 25 visar ett annat scenario där en person får sin åtkomst nekad. I dessa fall där personer inte borde få tillgång till ett rum, kommer de att nekas åtkomst, detta kommer också att visas i applikationens högra sida och till skillnad från personer som fått sin åtkomst beviljat så kommer dessa individer att ha en röd ikon vid "passagecard" och "pin code" samt ett statusmeddelande "notAuthorized".



Figur 25. "Darth John" har blivit nekad på sin begäran att få åtkomst till rummet.

I statusvyn visas statusinformation om komponenterna. Detta kan exempelvis vara om en kamera är avstängd eller en kortläsare ger felmeddelanden eller uppför sig på felaktigt sätt.

Figur 26 visar två nivåer av problem som kan uppstå i applikationen och hur dessa kategoriseras i applikationen. Beroende på hur kritiskt problemet är kan den antingen hamna under “kritisk”, “varning” eller “övrigt”. Under “kritisk” presenteras problem som är säkerhetskritiska vilket kan leda till intrång. Under “varning” presenteras problem som borde undersökas av operatören då de kan vara säkerhetskritiska. Under “övrigt” presenteras övriga problem, detta kan exempelvis vara att en individ inte lyckas med att få åtkomst och istället begär detta från operatören.



Figur 26. Statusvyn visar 2 “issues”.

6. Slutsats

Examensarbetets syfte var dels att undersöka existerande lösningar för åtkomstkontroll och dels att utveckla en prototyp som var både effektivare och mer tidsbesparande. Som tidigare nämnt erbjöd Axis Communications en applikation som tillåter övervakning av en dörr. Ur en kunds perspektiv kan detta anses ineffektivt då åtkomstkontrollerade miljöer kan bestå av mer än en dörr. Lösningen som presenteras i resultatkapitlet bidrar till den önskade effektiviseringen och tidsbesparingen som var syftet med examensarbetet, genom att den framförallt tillåter övervakning av flera dörrar samtidigt. Övervakningen i lösningen som presenterats möjliggörs genom en interaktiv planritning med ikoner för olika komponenter placerade i planritningen där operatören har en överblick över alla komponenter samtidigt och kan på ett tidseffektivt sätt komma åt funktioner samt se status för dessa komponenter.

Resultatet av detta examensarbete visar att ett system för övervakning av flera dörrar kan konstrueras som kan underlätta arbetsprocessen för operatörer genom att eliminera visualiseringen som användarna måste göra i ett system som inte erbjuder en visuell representation av en åtkomstkontrollerad miljö.

Med texten skriven ovan kan det konstateras att syftet samt målen för examensarbetet uppfyllts, även om det skett mindre förändringar beträffande funktionalitet från UI/UX-prototypen till det interaktiva prototypen i resultatet. Problemformuleringarna för examensarbetet samt hur de besvarats presenteras i kapitel 6.1.

6.1 Problemformulering

I kapitel 1 introducerades tre problemformuleringar dessa besvaras här.

1. Vilka nuvarande lösningar finns för åtkomstkontroll?

Under arbetets gång undersöktes lösningen som Axis Communications erbjuder, vilket var en webbapplikation som möjliggör övervakning av en dörr. Lösningen presenterades av handledaren på Axis Communications. Därefter undersöktes möjliga effektiviseringar vilket ledde till det andra problemet som behandlas i detta examensarbete.

2. Vilka åtgärder kan införas för att bidra till effektivisering av en redan implementerad övervakning av en åtkomstkontrollerad miljö?

Det nuvarande lösningen som presenterades av Axis Communications hade begränsningar som diskuterades av handledaren och examensarbetarna. Den största begränsningen var att övervakningen av flera dörrar i en åtkomstkontrollerad miljö inte kunde göras på ett intuitivt och översiktligt sätt. Detta diskuterades vidare med UX-designers och ett tydligt problem som kan uppstå är att det kan vara svårt för icke-tekniker att visualisera alla enheter inklusive deras positionering i den åtkomstkontrollerade miljön.

Lösningen som presenterades i detta examensarbete är att införa en planritning som kan kontinuerligt uppdateras med individers och enheters positionering, statusinformation för enheterna samt ge användaren en översikt av alla enheter i den åtkomstkontrollerade miljön.

3. Vad krävs för att den föreslagna lösningen skall komma att användas i framtiden?

Problemet med den föreslagna lösningen upptäcktes av examensarbetarna under utvecklingens gång och det är att planritningarna måste ritas och anpassas för varje kunds unika behov. Svårigheten med detta diskuterades även med UX-designers på Axis Communication, där de hade stött på liknande lösningar som erbjuds till specifika företag och deras unika behov. Möjligheten att introducera ytterligare system som förenklar ritandet och anpassningen av planritningarna samt integrering med applikationen kan möjligtvis leda till användningen av lösningen. Att undersöka denna möjlighet rekommenderades av en UX-designer på Axis Communications dock faller detta utanför syftet med examensarbetet.

Ett annat problem som diskuterades med UX-designers på Axis Communications och som måste undersökas vidare är att systemet möjliggör precis lokalisering av individer som befinner sig i den åtkomstkontrollerade miljön. Detta är en fråga gällande integritet som måste undersökas om prototypen ska realiseras och användas. Även detta faller utanför syftet med examensarbetet, examensarbetarna har valt att inte gå vidare med undersökningar inom detta.

6.2 Reflektion över etiska aspekter

Under detta examensarbete tillkom etiska aspekter som examensarbetarna behövde ta hänsyn till. Som ovan nämnt upptäcktes ett problem när det gäller lokalisering av individer genom att en ikon med identifieringsbild placeras i det rum de begär åtkomst till. I intervju med UX-designers på Axis Communications diskuterades huruvida det är etiskt rätt att konstant veta positionen på alla kortinnehavare. Eftersom detta etiska problem handlar om individers integritet krävs att vidare undersökningar görs för att hitta ett alternativ som inte strider mot etiska och sociala normer. En tänkbar lösning skulle vara att inte inkludera identifieringsbild, istället skulle endast antal personer i varje rum inkluderas. Detta innebär att enskilda individer inte lokaliseras. Med tanke på att syftet med applikationen är åtkomstkontroll behöver operatören samt applikationen kunna identifiera individen som begär åtkomst för att kunna fatta ett beslut. Därför skulle den tänkta lösningen inte visa sig vara effektiv i praktiken.

6.3 Framtida utvecklingsmöjligheter

Utifrån informationsinsamlingen samt eliciteringen som skedde utvecklades UI/UX-prototypen, designen för denna prototyp var för ett helt system som stödjer exempelvis inloggning, dashboard, filtrering, låsskärm mm. Detta är funktionalitet som kan vidareutvecklas i framtiden för att erbjuda användaren en helhetslösning.

Ytterligare en funktionalitet som diskuterades med handledaren på Axis och som var tänkt att implementeras var ansiktsigenkänning. Detta gör det möjligt att säkerställa att en individ med ett giltigt passerkort som beviljas åtkomst verkligen är ägaren av passerkort. Detta innebär att ett foto skulle tas på personen som begär åtkomst och via användning av datorseende kan en jämförelse göras mellan det tagna fotot och fotot på passerkortet. Denna funktionalitet utvecklades inte på grund av brist på tid.

Som tidigare nämnt är det svårt att möjliggöra den presenterade lösningen i praktiken på grund av anpassningen som måste göras för olika kunder. Detta innebär att lösningen kommer att kräva nya ritningar för varje kund. För att lösa detta kan det undersökas om det finns en plausibel lösning som förenklar ritandet av nya planritningar.

Utöver det finns det möjligheter att förbättra UI/UX-upplevelsen genom användartester med potentiella användare och förbättring av prestandan i applikationen.

7. Terminologi

1. DOM - Document Object Model

Är ett gränssnitt som presenterar HTML-dokument i ett trädstruktur där varje nod i trädet är en element i dokumentet.

2. Applikation/interaktiv prototyp

I denna rapport är användningen av “applikation” och “interaktiv prototyp” utbytbara. Båda refererar till samma sak, vilket är React-applikationen som har utvecklats för Axis Communications.

3. Task

En “task” refererar till ett kort på scrumboarden med en uppgift, vilket är en funktionalitet som måste utvecklas för applikationen

4. Snake case

Är en stil där mellanslag i kod ersätts med “_”, detta innebär att exempelvis “this is a variable” blir “this_is_a_variable”.

5. Avatar

Är en bild på en person, kan ses som en profilbild.

6. Dashboard

Är en sida i en applikation som innehåller viktig information för läsaren.

7. Informell test

Detta innebär att testandet inte har utförts utifrån någon testspecifikation och själva testerna har inte heller dokumenteras.

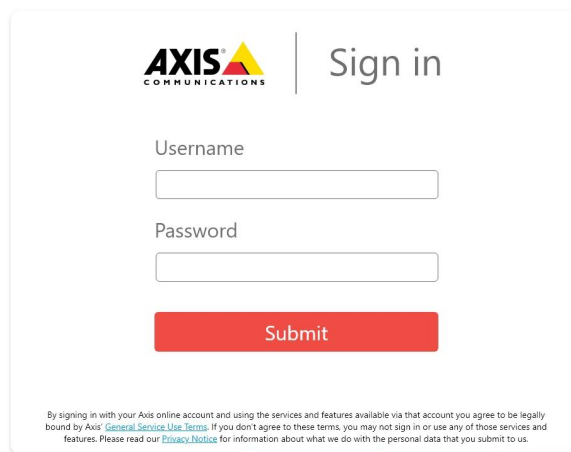
8. Källförteckning

- [1] W.Stallings, L. Brown, Computer Security: Principles and Practice, 4th ed. Harlow, UK: Pearson, 2018 pp 128-131
- [2] Axis, “About Axis”, *axis.com*, [Online]. Available: <https://www.axis.com/sv-se/about-axis>. [Accessed: Feb. 11, 2021]
- [3] React, “A JavaScript library for building user interfaces”, *reactjs.org*, [Online]. Available: <https://reactjs.org/>. [Accessed: Feb 19, 2021]
- [4] React, “Introducing JSX”, *reactjs.org*, [Online]. Available: <https://reactjs.org/docs/introducing-jsx.html>. [Accessed: Feb 19, 2021]
- [5] Babel, “What is Babel?”, *babeljs.io*, [Online], Available: <https://babeljs.io/docs/en/>. [Accessed: Feb 19, 2021]
- [6] React, “State and Lifecycle”, *reactjs.org*, [Online], Available: <https://reactjs.org/docs/state-and-lifecycle.html>. [Accessed Feb. 20, 2021]
- [7] React, “Components and Props”, *reactjs.org*, [Online]. Available: <https://reactjs.org/docs/components-and-props.html>. [Accessed Feb. 20, 2021]
- [8] React, “Lifting State Up”, *reactjs.org*, [Online]. Available: <https://reactjs.org/docs/lifting-state-up.html>. [Accessed Feb. 20, 2021]
- [14] Google, “Get started”, *material.io*, [Online]. Available: <https://material.io/resources/get-started#design>. [Accessed Feb. 21, 2021]
- [15] MDN Open Source Contributors, “Getting Started with SVG”, *developer.mozilla.org*, [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/SVG>. [Accessed Feb. 21, 2021]
- [16] MDN Open Source Contributors, “Document Object Model (DOM)”, *developer.mozilla.org*, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model#svg_interfaces. [Accessed Feb. 21, 2021]
- [17] MDN Open Source Contributors, “SVG Event Attributes”, *developer.mozilla.org*, [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/Events>. [Accessed Feb. 21, 2021]
- [18] Scrum, “What is Scrum?”, *scrum.org*, [Online]. Available: <https://www.scrum.org/resources/what-is-scrum>. [Accessed Mar. 19, 2021]
- [19] Discord, “What makes Discord different?”, *Discord.com*, [Online]. Available: <https://discord.com/why-discord-is-different>. [Accessed Mar. 19, 2021]
- [20] Trello, “About Trello”, *Trello.com*, [Online]. Available: <https://trello.com/about>. [Accessed Mar. 19 2021]
- [21] Microsoft Teams, “Microsoft Teams”, *Microsoft.com*, [Online]. Available: <https://www.microsoft.com/en-us/microsoft-teams/group-chat-software>. [Accessed Mar. 19, 2021]
- [22] Zoom, “Zoom”, *zoom.us*, [Online]. Available: <https://zoom.us/>. [Accessed Mar. 19, 2021]
- [23] S. Lauesen, Software Requirements: Styles and Techniques, 1st ed. UK: Pearson, 2002
- [24] OpenJS Foundation, “About Node.js”, *nodejs.org*, [Online]. Available: <https://nodejs.org/en/about/>. [Accessed Mar. 22, 2021]
- [25] W3Schools, “Node.js Introduction”, *w3schools.com*, [Online]. Available: https://www.w3schools.com/nodejs/nodejs_intro.asp. [Accessed Mar. 22, 2021]

- [26] I. Fette, Google, Inc., A. Melnikov, Isode Ltd., “The WebSocket Protocol”, *tools.ietf.org*, [Online]. Available: <https://tools.ietf.org/html/rfc6455>. [Accessed Mar. 22, 2021]
- [27] Material-UI, “Material-UI”, *material-ui.com*, [Online]. Available: <https://material-ui.com/>. [Accessed Mar. 22, 2021]
- [28] Adobe XD, “Design like you always imagined”, *adobe.com*, [Online]. Available: <https://www.adobe.com/se/products/xd.html>. [Accessed Apr. 2, 2021]
- [29] Visual Studio Code, “Code editing. Redefined.”, *code.visualstudio.com*, [Online]. Available: <https://code.visualstudio.com/>. [Accessed Apr. 10, 2021]
- [30] Create React App, “Create React App”, *github.com*, [Online]. Available: <https://github.com/facebook/create-react-app>. [Accessed Apr. 18, 2021]
- [31] NPM, “Build amazing things”, *npmjs.com*, [Online]. Available: <https://www.npmjs.com/>. [Accessed Apr. 18, 2021]
- [32] GitLab, “GitLab is the open DevOps platform”, *about.gitlab.com*, [Online]. Available: <https://about.gitlab.com/>. [Accessed Apr 18, 2021]
- [33] Pierre Carbonnelle, “Top IDE index”, *pypl.github.io*, [Online]. Available: <https://pypl.github.io/IDE.html>. [Accessed Apr 22, 2021]
- [34] React, “Create a New React App”, *react.js*, [Online]. Available: <https://reactjs.org/docs/create-a-new-react-app.html>. [Accessed Apr. 22, 2021]
- [35] Adobe Illustrator, “Adobe Illustrator”, *adobe.com*, [Online]. Available: <https://www.adobe.com/products/illustrator.html>. [Accessed Apr. 23, 2021]
- [36] Hugo Zapata, “react-svgmt”, *github.com*, [Online]. Available: <https://github.com/hugozap/react-svgmt>. [Accessed Apr. 26, 2021]
- [37] Axis Communications, “AXIS A1601 Network Door Controller”, *axis.com*, [Online]. Available: <https://www.axis.com/products/axis-a1601>. Available: [Accessed Apr. 26, 2021]
- [38] Kent Beck, Ken Schwaber, Jeff Sutherland, “Manifesto for Agile Software Development”, *agilemanifesto.org*, [Online]. Available: <http://agilemanifesto.org/iso/en/manifesto.html>. Available: [Accessed. May. 17, 2021]

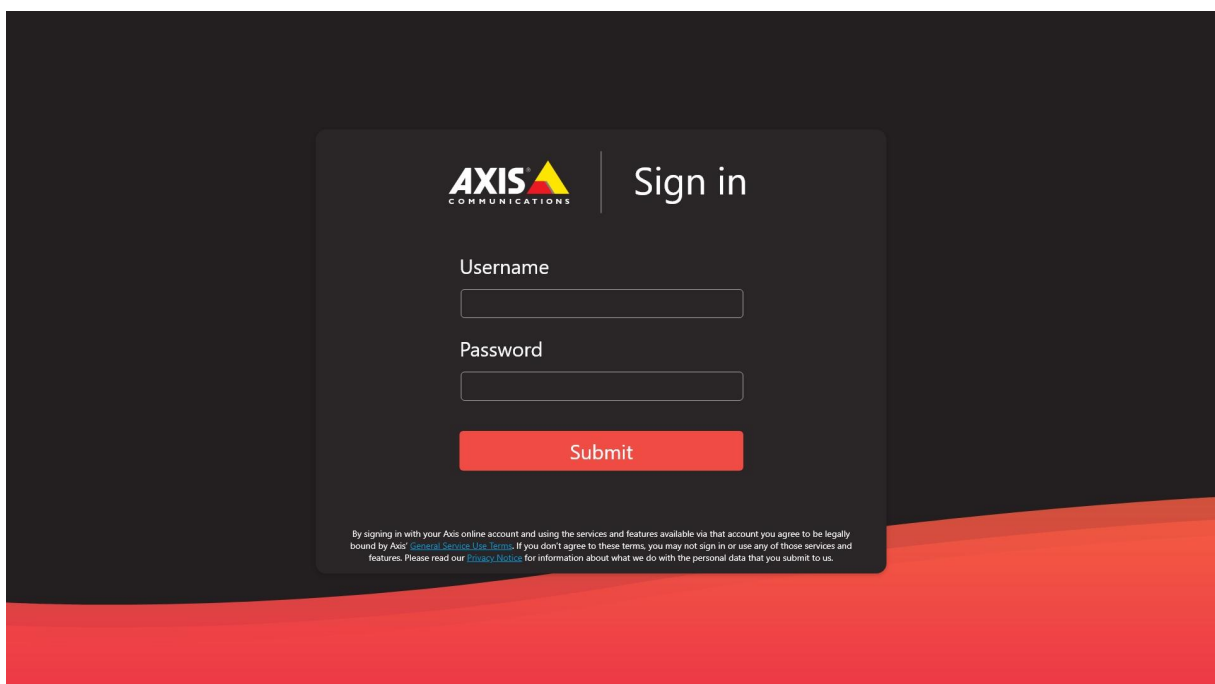
9. Appendix

9.1 UI/UX Prototyp



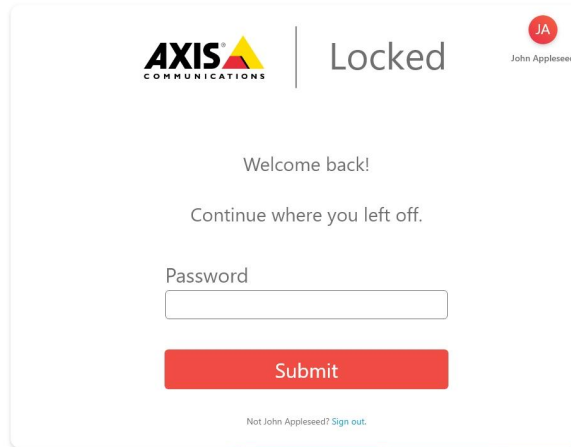
The image shows a light-themed sign-in form prototype. At the top left is the 'AXIS COMMUNICATIONS' logo. To its right is the text 'Sign in'. Below the logo and text are two input fields: 'Username' and 'Password'. A red 'Submit' button is positioned below the password field. At the bottom of the form, there is a small line of text: 'By signing in with your Axis online account and using the services and features available via that account you agree to be legally bound by Axis' [General Service Use Terms](#). If you don't agree to these terms, you may not sign in or use any of those services and features. Please read our [Privacy Notice](#) for information about what we do with the personal data that you submit to us.'

A.1 Prototyp av inloggningsidan (ljus läge).



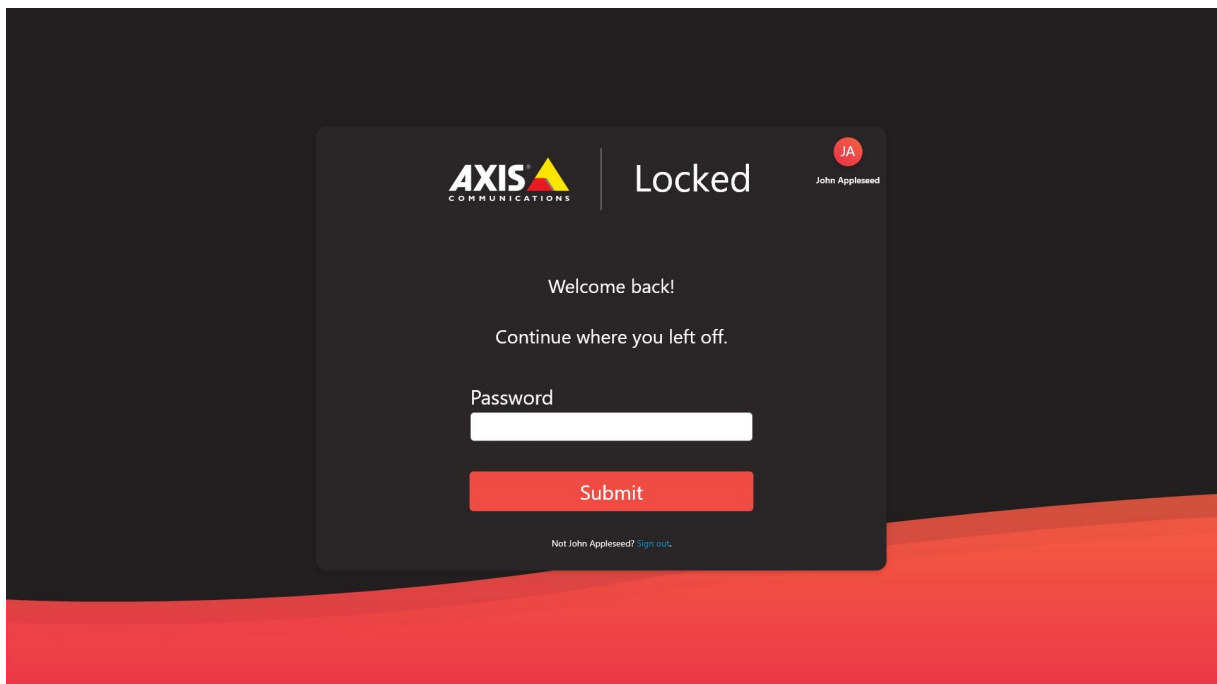
The image shows a dark-themed sign-in form prototype. At the top left is the 'AXIS COMMUNICATIONS' logo. To its right is the text 'Sign in'. Below the logo and text are two input fields: 'Username' and 'Password'. A red 'Submit' button is positioned below the password field. At the bottom of the form, there is a small line of text: 'By signing in with your Axis online account and using the services and features available via that account you agree to be legally bound by Axis' [General Service Use Terms](#). If you don't agree to these terms, you may not sign in or use any of those services and features. Please read our [Privacy Notice](#) for information about what we do with the personal data that you submit to us.'

A.1.1 Prototyp av inloggningssidan (mörkt läge).



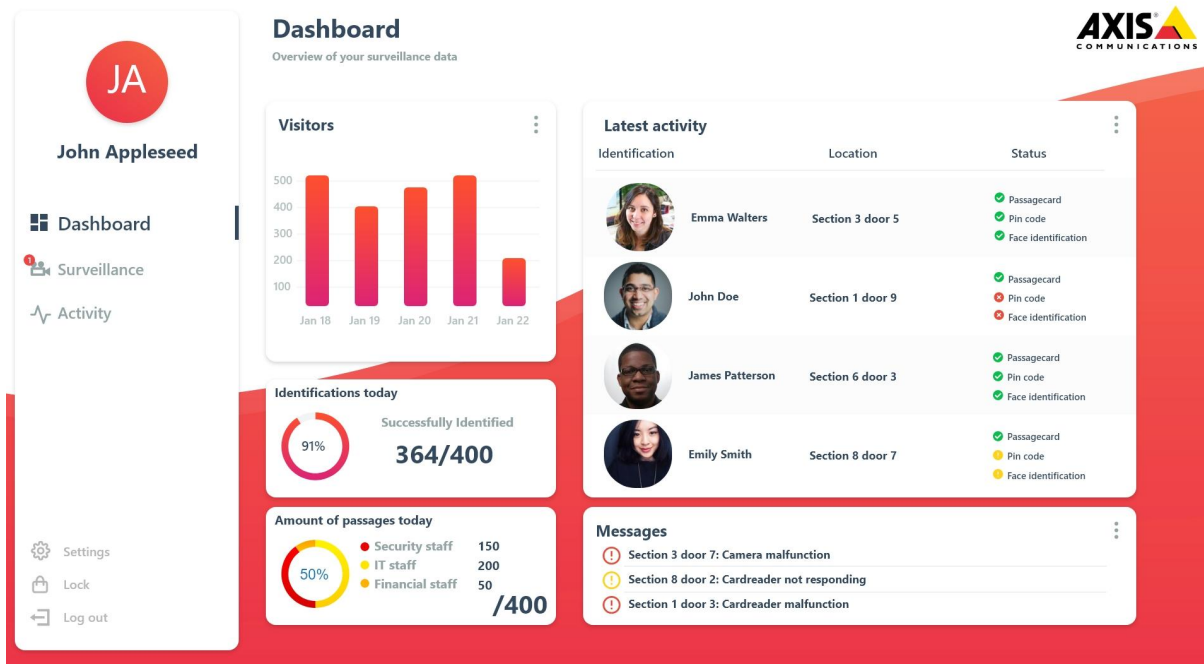
A prototype of a login form in a light theme. The form is white with a thin border and is centered on a white background. At the top left is the 'AXIS COMMUNICATIONS' logo. To its right is the word 'Locked' in a large, bold font. Further right is a user profile icon with the initials 'JA' and the name 'John Appleseed'. Below the logo and 'Locked' text, the text 'Welcome back!' and 'Continue where you left off.' is displayed. A 'Password' label is positioned above a white input field. Below the input field is a red 'Submit' button. At the bottom of the form, there is a small link: 'Not John Appleseed? Sign out.'

A.2 Prototyp av låsskärmen (ljus läge).



A prototype of a login form in a dark theme. The form is dark gray with a thin border and is centered on a dark gray background. At the top left is the 'AXIS COMMUNICATIONS' logo. To its right is the word 'Locked' in a large, bold font. Further right is a user profile icon with the initials 'JA' and the name 'John Appleseed'. Below the logo and 'Locked' text, the text 'Welcome back!' and 'Continue where you left off.' is displayed. A 'Password' label is positioned above a white input field. Below the input field is a red 'Submit' button. At the bottom of the form, there is a small link: 'Not John Appleseed? Sign out.'

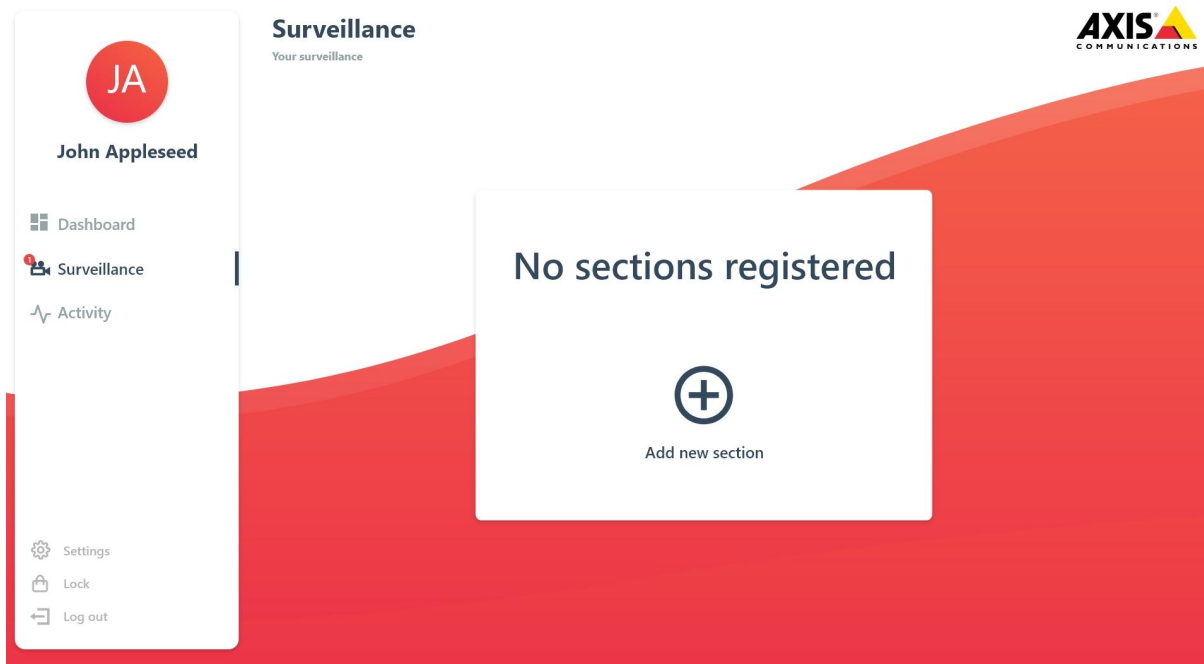
A.2.1 Prototyp av låsskärmen (mörkt läge).



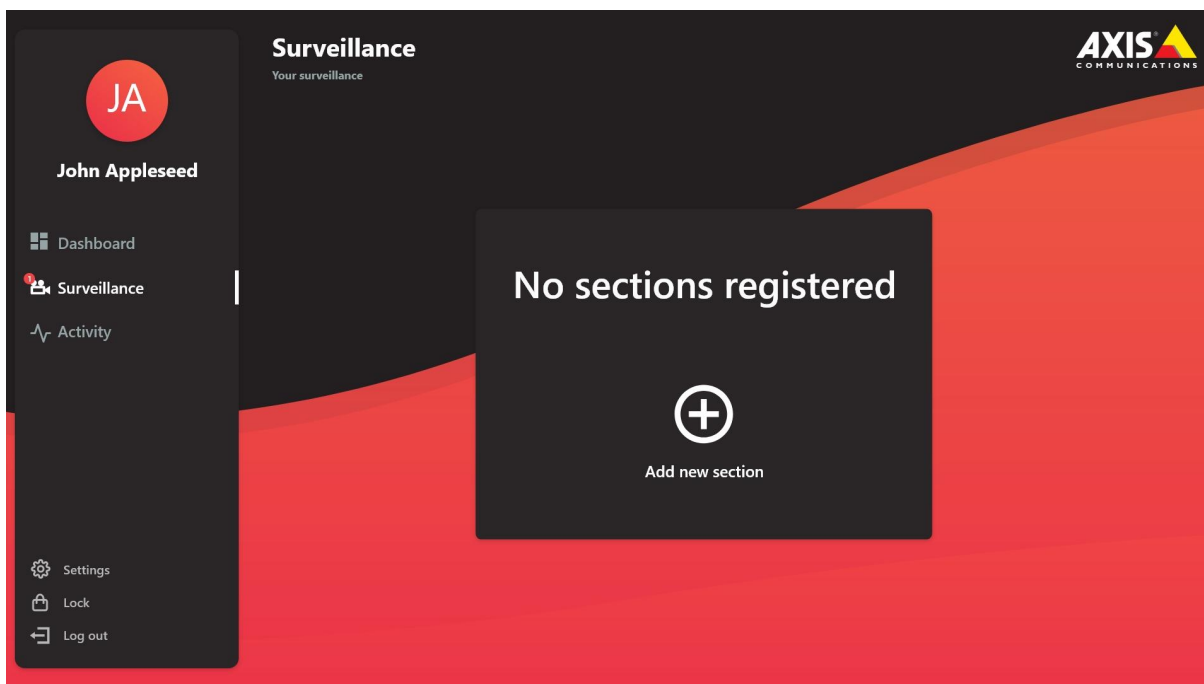
A.3 Prototyp av Dashboard-sidan (ljusst läge).



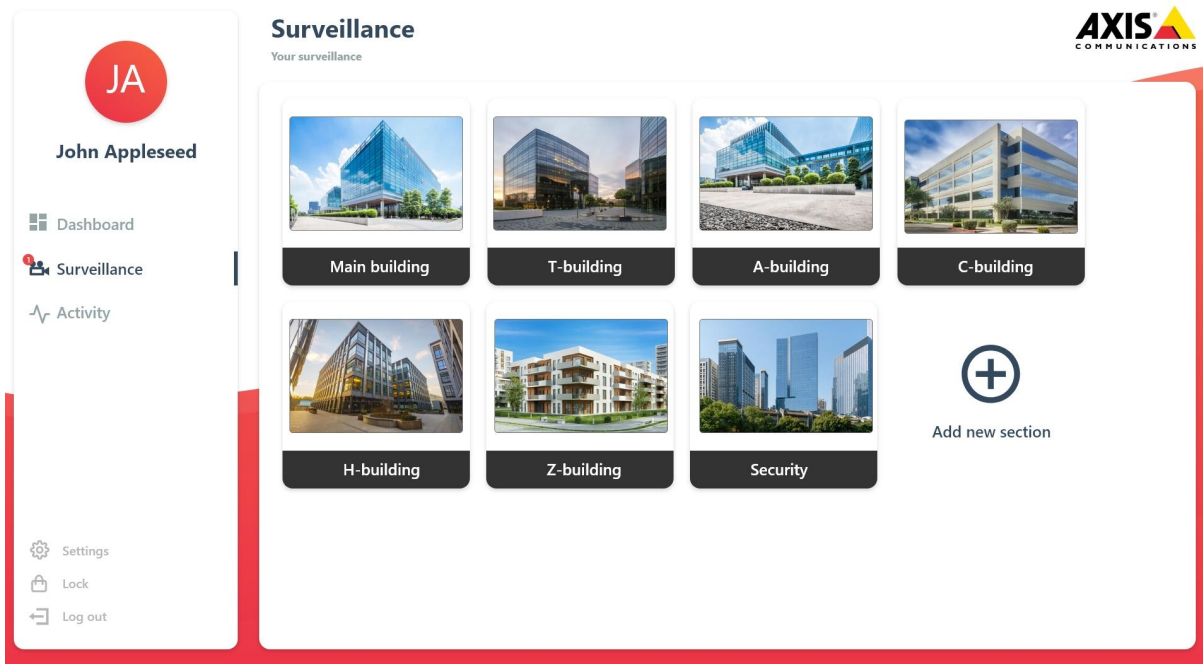
A.3.1 Prototyp av Dashboard-sidan (mörkt läge).



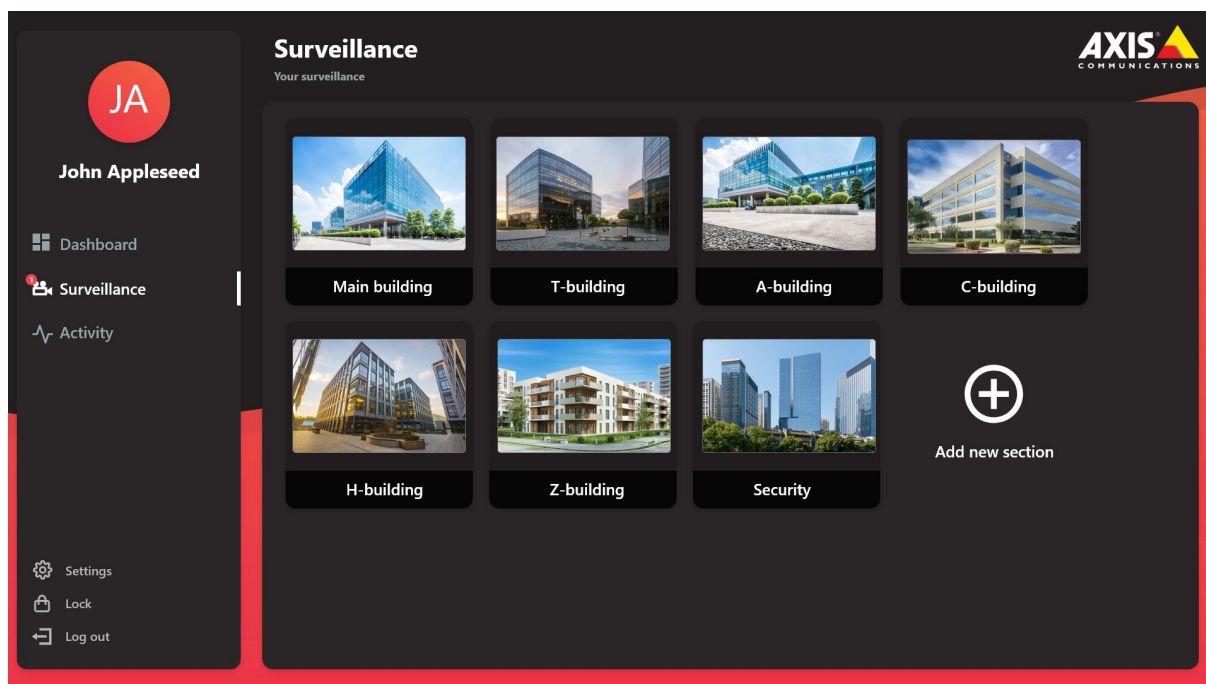
A.4 Prototyp av övervaknings-sidan (utan zoner eller sektioner) (ljus läge).



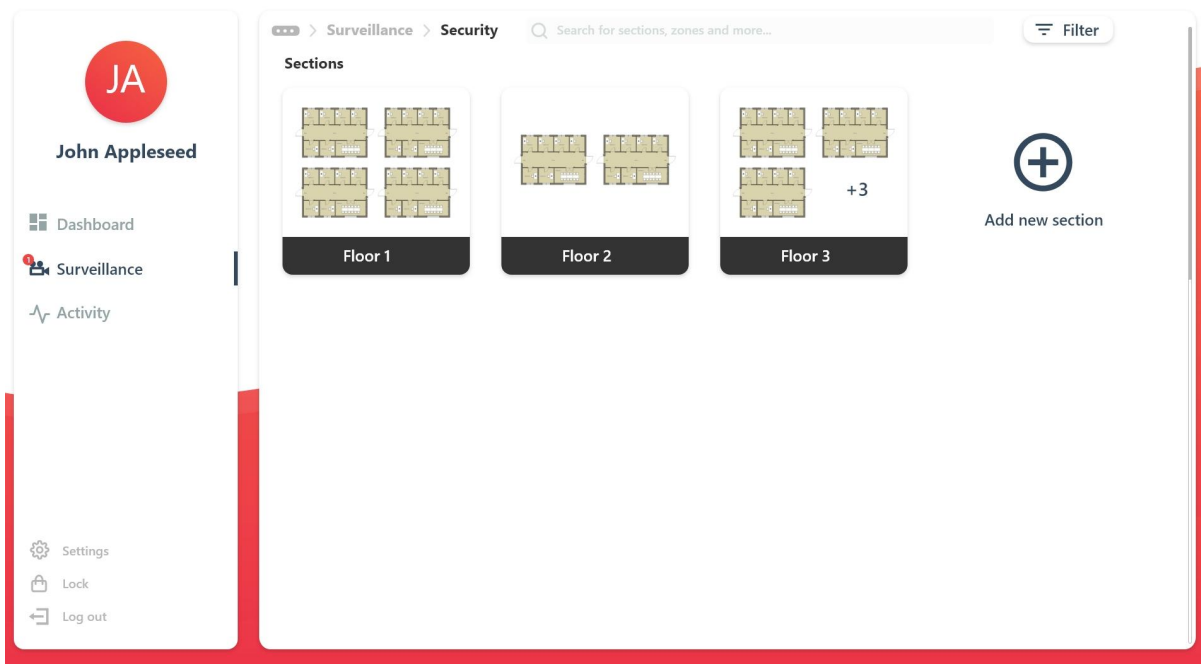
A.4.1 Prototyp av övervaknings-sidan (utan zoner eller sektioner) (mörkt läge).



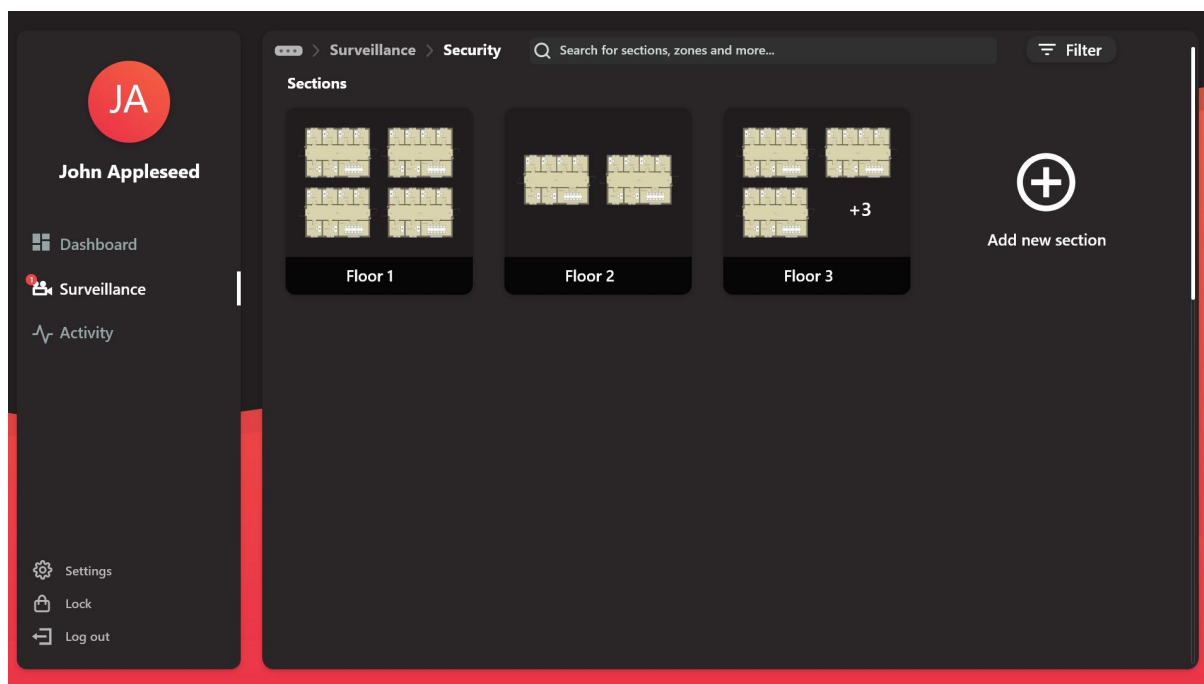
A.5 Prototyp av övervaknings-sidan (med zoner eller sektioner) (ljus läge).



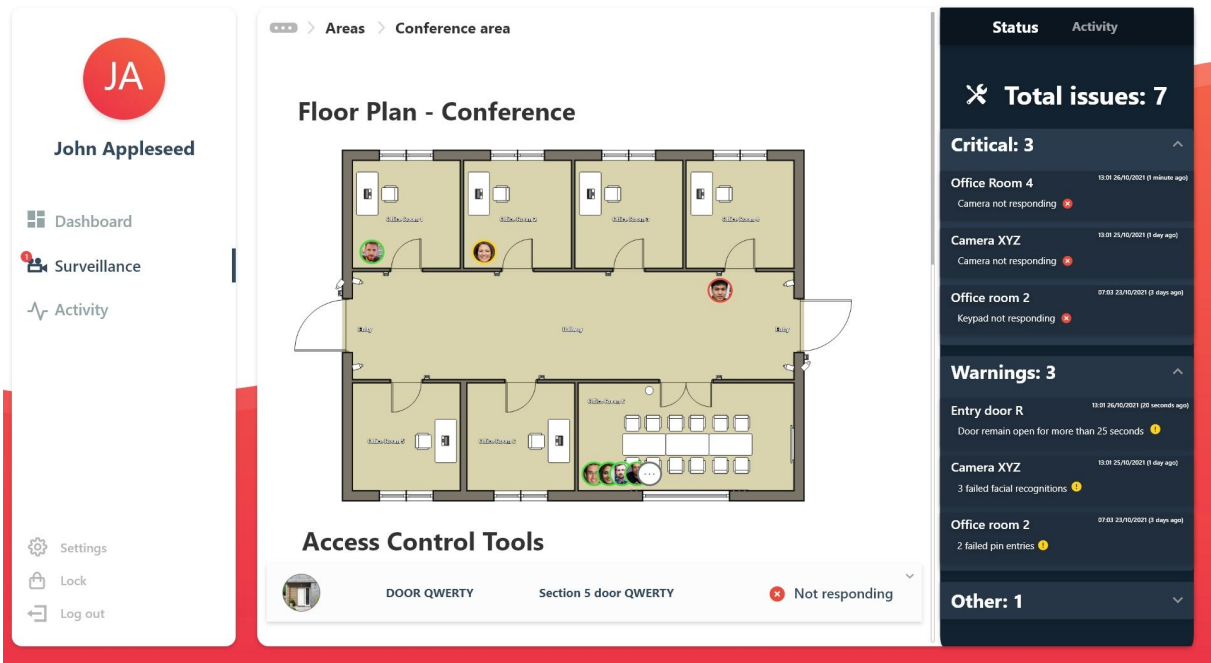
A.5.1 Prototyp av övervaknings-sidan (med zoner eller sektioner) (mörkt läge).



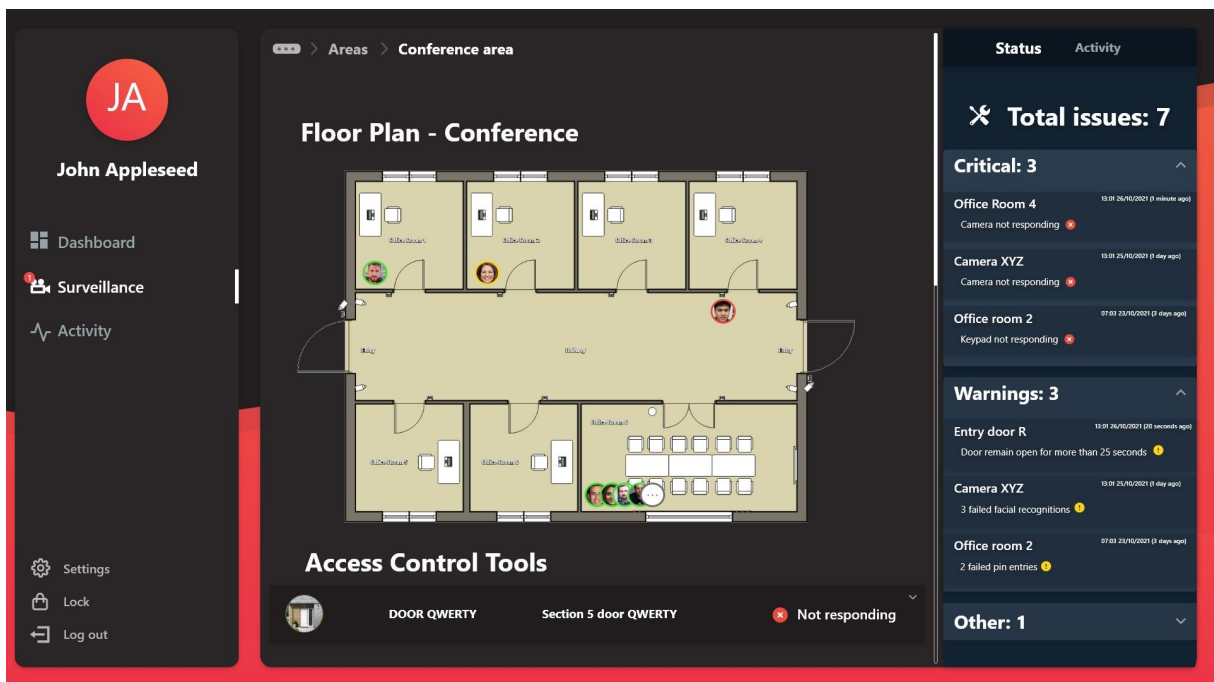
A.6 Prototyp av övervaknings-sidan för zoner med dess tillhörande planritningar (ljus läge).



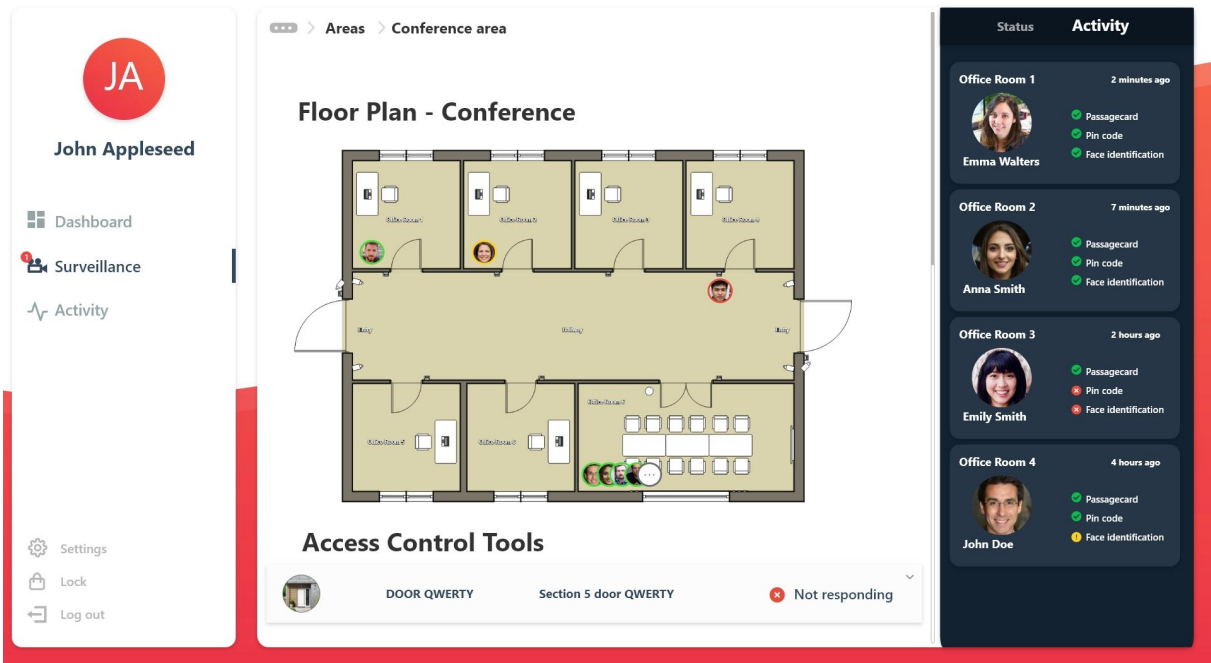
A.6.1 Prototyp av övervaknings-sidan för zoner med dess tillhörande planritningar (mörkt läge).



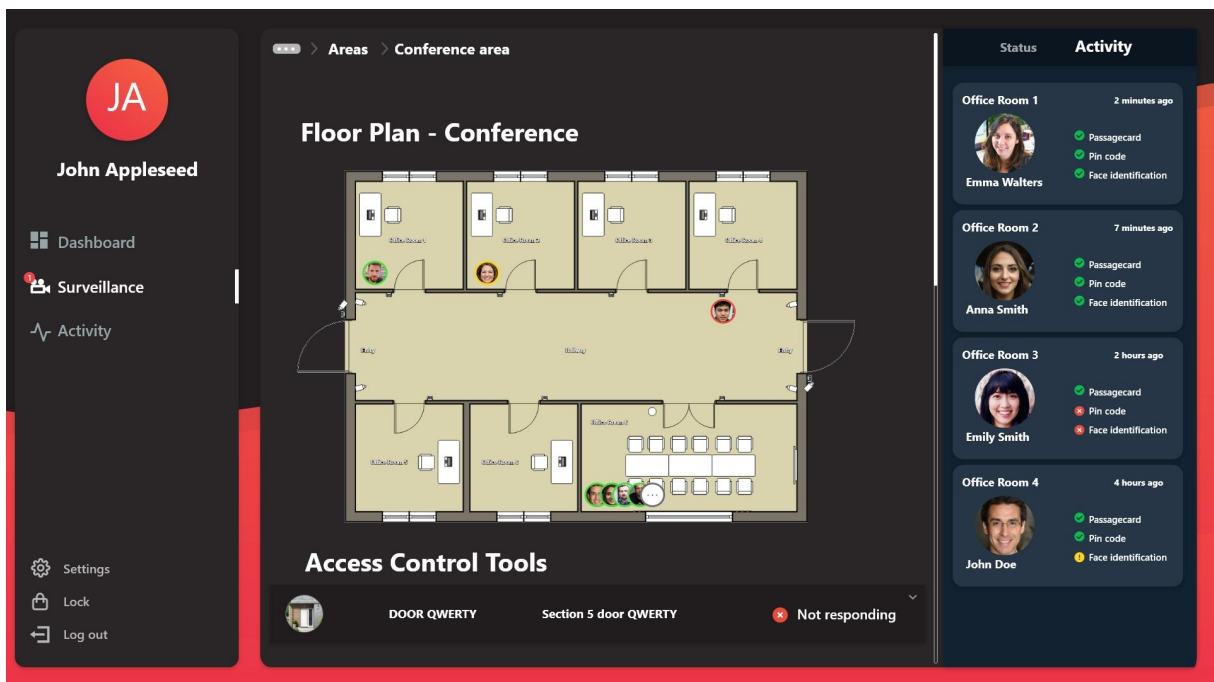
A.7 Prototyp av övervaknings-sidan för en vald planritning samt dess status på höger sida (ljus läge).



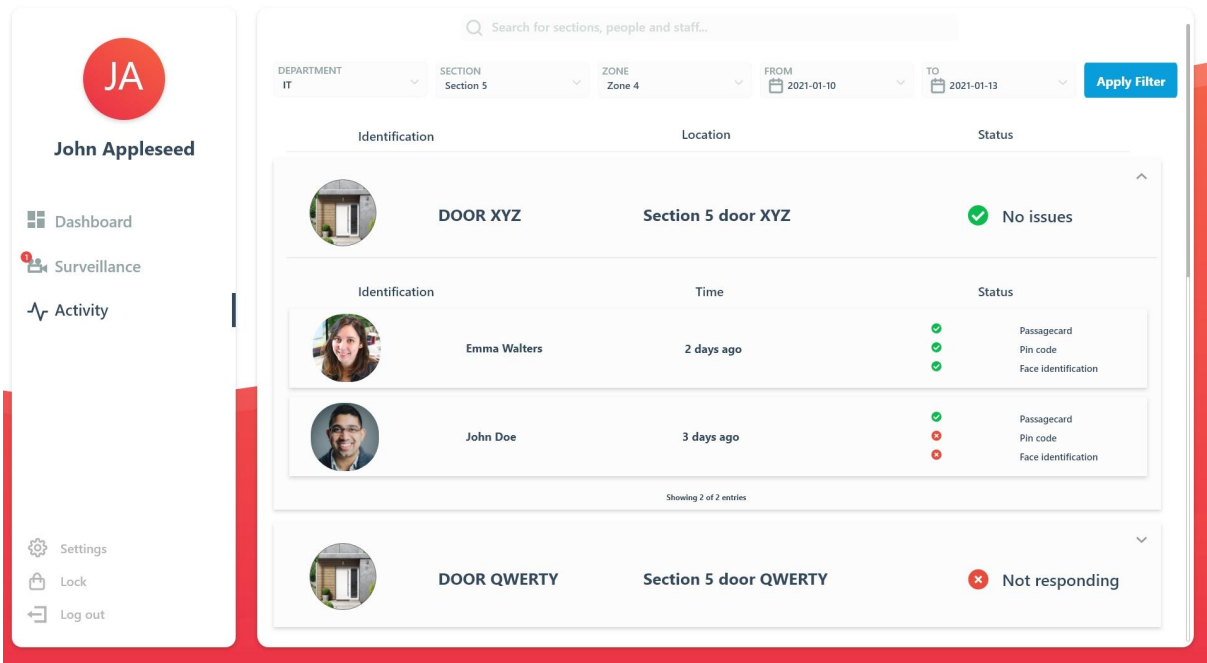
A.7.1 Prototyp av övervaknings-sidan för en vald planritning samt dess status på höger sida (mörkt läge).



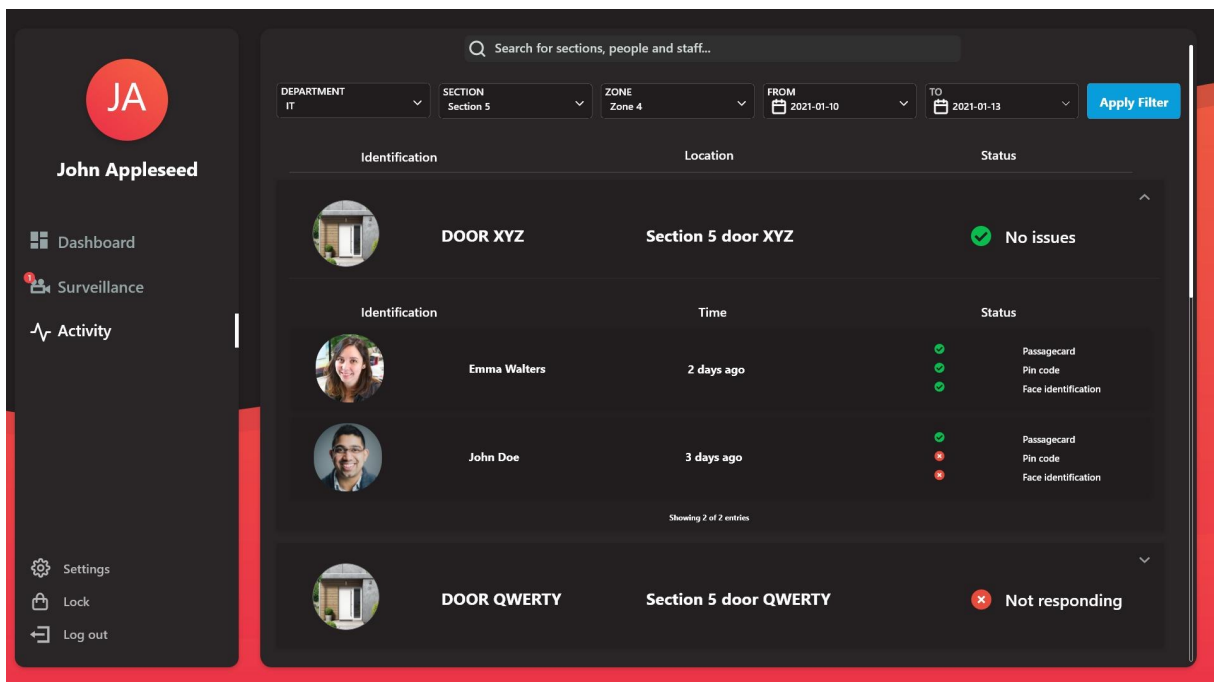
A.8 Prototyp av övervaknings-sidan för en vald planritning samt dess aktivitet på höger sida (ljust läge).



A.8.1 Prototyp av övervaknings-sidan för en vald planritning samt dess aktivitet på höger sida (mörkt läge).



A.9 Prototyp av aktivitets-sidan för alla enheter (ljus läge).



A.9.1 Prototyp av aktivitets-sidan för alla enheter (mörkt läge).

9.2 Intervju med Axis UX-designers

Nedan presenteras frågorna som ställdes till Axis UX-designers, svaren har sammanfattats.

Det är tänkt att vår prototyp ska användas av operatörer som övervakar flera dörrar (en sorts utökning av 1 dörr systemet som Axis utvecklat), vilken målgrupp brukar dessa sorters applikationer ha och finns det något man bör vara extra noggrann med när man utvecklar en applikation för denna målgruppen?

- Målgruppen är lite blandad, kan vara IT-avdelning eller receptionister som kollar på systemet, man kan också överlåta ansvar till andra avdelningar.

För att göra det möjligt att övervaka flera dörrar samtidigt så har vi valt att skapa en interaktiv planritning som operatörer kan använda för att status på dörrar och enheter i en viss plan. Är detta en lösning som kan fungera bra i praktiken? Operatörer kommer att interagera med interaktiva planritningen, vad bör man tänka på UX-mässigt här?

- Detta är ett svårt problem att lösa eftersom man måste koppla samman det visuella med data. Detta innebär också att man måste skapa dessa planritningar för alla de företag/kunder som ansluter sig till tjänsten (använder sig av applikationen). Det finns företag som erbjuder detta med interaktiva planritningar som produkt men det är svårt att generalisera detta för många kunder. Det är dock nästa steg inom dessa sorters applikationer, eftersom detta är en annan abstraktionsnivå som inte erbjuds i den nuvarande lösningen som Axis har. Detta kan förenkla det för mindre tekniskt kunniga eftersom det nuvarande befintliga systemet tvingar en användare att kunna förstå hur alla dörrar och helheten hänger ihop när man har flera dörrar som måste övervakas. En interaktiv prototyp kan förenkla detta genom att ge en helhetsbild. Dock är detta en svår grej att lyckas med i praktiken.

I många av dagens applikationer erbjuds ett ljus/mörkt läge för applikationens UI. Exempelvis så finns valet av ljus/mörkt läge i Axis applikationen för övervakning av en dörr. Är detta något som kunder förväntar sig i säkerhetsbranschen? Och hur bör man tänka kring att erbjuda en sådana alternativ till användaren?

- Det är ganska populärt, man önskar nog ha mörkt läge före ljus eftersom systemet kan vara igång på skärmen under en längre tid. En äldre anledning till varför mörkt läge blev populärt var att ljus läge brukade bränna fast pixlarna i äldre bildskärmar annars så föredras mörkt läge också om man sitter i ett mörkt rum för att inte skada ögonen.

I applikationen tänker vi använda oss av data som kan i realtid uppdateras och vissa användaren exempelvis om en dörr är öppen eller stängd live på interaktiva planritningen, finns det något att vara extra försiktig med här för att ge användaren en bra upplevelse?

- Vi borde inte "lova" realtid eftersom fel kan uppstå som gör att det inte blir realtid eller så kan det ta tid för något att uppdateras. Ett annat problem som identifierades handlar angående integritet för personerna som visas på den interaktiva prototypen, det är stor risk att personerna inte vill kunna "pin-pointas" på en skärm.

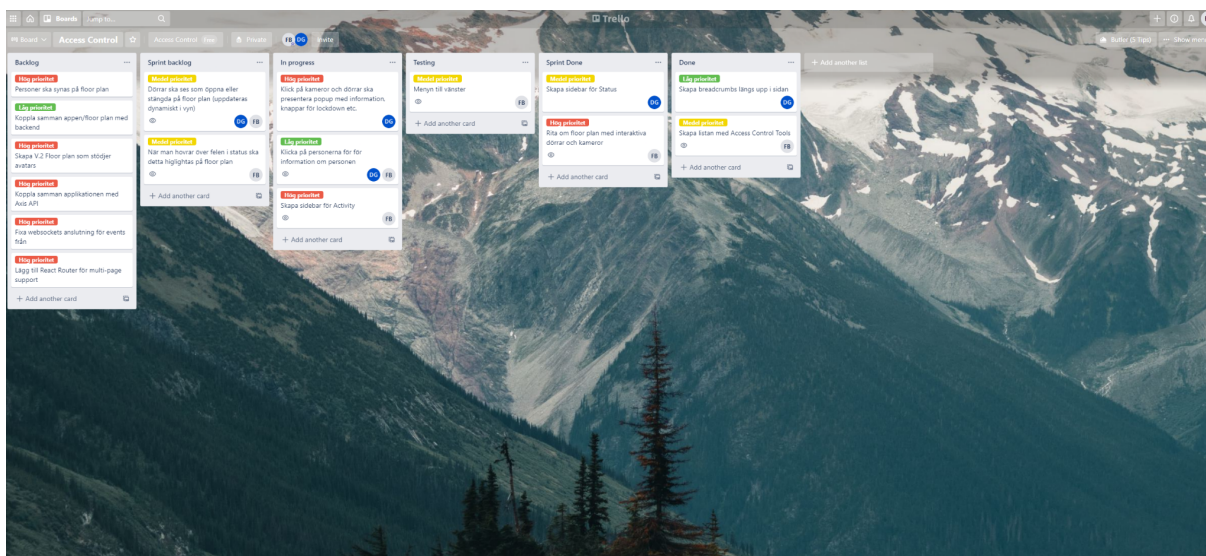
Var går gränsen om man använder sig av en design standard som Material Design i hur mycket av designen man kan modifiera och skraddarsy själva designen? Man vill givetvis ge användaren något familjärt men samtidigt vill man även skapa något unikt som sticker ut för användaren. Vilka delar av en populär design standard som Material Design är en "no touch zone" och vilka delar kan man mer fritt experimentera med? (Fråga specifikt om Navigationsmenyn som vi designat i vår applikation)

- Att avvika från en standard är en okej sak att göra om denna inte uppfyller de behov som man har. Det är dock så att de oftast bygger på någon universell standard som förenklar användningen för användaren. Att skraddarsy sin design är en sak som är möjligt att göra med Material-UI komponenter dock så specificerades det att detta är en svår grej att göra, och detta kan leda till att komponenten behöver byggas upp från grunden.

Slutligen finns det generellt sätt något man bör tänka på och som är enkelt att glömma när man försöker skapa ett bra UX? Vad ska man tänka på när det gäller färger och att färgkoda saker?

- Om man färgkodar mycket ska man vara noga med att det viktiga markeras tydligt så inte färgblinda inte vet vad som händer i applikationen (ex. Dörren highlightas)

9.3 Scrumboard



A.10.1 Skärmdump av scrumboarden som användes under examensarbetets gång.