

Kasper J. Hermansen, Samir A. Habibi

# CONVERGE

## Preprojekt Journal

Bachelors

Bachelor's: Information, kommunikation og teknologi

submitted to



## Aarhus School of Engineering

Supervisor

Jesper Michael Kristensen

Aarhus School of Engineering  
Lektor

Aarhus, Juni 2019

Projekttitel:	Converge		
Projektnr.:	Udbyder	Vejleder	Bivejleder
77		Jesper Michael Kristensen	
Deltagernr	Studienr	Navn	Underskrift
1	201607110	Kasper Juul Hermansen	
2	201606697	Samir Ahmad Habibi	

# Indhold

<b>Abstract</b>	<b>ii</b>
<b>1 Introduktion</b>	<b>1</b>
<b>2 Opgavebeskrivelse</b>	<b>2</b>
<b>3 Kravspecifikation</b>	<b>3</b>
3.1 Overordnede krav . . . . .	3
3.1.1 Must have . . . . .	3
3.1.2 Should have . . . . .	3
3.1.3 Could have . . . . .	4
3.1.4 Won't have . . . . .	4
3.2 Funktionelle krav . . . . .	4
3.2.1 Aktører . . . . .	4
3.2.2 Epics . . . . .	5
3.3 Ikke-funktionelle krav . . . . .	8
3.3.1 Begrebsliste . . . . .	8
3.3.2 Krav . . . . .	8
<b>4 Projekt plan</b>	<b>10</b>
<b>5 Experiment og forsøgsgrupper</b>	<b>11</b>
<b>6 Teknologier</b>	<b>12</b>
6.0.1 Frontend . . . . .	12
6.0.2 Backend . . . . .	12
6.0.3 Meta Application . . . . .	13
<b>7 Research</b>	<b>14</b>
<b>8 Arbejdssted</b>	<b>15</b>
<b>9 Tidsplan</b>	<b>16</b>
<b>10 Konklusion</b>	<b>17</b>
<b>11 Underskrifter</b>	<b>18</b>
<b>Appendix</b>	<b>19</b>
<b>Bibliografi</b>	<b>24</b>

## Figurer

4.1	Scrum board . . . . .	10
9.1	Sprint tidsplan . . . . .	16

# 1 Introduktion

Denne forprojekt journal skal skabe et solidt fundament for Converge bachelors projekt. Det indeholder sektioner som en: kravspecifikation og en opgavebeskrivelse. Samt research af ligende produkter, eller andet brugbar materiale. Der vil også være en sektion der beskriver process, tidsplaner og hvor projekt gruppen befinder sig.

## 2 Opgavebeskrivelse

Internettet er blevet sådan en integreret del af vores liv, at det er tid til at sige farvel til nogle gamle vaner og normer. Converge er en platform so skal være et bud på at fange en relativ ny type arbejdere, freelancere. Freelancing er ikke en ny ting, men måden arbejdsformen er adapteret internettet er. Freelancing er projekt baseret arbejde, hvor arbejderer tager jobs på en decentralized måde. Arbejdsgiveren can vælge den freelancer der passer bedst til arbejdet, og derfor få den maksimale tilfredsthildelse for sit arbejde.

Fordelene er mange, men der er også nogle ulemper. Såsom, at det er vanskeligt at afgøre niveauet af en persons ekspertise, baseret på kun en tekst chat og en cv. Converge prøver at takle dette ved at give avancerede værktøjer til at afgøre ens persons ekspertise, og video chat så freelanceren og arbejdsgiveren nærmere kan afgøre færdigheder. En persons ekspertise kan afgøres via tests og derefter blive vidst som et badge. Video chat kan bruges til at løse konflikter eller afgøre misfortæelser.

Som en web portal skal der bruges mange features, såsom authentication (Godkendelse), authorization (Bemyndigelse), payments (Betalning), search (søgning), video/text (video/tekst) chat, osv. Converge vil blive implementeret med et cloud native indstilling. Hvor moderne koncepter, såsom DevOps (Developer Operations kultur), CI (kontinuert integration), CD (kontinuert deployment) er sat som første prioritet. Sikkerhed vil også være en stor prioritet, eftersom denne app ikke er triviell, eftersom der er en betalings tjeneste, hvor man skal kunne sende og modtage penge.

Converge vil blive udviklet som en SPA (single page application) med et moderne web framework, dette er for at give den bedste bruger oplevelse, på en platform hvor brugeren tit skal bladde i siderne. Converge vil have mange features, for at give en god oplevelse, som et byde system for jobs, projekt håndtering, og iterationer. Der vil også være et meta-system, med statistik og overblik. Dette vil være en integreret del af en administrators værktøj. Dette inkludere også et ticketing system, som skal bruges til at forbedre systemet.

## 3 Kravspecifikation

Krav specifikation sætter fokus på de features som skal designes og implementeres, for at udfylde både krav for brugeren og for systemet.

### 3.1 Overordnede krav

For at specificere krav er der brugt MoSCoW metoden (Wikipedia-contributors, 2019). Denne metode gør det muligt at finde ud af at bestemme hvilke krav der hører hjemme for et Minimal Viable Product (MVP), som i lean udvikling betyder det mindst mulige produkt med produktets essens kan udrulles. Det skal dog siges at der bruges en udvidet version af MoSCoW til at fikse nogle af de mangler metoden har. Såsom, fokus på iterationer og forbedring af eksisterende features, samt en ekstra versionering af denne listing for at kunne vise at der er features der vil komme i en fremtidig udgivelse.

#### 3.1.1 Must have

Disse krav skal implementeres i systemet.

- En bruger skal kunne være enten employer eller freelancer
- En employer skal kunne udgive jobs
- En freelancer skal kunne tage jobs og fuldføre dem
- En bruger skal kunne modtage eller overføre penge
- En bruger skal kunne chatte (text) med en anden bruger
- Systemet skal have en webportal
- En bruger skal kunne logge ind

#### 3.1.2 Should have

Disse krav burde implementeres i systemet.

- En employer bør kunne oprette et projekt
- En freelancer bør kunne indlevere materiale til et projekt
- En bruger bør kunne snakke med en anden bruger over video chat
- En freelancer bør kunne søge efter jobs via. genre
- En bruger bør kunne tage tests for at bevise sin ekspertise

### 3.1.3 Could have

Disse krav kan implementeres i systemet.

- En bruger kan registreres med google sso (single-sign-on)
- En bruger kan slette sin konti (GDPR)
- En bruger kan gendanne sit password, og modtage en mail med et nyt
- En bruger kan ændre sin email & password
- En bruger vil modtage emails med opdateringer fra platformen
- En bruger vil modtage notifikationer med opdateringer fra platformet

### 3.1.4 Won't have

Disse krav kan implementeres i systemet.

## 3.2 Funktionelle krav

I denne sektion vil de funktionelle krav blive beskrevet. Funktionelle krav til systemet beskrives ved brug af User Stories (Cohn, 2019). Fordi der laves User Stories til at beskrive systemet, vil det være logisk og gruppere disse under såkaldte Epics. Epics beskriver en gruppe User Stories og deres samlede historie. Dette er et logisk samplepunkt for User Stories og deres tilhørende persona.

Epics er beskrevet som navnet på en funktionalitet, og User Stories efter formen, "As a ... I want to ... so that ...", dette er splittet op i persona, handling og resultat. Hvor persona er en personifikation af en aktør for systemet, handlingen er persona'en der gør noget som aktivere en funktion, for at få et resultat, hvilket er målet og motivationen for brugeren.

### 3.2.1 Aktører

I dette afsnit vil aktører blive beskrevet for systemet, Aktører er ikke det samme som en persona, men en persona udspringer af en aktør men er et virkeligt billede af dette.

Tabel 3.1: Aktør beskrivelse for employer

Navn	Employer
Beskrivelse	Employer er en bruger som ønsker at få lavet et stykke arbejde som han/hende ønsker gjort. Personen søger den ekspertise der er på platformen, og derfor nogle af de freelancers der befinder sig.
Persona	Jens er en entrepeneur på 40 år, han har sit eget murefirma og har brug for en ny brochure til sit firma, hans budget er ikke stort, men han vil stadig gerne have god kvalitet.  Karsten er ansat i en større design virksomhed, denne virksomhed bruger som regel outsourcing til at fremstille websites fra deres designs. Han er den primære person ansat til at styre outsourcing
Mål	At udnytte Converge til at finde god og billig hjælp til opgaver som kræver ekstra hænder.

### 3 Kravspecifikation

Tabel 3.2: Aktør beskrivelse for freelancer

Navn	Freelancer
Beskrivelse	Freelancer er en bruger som ønsker at bruge Converge til at tjene penge ved brug af sin ekspertise
Persona	Mette er en Web designer som bruger Converge ved siden af sit fuldtidsjob. Mette har brug for nogle ekstra penge, så hun tager gerne mange klienter og er en flittig bruger af systemet.
Mål	At udnytte Converge til at finde klienter, både kortvarigt og langsigtet.

Tabel 3.3: Aktør beskrivelse for admin

Navn	Admin
Beskrivelse	Admin er en administrator ansat af Converge til at holde styr på eller udvikle på Converge, dette indebærer tekniske løsninger
Persona	Lars er en Admin ved Converge og bruger meget tid på at kigge fejlsøge i eksisterende løsninger, samt at oprette detaljerede rapporter til Developers
Mål	At bruge meta applikationer omkring Converge til at fejlsøge i systemet.

Tabel 3.4: Aktør beskrivelse for developer

Navn	Developer
Beskrivelse	Developer er en person ansat af Converge til at udvikle på platformen, dette vil sige alt fra design til at skubbe kode til produktion
Persona	Samir er en udvikler ansat på deltid ved Converge, og udvikler aktivt på applikationen, han er flittig til at tage issues og udbedre fejl
Mål	At forbedre Converge, så det bliver den bedste freelancing platform

Tabel 3.5: Aktør beskrivelse for supporter

Navn	Supporter
Beskrivelse	Supporter har kunde kontakt og informere admins omkring mulige fejl
Persona	Kasper er ansat som supporter ved Converge, hans dagligdag går med at snakke med kunder omkring deres problemer, han har et tæt samarbejde med flere af firmaets Admins
Mål	At give kunderne den bedste oplevelse muligt, så Converge kan blive den bedste freelancing platform

#### 3.2.2 Epics

- Logon & Registering
- Employer flow
- Freelancer flow
- Projects
- Video chat
- Text chat
- Search
- Cloud Native Tooling (meta applikation)



### 3 Kravspecifikation

Tabel 3.6: Epic - Logon & Registering

Nr	User Story
1	As a User I can register with my email and password, so that I can access the app
2	As a User I can register with my google account, so that I can access the app
3	As a User I can login with my my email and password, so that I can access the app
4	As a User I can login with my google account, so that I can access the app
5	As a User I can reset my password, so that I can regain access to the app

Tabel 3.7: Epic - Employer flow

Nr	User Story
1	As an Employer I can create a project so that I can get my commision done
2	As an Employer I can edit a project so that I can fix any mistakes I've made
3	As an Employer I can delete a project so that I can stop a commision
4	As an Employer I can receive notifications as a freelancer has placed a bid on my project so that I can choose the freelancer
5	As an Employer I can hire a freelancer, so that he/she can do my commision
6	As an Employer I can accept the material the freelancer has uploadet
7	As an Employer I can place a comment on the material the freelancer has uploadet, so that we can create the best product for me
8	As an Employer I can check the credentials of the freelancer, so that I can see that he/she is the right person for the commision
9	As an Employer I can choose to pay the freelancer for the work, so that I receive the final result
10	As an Employer I can report a freelancer, if he/she didn't present work, or didn't handin the assignment by the specified date

### 3 Kravspecifikation

Tabel 3.8: Epic - Freelancer flow

Nr	User Story
1	As an Freelancer I can take a test to show my experience in a subject
2	As an Freelancer I can show a badge so that my experience is validated externally
3	As an Freelancer I can browse available commissions so that I can choose on that fits my skill level and genre
4	As an Freelancer I can bid on an available commission, so that I can become the freelancer for the commission
5	As an Freelancer I can upload material to the project associated with the commission, so that I can fulfill my part of the bargain
6	As an Freelancer I can comment on a project to communicate with the Employer
7	As an Freelancer I can upload a preview, so that employer has access to my work
8	As an Freelancer I can choose to handin the final revision, so that employer can complete the commission
9	As an Freelancer I can report a project if the Employer wasn't fair

Tabel 3.9: Epic - Project

Nr	User Story
1	As an User I can view my ongoing projects to get the material presented
2	As an User I can subscribe to my ongoing project to get notifications about updates

Tabel 3.10: Epic - Video chat

Nr	User Story
1	As a User I want to be able to call another user
2	As a User I want to be able to stop a call

Tabel 3.11: Epic - Text chat

Nr	User Story
1	As a User I want to be able to text another user
2	As a User I want to be able to delete a message

Tabel 3.12: Epic - Search

Nr	User Story
1	As a User I want to be able to search for users
2	As a User I want to be able to search for commissions
3	As a User I want to be able to search by genre

## 3 Kravspecifikation

Tabel 3.13: Epic - Cloud Native tooling

Nr	User Story
1	As an Admin I want to be able trace a request
2	As an Admin I want to be able to receive updates about events in the system
3	As an Admin I want to be able to display graphs about the system
4	As a Supporter I want to be able to send issues about the system
5	As a Developer I want to be able to push new applications to the system
6	As a Developer I want to be able to test the system

### 3.3 Ikke-funktionelle krav

I dette afsnit vil de ikke-funktionelle krav blive beskrevet. De ikke-funktionelle krav er inddelt i kategorier, alt efter hvilken type krav der er tale om, f.eks. A for anvendelighed, S for sikkerhed osv. Derefter en enumerering for at kunne skelne mellem de forskellige krav

#### 3.3.1 Begrebsliste

Der vil blive henvist til begreber som måske ikke er åbenlyse, derfor referes der til begrebslisten, hvor ydre detaljer og nyancer er beskrevet.

- **Kubernetes** is an application that can run on any kind of machine, it orchestrates application and binds them together as one cohesive mass of services.
- **Docker** is a container runtime, which can package applications so that they can be the same on any platform. This is essential for Kubernetes.
- **Cloud Native** is a mindset for how to develop modern cloud applications. It is the process of creating everything in an automatic way, so that trivial steps are automated.

#### 3.3.2 Krav

##### Anvendelighed

Ikke-funktionelle krav der omhandler anvendelighed af systemet

Tabel 3.14: ikke-funktionelle krav i kategorien "Anvendelighed"

Index	Beskrivelse
A1	SLA (Service Level Agreement) 99.0% uptime
A2	Anmodningssvar tid 250 ms
A3	Max nestede sider 4.

### 3 Kravspecifikation

#### Sikkerhed

Ikke-funktionelle krav der omhandler sikkerheden af systemet

Tabel 3.15: ikke-funktionelle krav i kategorien "Sikkerhed"

Index	Beskrivelse
S1	Passwords hashet med en hashing funktion
S2	Applikation bruger tokens til authentication
S3	Applikation har aldrig password i cleartext, forbind med sessionsnøgle

#### Vedligeholdelse

Ikke-funktionelle krav der omhandler vedligeholdelse af systemet

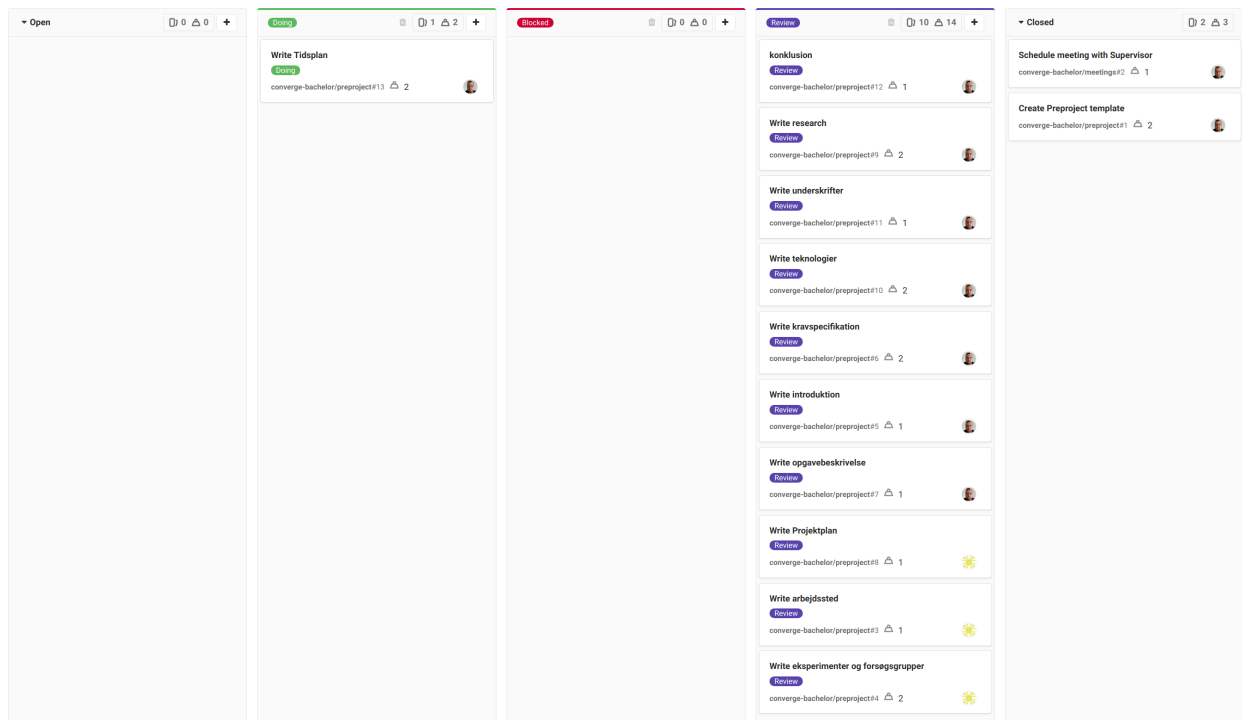
Tabel 3.16: ikke-funktionelle krav i kategorien "Vedligeholdelse"

Index	Beskrivelse
V1	Backend cluster skal kunne rejses op på max 2 min
V2	At slette sin bruger man maks tage 2 min
v3	At oprette sin bruger må maks tage 10 sekunder

## 4 Projekt plan

Projektplan skal være med til at repræsentere det formelle køreplan for projektet. Der vil være en formulering af de nødvendige skridt for at opnå et angivet mål ved at indføre påviselige mål og målbare resultater. Målet er at have en effektiv plan der tjener som vejledende dokument, der gør det muligt at realisere et resultat gennem effektivt teamsamarbejde. Disse mål vil opnås ved hjælp af forskellige værktøjer, scrum, scrum-poker og derudover vil der gøres brug af zenhub boards.

Projektplanen er opstillet i tidsplanen, og vises med et overblik i appendix. Projektplanen er sat op som scrum board, som set på figur 4.1



Figur 4.1: Scrum board

## 5 Experiment og forsøgsgrupper

Der vil udføres forskellige test om de faktiske resultater matcher de forventede resultater og for at sikre, at systemet er Defektfrit. Det indebærer udførelse af en software komponent eller systemkomponent for at evaluere en eller flere egenskaber af interesse. Testene skal også være med til at kunne finde fejl og det sker enten manuelt eller ved hjælp af automatiserede værktøjer (Jenkins X).

Derudover vil der udføres tests, i form af forsøgspersoner. Dette betyder at det er forsøgspersonerne der står for at teste systemet. Målet er at få nogle kunder der benytter vores system, det vil sige enten en employer (tester systemet som employer) og freelancer (tester systemet som freelancer).

## 6 Teknologier

Til Converge skal der bruges mange forskellige teknologier, da projektet skal implementeres med et Cloud Native mindset, så er der rig mulighed for at afprøve en masse forskellige teknologier og værktøjer.

Systemet er delt op i 3 dele, som hver udgrene sig.

- Frontend (hjemmeside)
- Backend (server)
- Meta applikation (support)

### 6.0.1 Frontend

Frontend eller nærmere hjemmesiden, er det produkt som brugerne interagerer med, dette er både for Employer, Freelancer, Admin og Udvikler.

Hjemmesiden bliver skrevet i React, og er en SPA (single page application) som emulere en desktop oplevelse, med minimale page loads og lav latency.

De teknologier der skal bruges er overordnet

- WebRTC (Web Real Time Communication, video og text streaming)
- Http Rest / GraphQL
- Reverse Proxy, SSL (TLS), gzip etc.
- Redux
- React Router

### 6.0.2 Backend

Backend er den server som leverer Converge til brugeren, dette er både hjemmesiden og alt den information der skal til når brugeren trykker på noget på hjemmesiden. Da backenden følger et Cloud Native mindset, bruges der Kubernetes med Docker som container runtime. Kubernetes bruges til orkestrering af applikationer. Docker til at pakke applikationer ind i et flytbart medie. Til at hoste Kubernetes bruges GKE (Google Kubernetes Engine) som tilbyder en up-to-date version af kubernetes, samt andre gode features. Til at pakke applikationer bruges Helm med Chart template, som kan forstå at udrulle containers på kubernetes. Dette gøres ikke manuelt, men via en GitLab Pipeline som tester applikationen og hvis der ikke er fejl, så bliver den automatisk skubbet til et test cluster.

Til at skrive de forskellige applikationer bruges .NET Core med ASP.NET Core framework, som bruges til at lave web-api'er og håndtere en besked fra en central Message Queue (RabbitMQ). Der vil blive brugt en blanding af Python, C# og Go til at skrive applicationerne i det sprog der passer bedst. Til at teste applikationerne bruges Jenkins X hvilket også kører i clusteret.

- Kubernetes
- Docker
- RabbitMQ
- Istio (Service Mesh)

- Helm
- Jenkins X
- C#
- Python
- Go
- NodeJs
- TDD (Test-driven development)

### 6.0.3 Meta Application

Meta applikationen er den applikation der beskriver selve Converge backenden. Det er applikationer som Slack, Versions control osv.

- Git (Source Control, GitOps styring af applikation med git)
- GitLab (Source Control Server)
- GitHub (Source Control Server, Miljøer)
- ZenHub (Scrum)
- Slack (ChatOps, styring af applikation med slack)
- ELK stack (Elastic Search, Logstach, Kibana, brugt til logning)
- Prometheus (Logning og status)



## 7 Research

Til research er der fundet inspiration i lignende applikationer.

- UpWork.com
- Freelancer.com
- Fiverr.com
- GitLab (GitLab, Inspiration til projekt styring)

Disse hjemmesider danne en grundsten for hvad Converge skal være. De har alle fordele og lumper, som Converge vil lære af.

Til test projekter er der fundet:

- HipsterStore (GitHub, Google/IBM)
- eShopOnContainers (GitHub, Microsoft)

## 8 Arbejdssted

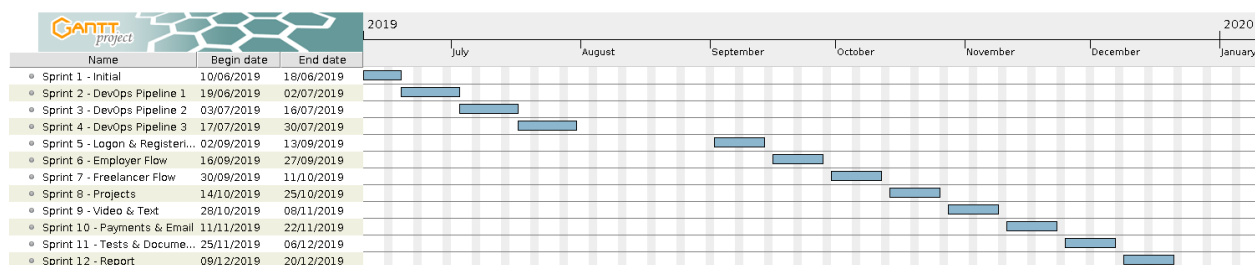
Der er enighed i gruppen at arbejdsstedet kommer til at være på skolen (Aarhus universitet, Kathrinebjerg), hvor der skal søges om at få et lokale. Lokalet vil blive benyttet til at arbejde på projektet hver eneste dag, derudover vil der også være daglige møder, hvor tingene bliver diskuteret, samt et ugentligt vejledermøde.

## 9 Tidsplan

Til en tidsplan præsenteres der et gantt diagram med et overblik over planlagte sprints.

Disse sprints er delt ind i tre bider:

1. DevOps
2. Applikations udvikling & Dokumentations fremstilling
3. Rapport skrivning & afrunding



Figur 9.1: Sprint tidsplan

Nærmere information om gantt kan ses i appendix.

## 10 Konklusion

I denne journal er grundstenene lagt for et solidt udviklingsprojekt. Krav er stillet op, så rammerne er sat for hvad der skal laves, og hvad der har mest prioritet. Det er blevet udpenslet nogle User Stories der definere hvilke løsninger der skal laves. Der er samtidig også fremstillet nogle personaer, for at opstille en autoritet for de krav brugere har for systemet. Samtidig er der lavet research på lignende teknologier, samt hvilke teknologier der skal bruges til udførslen af projektet.

## 11 Underskrifter

Tabel 11.1: Underskrifter

Projekttitel:	Converge		
Projektnr.:	Udbyder		Vejleder
77			Bivejleder
			Jesper Michael Kristensen
Deltagernr	Studienr	Navn	Underskrift
1	201607110	Kasper Juul Hermansen	Hermansen
2	201606697	Samir Ahmad Habibi	Samir Habibi

# Appendix

## Bachelor Converge

19 Jun 2019

<http://converge-app.net>

Project manager

Project dates

10 Jun 2019 - 21 Dec 2019

Completion

0%

Tasks

12

Resources

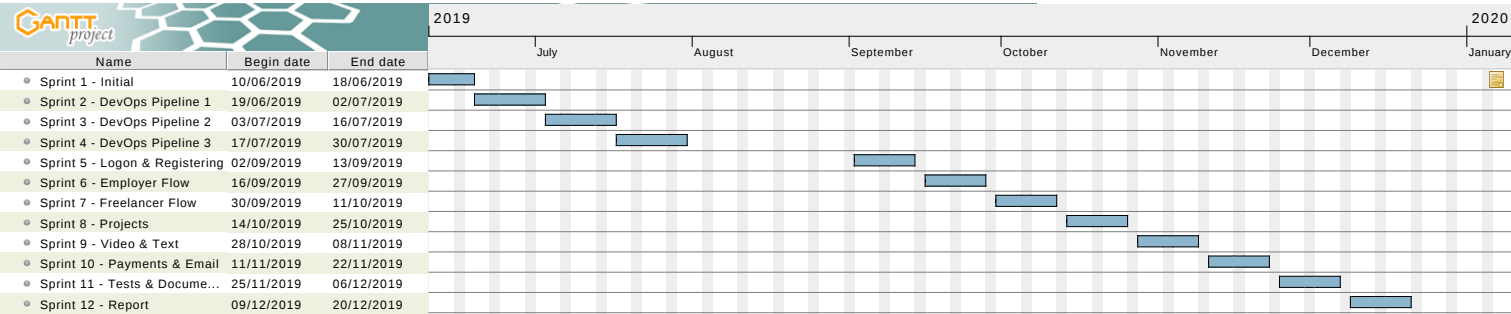
0

Tasks


Name	Begin date	End date
Sprint 1 - Initial	10/06/2019	18/06/2019
Initial sprint to get started		
Sprint 2 - DevOps Pipeline 1	19/06/2019	02/07/2019
Sprint 3 - DevOps Pipeline 2	03/07/2019	16/07/2019
Sprint 4 - DevOps Pipeline 3	17/07/2019	30/07/2019
Sprint 5 - Logon & Registering	02/09/2019	13/09/2019
Sprint 6 - Employer Flow	16/09/2019	27/09/2019
Sprint 7 - Freelancer Flow	30/09/2019	11/10/2019
Sprint 8 - Projects	14/10/2019	25/10/2019
Sprint 9 - Video & Text	28/10/2019	08/11/2019
Sprint 10 - Payments & Email	11/11/2019	22/11/2019
Sprint 11 - Tests & Documentation	25/11/2019	06/12/2019
Sprint 12 - Report	09/12/2019	20/12/2019



Gantt Chart



Resources Chart

		2019								2020
Name	Default role		July	August	September	October	November	December	January	

# Bibliografi

- Cohn, Mike (2019). *User Stories and User Story Examples*. URL: <https://www.mountaingoatsoftware.com/agile/user-stories> (se s. 4).
- Wikipedia-contributors (2019). *MoSCoW method*. URL: [https://en.wikipedia.org/wiki/MoSCoW\\_method](https://en.wikipedia.org/wiki/MoSCoW_method) (se s. 3).