



Linnéuniversitetet

Kalmar Vaxjö

## Slutrapport

# nocKnoc

*Meridium*



*Författare:* Joel Carlsson, David Stureson,  
Karl Erelöf, Carl Einarsson, Marcus  
Thureson  
*Handledare:* Tobias Ohlsson  
*Examinator:* Tobias Ohlsson  
*Termin:* Våren 2019  
*Ämne:* Datavetenskap  
*Nivå:* Grundnivå



**Linnéuniversitetet**  
Kalmar Vaxjö



## Innehåll

<b>Innehåll</b>	<b>iii</b>
<b>1 Inledning/Bakgrund</b>	<b>1</b>
1.1 Syfte	1
1.2 Mål	1
1.2.1 Baskrav	1
1.3 Projektorganisation	2
<b>2 Genomförande</b>	<b>3</b>
2.1 Metodik	3
2.2 Teknik	3
2.2.1 Client	3
2.2.2 Server	4
<b>3 Resultat</b>	<b>7</b>
3.1 Acceptanstest	7
3.2 Client	7
3.3 Server	7
<b>4 Avvikelser/efterkalkyl</b>	<b>11</b>
<b>5 Slutsats</b>	<b>13</b>
5.1 Förslag på vidareutveckling	13
5.2 Övertagande organisation	14
5.3 Förslag till förbättringar inför kommande projekt	14





## 1 Inledning/Bakgrund

Meridium är ett konsultföretag som arbetar med att utveckla digitala tjänster. Företaget grundades i Kalmar och finns idag både i Kalmar och i Stockholm. Meridium Kalmar har precis flyttat till sina nybyggda lokaler där de kommer fullfölja sina planer på att expandera. På Meridiums nya byggnad i Kalmar har de idag ett problem. Deras entrédörr går endast att öppna via nyckeltagg eller från insidan och personalen befinner sig två våningar ovanför entrédörren. Detta gör att besökare idag hamnar utanför dörren där en papperslapp presenterar ett telefonnummer till Meridiums växel. Visionen med detta projekt är att leverera en lösning som underlättar för besökare och Meridiums anställda när besökare kommer till byggnaden.

### 1.1 Syfte

Meridium vill att vi tar fram en lösning som underlättar proceduren för besökare att kontakta anställda inne i byggnaden vid ankomst. Besökare ska kunna välja vem de söker och denna person ska snabbt bli notifierad att någon söker denne.

Besökaren kommer få välja deras relation till företaget och beroende på val kommer de få möjlighet att kontakta företaget, antingen genom val av person eller ett förslag om de inte vet vem de ska kontakta.

### 1.2 Mål

Målet är att vi ska leverera ett system som underlättar för besökare och dess kontaktpersoner när besökare kommer till byggnaden. Systemet ska bestå av en applikation, installerad på en surfplatta som är placerad vid entrédörren. Applikationen kommer att kommunicera med ett API som utvecklas till applikationen. Detta API kommer kopplas till Meridiums workspace i det digitala kommunikationsmedlet Slack som företaget huvudsakligen använder för kommunikation internt. API:t kommer att kommunicera med medlemmar i workspacet huvudsakligen genom teamkanaler men även genom direkta meddelanden till medlemmar och mot gemensamma kanaler för alla anställda på Kalmarkontoret.

#### 1.2.1 Baskrav

- Applikationen ska underlätta för Meridiums besökare.
- Applikationen ska underlätta för Meridiums anställda.
- När besökare söker någon ska ett meddelande om detta skickas i Slack.
- Meddelande om att person söks ska i första hand gå ut till den personens team, i andra hand en generell kanal.
- Meddelandet skall kunna besvaras av de anställda.
- Systemet ska vara lättare och mindre tidskrävande än den nuvarande lappen på dörren.
- Hårdvaran ska vara en tablet som klarar utomhusbruk.



- Applikationen skall ha ett grafiskt UI som följer Meridiums grafiska profil.
- Systemet ska vara säkert och inte äventyra säkerhet/känslig information på Meridium.

## 1.3 Projektorganisation

Projektorganisationen består utav fem campus studenter som har beslutat att inte genomföra projektet med en hierarkisk uppdelning på gruppen.

Eftersom Meridium är en platt organisation utan chefer och gruppleddare valde vi att följa detta också. Varje alternativ övervägdes demokratiskt i gruppen och flest röster blev den valda lösningen. Ingen gruppleddare behövde tilldelas vid användning av denna metoden.

Detta resulterade i en positiv utgång för alla medlemmar i gruppen. Det är viktigt att tillägga att detta kan vara en ytterst olämplig lösning för andra grupper. Alla medlemmar i grupp Meridium har känt varandra sen i början av utbildningen och alla vet vad alla är kapabla till och hur varje medlem behandlar olika situationer. Gruppdynamiken var byggd på en solid grund långt innan projektets startskott, vilket kan vara en stor anledning till metodens positiva resultat.

Teamet delades upp i två delar. Två i Client och tre i Server. Grundplanen var att hålla personer roterande i grupperna för att hålla ett gemensamt engagemang och gemensam delaktighet i systemets alla delar. Detta var något som försvann nästintill direkt och grupperna hölls samma genom hela projektet. Med den begränsade tidsbudget som fanns tillgänglig var det helt enkelt inte möjligt att alla skulle kunna vara delaktiga i allt.



## 2 Genomförande

### 2.1 Metodik

Gruppen blev tilldelade en egen arbetsplats på Meridiums kontor. Där spenderar vi ungefär tre fulla dagar i veckan. Vi utförde projektet genom att dela projektiden i iterationer där varje iteration är en vecka. Sammanställning utav tidigare vecka och planering för kommande vecka sker på måndagen varje ny vecka. De stora händelserna som milestones och liknande lades in i planen redan vecka ett så att endast utveckling och dokumentation behöver vara i åtanke när veckan planeras.

Vi kommer främst använda oss utav Github Wiki för dokumentation. Dokument som bäst administreras via kalkylark kommer att skrivas i Google Docs. De dokumenten kommer att länkas till Github wikin så att all dokumentation finns samlad på samma plats.

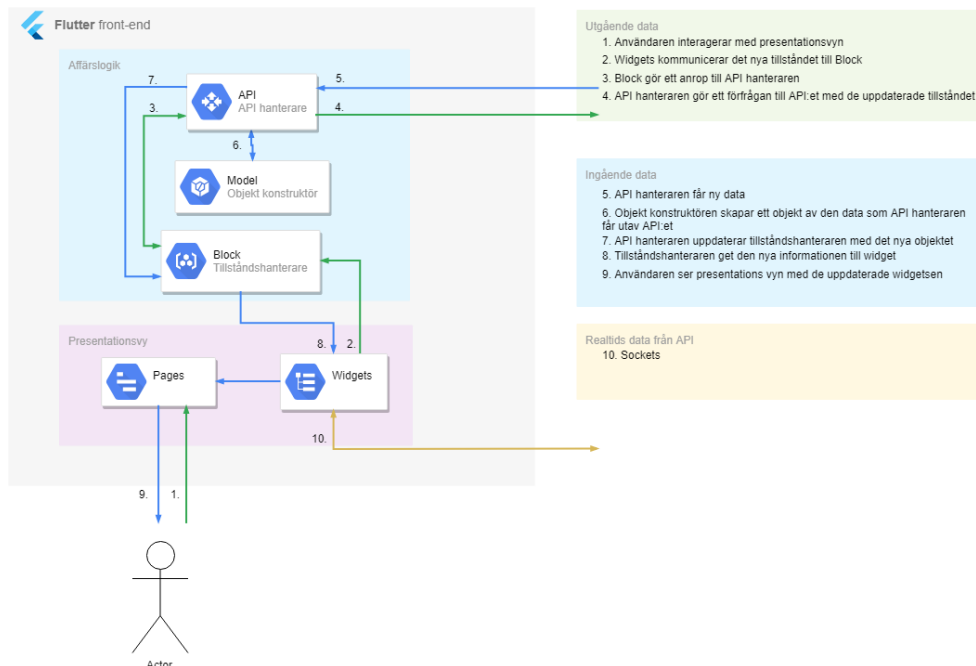
Koden kommer versionshanteras via ett delat Github repo. Där vi har två branches en för server och en för klient. Arbetsdagen börjas med att alla i organisationen gör en git pull för att hämta den senaste versionen på systemet. I slutet på varje arbetsdag så mergar vi ihop de nya implementationerna och går igenom möjliga konflikter.

### 2.2 Teknik

#### 2.2.1 Client

Språk	Dart
Ramverk	Flutter
State management	Bloc
Emulation	Android Studio
Kod Editor	Visual Studio Code(Live share)
Versionshantering	Git/Github

Besökaren kommer att interagera med en “Native-app” byggd i språket Dart med ramverket Flutter. Valet av språk är en effekt av det aktiva valet av ramverket Flutter. Flutter fungerar på samma vis för både iOS och Android och skapar därför ett bredare utbud av plattform. Detta avlönas om kund någon gång väljer att byta ut enheten vid entrédörren. Flutter är utvecklat av Google och är fortfarande ett nytt ramverk men har stabila versioner vilket innebär att de inte kommer tas ur bruk och sluta fungera. Likt “React Native” arbetar flutter med komponenter. Komponenterna i flutter behöver nödvändigtvis inte tas från tredje-parts paket då en välutvecklad grund av komponenter kommer med ramverket och på så vis undviks risken för paketberoenden.



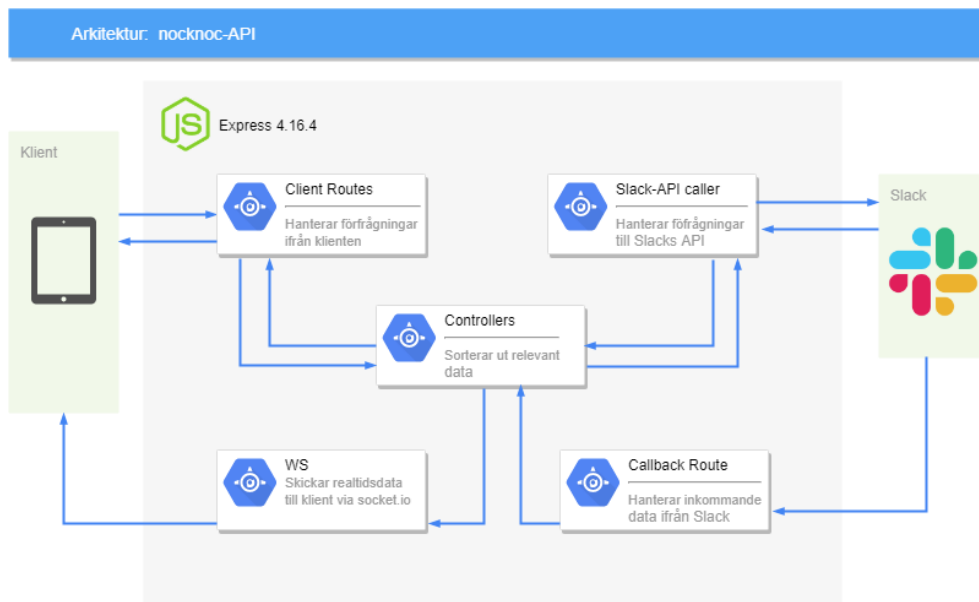
Figur 2.1: Klient-diagram

## 2.2.2 Server

Språk	NodeJS
Ramverk	Express.js
State management	Bloc
Lagring	JSON
API	Slack
Kod Editor	Visual Studio Code(Live share)
Kodstandard	Enligt JavaScript Standard Style
Versionshantering	Git/Github

Servern kommer bestå utav ett node.js-system med express som ramverk. Servern kommer ta hand om requests och uppfylla klientens önskan genom kontakt med Meridiums Slack-workspace genom paketet snack.js som kommunicerar med slacks web-API. Servern kommer att tillhandahålla listor med anställda och arbets-grupper. objekt i dessa listor kommer att kunna kontaktas i slack, genom request till servern. Mot slack kommer servern skicka meddelanden till kanaler ta emot svar från slack om meddelandet bekräftas. Servern kommer även ta bort meddelanden från slack för att hålla ett bra flöde för slackanvändarna då appens meddelande endast fyller en funktion när en besökare är vid dörren. graphicx subfig





Figur 2.2: API-Diagram



(DENNA SIDA HAR LÄMNATS TOM MED AVSIKT)



## 3 Resultat

### Resultatbeskrivning/måloppfyllelse

Utav våra baskrav är samtliga utav de mätbara kraven uppfyllda i nuläget. De krav som är svårare att mäta, exempelvis att det ska vara enklare för besökaren än att ringa växeln, är något som tester visar positivt på men som inte går att avgöra om det är helt uppfyllt förrän ett tag in efter produktionssättning. Förhoppningen är att samtliga utav dessa krav även ska vara uppfyllda.

### 3.1 Acceptanstest

Kund är nöjda med resultatet, Kända buggar offentliggjordes på mötet. Miljön sätts upp måndagen efter. Då samlas samtliga medlemmar på Meridium tillsammans med Niklas och skapar servermiljön. Inga baskrav är missade. Vi klonar projektet till ett nytt repo och låter Meridiums organisation överta projektet. Meridium kommer att använda systemet och räknar med att få upp allt under nästa vecka.

### 3.2 Client

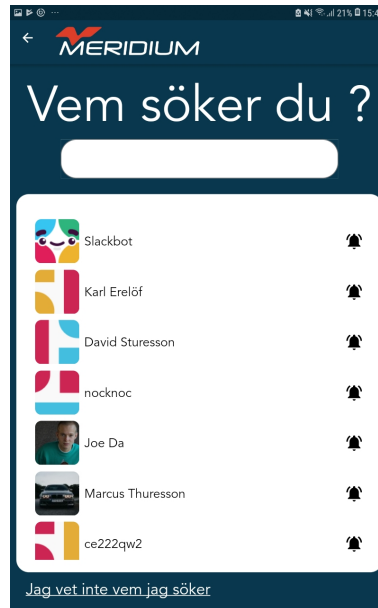
Första sidan av applikationen möts användaren av 4 olika val, *leverans*, *besökare*, *affärspartner* och *anställd*. Vid valet får användaren identifiera sig. Efter identifikation blir användaren presenterad med en lista av alla inne på Meridiums kontor, där har användaren möjlighet att skriva in vem användaren söker. Användaren har valet att göra en specifik kontakt eller en allmän kontakt. Vid allmän kontakt blir den valda fallback kanalen kontaktad. Vid kontakt har användaren 3 sekunder på sig att avbryta kontakten om användaren har valt önskad person användaren tryckt på, kan användaren trycka på klockan för att kontakta direkt. När en kontakt är skedd skickas en post till servern api, när den kontaktade svarar, visas en bild på den som har accepterat och ett meddelande om att personen är påväg. Vid en kontakt där personen i fråga inte svarar och emot all förmodan att ignen svarar i fallback kanalen kommer det att komma upp ett nummer till växeln för att kunna göra en telefonkontakt.

### 3.3 Server

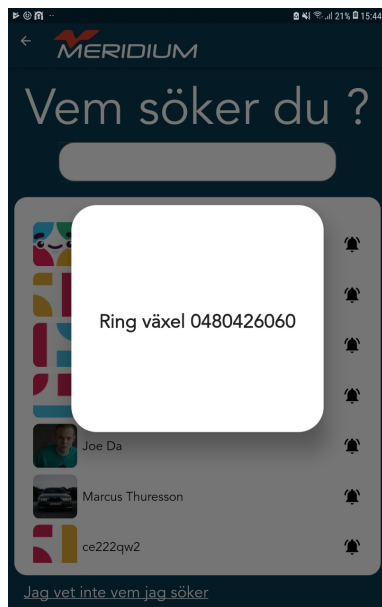
Servern har kontakt med Slack och klienten och medlar och räknar om data från båda sidor för att de ska passa för varandra. Servern sköter även utseendet mot Slack och bygger så kallade blocks som Slack använder för att göra listor och knappar. Detta gör att servern kan hämta användare från Slack, tolka dessa så att det passar klienten och skicka vidare det. När klienten svarar att en person på Meridium ska kontaktas, behandlar servern detta genom att bygga ett meddelande och en Accept-knapp som används för att godkänna att personen som trycker ska gå och öppna dörren. Detta meddelande skickas till den valda personens team. När en person klickar på accept skickar servern namn och bild på den som har klickat tillbaka till servern. Om ingen svarar på 30 sekunder skickas meddelandet istället till en gemensam kanal för alla anställda.



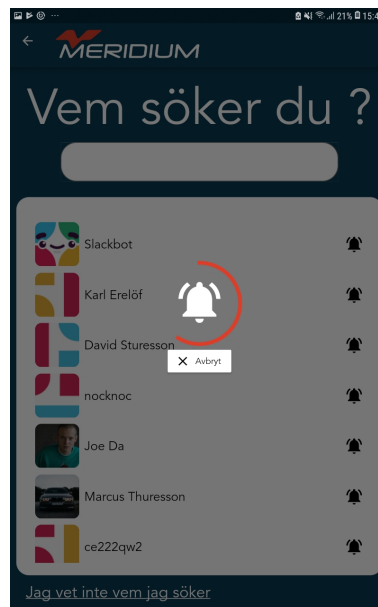
(a) Start sida



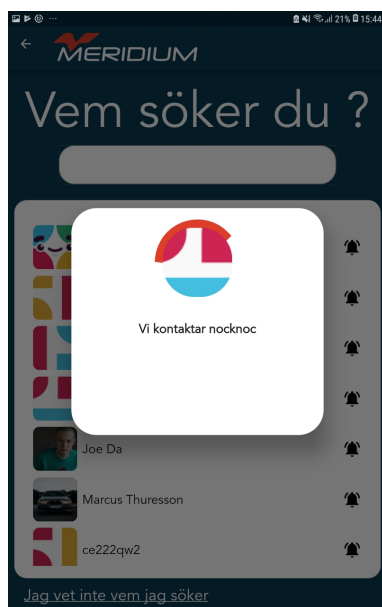
(b) Lista



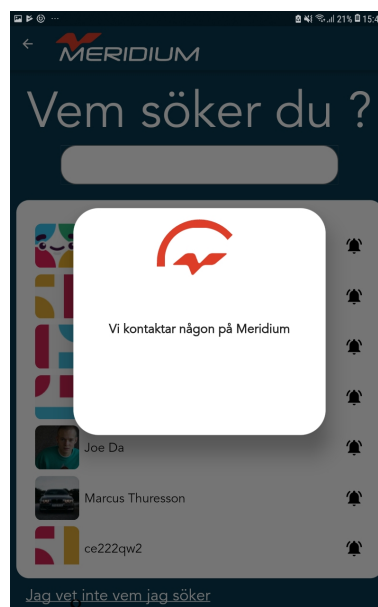
(c) Växel



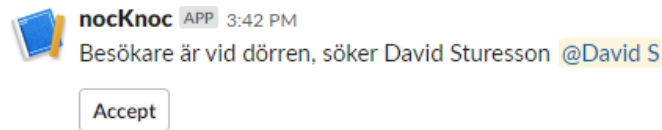
(d) Avbryt



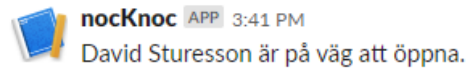
(e) Specifik kontakt



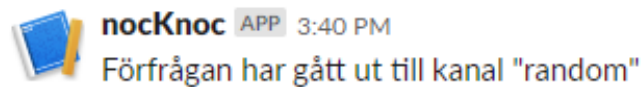
(f) Kontakta alla



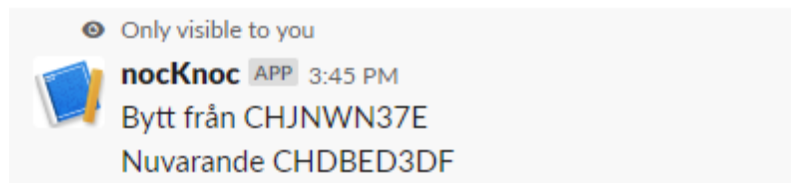
(a) Besökare vid dörren



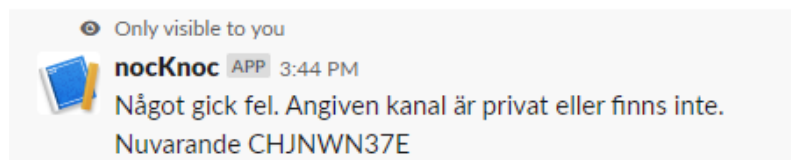
(b) På väg att öppna



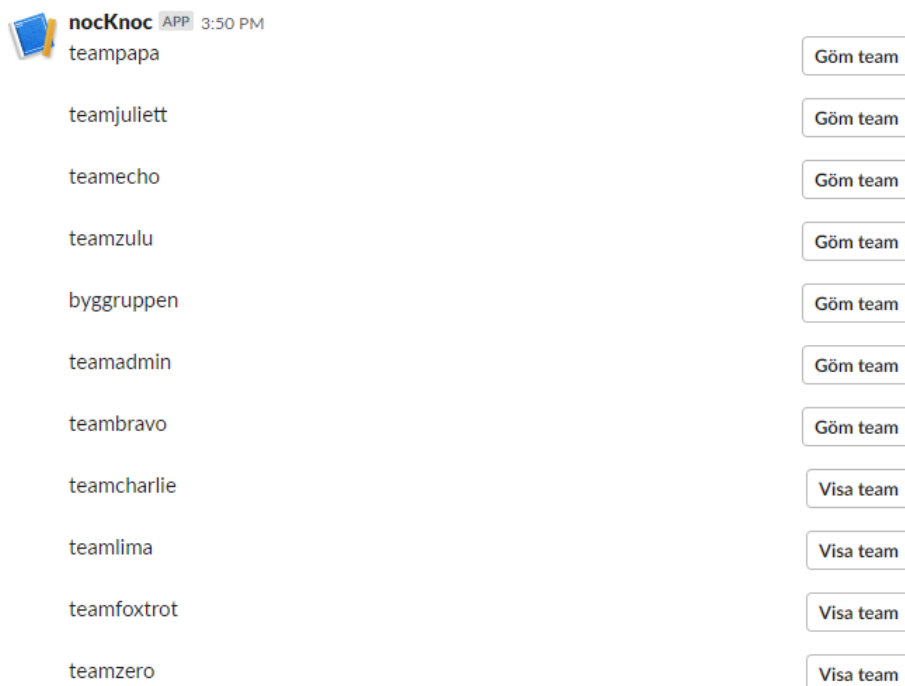
(c) Fallback-kanal



(d) Bytt fallback



(e) Ogiltig fallback



(f) Inställningar



(DENNA SIDA HAR LÄMNATS TOM MED AVSIKT)



## 4 Avvikelser/efterkalkyl

Samtliga krav som är testade är uppfyllda. Vi önskar skarpast att vi hade haft mer tid för att göra mer utav det som nu har hamnat på framtida utveckling. Vi hade även velat göra ordentliga användartester så att vi vet att samtliga krav är uppfyllda redan innan leverans.

Kunden gav oss väldigt fria händer, vilket är tacksamt för kreativt tänkande och underhållningsgraden av projektet under tidens gång. Det skapar dock en oro vid slutleverans eftersom vi är osäkra på om det blir precis som kunden ville. Vi har haft täta möten med kund och kund har godkänt våra förslag och krav. Osäkerheten kan eventuellt grundas i att det är det första utvecklingsprojektet mot en riktig kund för samtliga gruppmedlemmar. På grund utav de fria tyglarna kan ambitionsnivån nått för högt. Våra visioner i början av projektet har fått tonas ner efter hand och spår av dessa höga ambitioner kommer att leva kvar i koden av nocKnoc v1.0. Det blev inga tydliga överlämningar till kund eftersom vi alltid fanns i deras kontorsmiljö. Det var enkelt att lyfta små saker direkt samtidigt som det var lika enkelt att skjuta saker framför en. Detta satte en del viktiga punkter i kläm när de enkelt sköts på morgondagen, Den täta kontakten med kund gjorde också att mycket av det implementerade arbetet missas att redovisas i dokumentationen. Mötesprotokoll förs dåligt när mötena är spontana och korta, vilket resulterar i implementation med missad dokumentation.

Grundplanen vara att lyckas dra ut ström till surfplattan utanför och montera den permanent. Detta förstod vi ganska snabbt att det var omöjligt på tidsbudgeten om det skulle finnas någorlunda logik i den medföljande applikationen. Vi lyckades 3D-printa ett skal till plattan som gör det möjligt att sätta upp den utanför dörren och skalet blev både spacklat och spraymålat i matt-svart. För att vara en provisorisk lösning, är vi nöjda.

Planen från början var att skriva gemensamma dokument i Google docs, för att sedan fylla över dessa i Github wiki. Detta blev en omväg då det missades dokument och uppladningsflödet var instabilt. När det var tillfälle för opponering av vår tilldelade opponeringsgrupp blev det därför ofta stressigt att få över all dokumentation. Vi valde därför, efter ett tag att skriva direkt till wikin med undantag för kalkylark som iterationsplanen.

Vi är tacksamma att vi höll oss till node/express på servern eftersom inläringen av Flutter och dart har varit en stor del av projektet. Även att arbeta med Slack har tagit tid och att besitta kunskap och självsäkerhet i en stor del av serverns uppbyggnad har varit en stor del varför ett fungerande system finns tillgängligt vid projektets slut.

Vi valde att skapa ett eget workspace där vi kunde testa vår kod våldsmat och vilt genom att skicka notiser i Slack i workspaces alla hörn. Detta var positivt under de tungt funktionella utvecklingsfaserna. Vissa dagar kunde hundra notiser gå ut till flera kanaler och det hade inte gått att göra i deras offentliga workspace. Det straffade sig dock mot slutet när vi kopplade över vårt system på deras workspace och stöter på ofantligt många problem som vi inte lagt märke under utvecklingen i vårt Mock-workspace.

Timmarna som har lagts på dokumentation är oerhört många. Om kursen varit större är 7.5 hp hade det varit rimligt. På den tiden som finns tilldelad under denna kurs



innebär det att dokumentationen hamnar som en av de största delarna av projektet. Det är försvarbart eftersom det finns en stor vikt i att lära att dokumentera men när projektet utvecklas till en riktig kund, är det viktigt att sätta kunden i fokus.





## 5 Slutsats

Vi är nöjda med projektet och resultatet.

Vi är framförallt nöjda med arbetsteamets samarbete. Vi hade turen att bli en grupp med fem redan nära vänner och detta skapade stora möjligheter för projektet. Vi klarade oss undan den allt för långa lära-känna-fasen och vi kunde direkt bestämma att vi inte behövde utse en gruppleddare eller ansvarig. Alla känner till allas styrkor och svagheter och alla känner förtroende i att alla medlemmar strävar med samma engagemang mot det gemensamma målet. Detta har resulterat i en ytterst positiv studieperiod.

Samarbetet med kund har fungerat fantastiskt. Vi hade kunnat ställa mer krav på varandra än vi gjorde. Kunden lät oss utveckla fritt, samtidigt som vi undvek att störa kunden till så hög grad som möjligt. Det hade förmodligen inte varit några större komplikationer om kunden satte högre krav på vår utveckling eller om vi satte högre krav på vad kunden behövde tillhandahålla för oss. Vi hade exempelvis kunnat önska att få större tillgång till deras workspace i Slack och de hade kunnat kräva fler baskrav av oss. Dock kan anledningen vara att samarbetet har fungerat så pass bra redan från början, att tillit och förtroende sänkte behovet för rak kommunikation. Från gruppens sida stämmer detta och vi hoppas att Meridium känner det samma.

### 5.1 Förslag på vidareutveckling

I nocKnoc v2.0 hade gruppen gärna sett en ordentlig implementation utav att plocka ut teamkanaler. I denna versionen kommer samtliga kanaler för en användare att komma i en array. Klienten får välja kanal att skicka tillbaka, men endast en tas emot av server. Där hade vi velat se en bättre lösning där en specifik kanal väljes. Kanske den anställdas önskade kanal?

När en anställd kommer till dörren ska de kunna välja ett team att kontakta istället för att välja en anställd. Server har all funktionalitet för att leverera teams och kanaler för teamet.

Innanför dörrarna ligger två företag, Meridium (Tieto) Meriworks. Ett steg för att välja vilket företag utav dessa besökaren vill ha kontakt med.

Bygga stöd för engelska i klienten. Idag är det svenska både på klient och server. Klienten har knappar för att byta språk, men endast svenska går i nuläget att välja. I klienten önskas det även att lista där var av person sker, ska kunna gå att sortera. Huvudsakligen efter förnamn.

Server är förberedd för att skicka ut informationsmeddelanden i Slack från klienten men klienten skickar idag inga meddelanden av denna typ. Förslag på meddelande är notifiering vid lågt batteri i klient-enheten.

När systemet inte används hade en demofilm gjort sig bra på skärmen. I denna bör det visas hur systemet används. Detta klipp bör försvinna när skärmen reagerar på interaktion.

När servern svarar med statuskoder som exempelvis 500, eller om inget svar kommer (inget internet), bör ett meddelande med växelnummer visas i en kort tid.



Detta meddelande finns idag men visas endast om ingen anställd svarat på att öppna på två minuter.

## 5.2 Övertagande organisation

Meridium står som övertagande för projektet. Leverans sker på plats i Meridiums lokaler med målet att leverera i slutet på vecka 22 efter slutligt acceptanctest. Vi gör prototypmontering vid dörren under denna vecka tillsammans med projektansvariga från Meridium. Demo av systemet sker i samband med montering och görs inför företaget vid lunch vid produktsättningsdagen. Repo kommer att överlämnas för kundens möjlighet att själva uppdatera och skapa om delar av systemet.

## 5.3 Förslag till förbättringar inför kommande projekt

Vi kommer att ta med oss mycket ifrån detta projekt. Saker vi kommer att göra bättre i framtiden är:

Boka ordentliga veckomöten med kund. Vi arbetade med spontana och korta möten då vi satt med kunden i deras lokaler. Detta resulterade både i att vissa punkter sköts på framtiden, samt att mötesprotokoll fördes slarvigt. Ordentliga möten hade tvingat fram att mötespunkter skrevs ner emellan mötena. Detta hade resulterat i bättre dokumentation.

Göra kontinuerliga tester utanför mockad miljö. Vi behövde skapa ett workspace likt Meridiums för att kunna skicka notiser i den mängd vi önskade utan att störa de anställdas arbete. Detta gynnade oss med notiserna men när vi kopplade på mot Meridiums workspace var det många bitar som vi hade missat. Detta resulterade i en ordentlig period av reparationer, precis i projektets slutetapp.

Sätta rimliga krav och börja med baskrav. Vår kund gav oss nästan helt fria tygla vilket betydde att vi skapade kraven till största del själva. Detta satte vår ambitionsnivå en aning högt och grundläggande funktionalitet hamnade emellanåt i skymundan.

Skapa ett förutbestämt flöde för hur ny funktionalitet ska implementeras. Det hände emellanåt att ny funktionalitet implementerades och att dokumentationen missades. Att ha ett bestämt flöde för att inkludera dokumentation, testning och implementation när en ny funktionalitet tillkommer hade underlättat.