



SOFTWARE ENGINEERING. FØRSTE SEMESTER PROJEKT.

SEP 1 - COLOUR IT MANAGEMENT SYSTEM

ANSLAG: 62.400 M. FIGURER

DATO: 18/12/2020

GRUPPEMEDLEMMER

ANDREAS ØSTERGAARD - 304546

KASPER FALK MIKKELSEN - 304712

VEJLEDERE

Astrid Hanghøj

Michael Viuff



1	Introduction	3
2	Requirements	4
2.1	Functional Requirements	4
2.2	Non-Functional Requirements	5
3	Analysis	6
3.1	Use Case Diagram	6
3.2	Use Case Descriptions	8
3.2.1	Hjemmesiden	8
3.2.2	Java delen	9
3.3	Activity Diagram	10
3.4	UML Diagram	13
4	Design	14
4.1	Class Diagram / Domain model	14
4.2	Sekvensdiagram	15
4.3	Hjemmesidens design	16
4.4	GUI design	18
5	Implementation	25
	Search funktionen	26
6	Test	29
	Requirements	30
	Non-functional Requirements	31
6.1	Test Specifications	33
7	Results and discussion	34
8	Conclusion	35
9.	Project future	36
10	Sources of information	37
11	Appendices	37



Abstract

In this project, we're looking into creating a website as well as an administration interface, where an employee will be able to create, edit and search for projects within the firm. We began the process of making the website, where a customer could get a deeper look into the firm, and its projects. Here we create a frontpage, with slight information about the whole company. From here we made button with would take you to the following sites;

- About page for the company
- Contact page
- Developer page, where a customer can see the employees
- A projects page, where you can see other projects
- And a services page where a customer can see what services the company do

From here we began making the administrator interface, in java. This process included making FXML files in Scenebuilder, from where we sculpted each scene. We created a landing page, where you have four options for roles. Here you're meant to pick your role in the current project you're working on. After you pick a role, you get to the home screen, where you are greeted with 4 different buttons. Here you can choose between "Create New Project", "All Projects", "Search a project" and "Search employees". Each site takes you to the corresponding site. You're able to create a new project in the "Create new project" tab, and view it in both "All Projects" and the "Search a project"s tab. Each project will be saved as a binary file, so that it can be accessed in between restarts.

1 Introduction

Colour IT arbejder med IT løsninger på daglig basis, og har derfor gang i mange forskellige projekter på samme tid, og for at kunne holde bedre styr på det mangler de et system der kan håndtere dataet. Derfor har de kontaktet os. Colour IT mangler et Management system, de mangler en effektiv måde at holde styr på de opgaver der bliver arbejdet på i firmaet, hvem der arbejder på dem, og alt der er relevant til deres projekter. Derudover mangler de også et sted deres kunder kan se hvad de har gang i, så en hjemmeside mangler de også.

Det er vores mål at få løst colour IT's problem, ved hjælp af et Java program, der skal kunne overskueligt vise dem deres igangværende projekter, deres kollegerne, krav, og processer. Vi vil ikke inkludere et login baseret system fordi alt data gemmes lokalt, så det hele fungere kun fra et sted af,



og derfor ville det også være nemt for Colour IT selv at kontrollere hvem der er bruger af systemet. Derudover hvis nogen ikke skulle have fat på informationerne, ville et login system heller ikke hjælpe da alt de har brug for ligger på en fil der bare kan snappes. Herunder finder du vores kravspecifikationer, vores analyse og diagrammer af systemet, design og implementation.

2 Requirements

Vores requirements er opbygget af krav vi forventer er realistiske at nå inden for vores tidsramme. Eftersom vores mandefald i gruppen, har vi valgt at det er acceptabelt at ikke lave rollerne funktionelle, så at kravene hedder “Som Project creator, el. Scrum master mm.” fungere, men ikke med den specifikke rolle. Derudover har vi nedprioriteret krav vi tror er urealistiske inden for tidsrammen, så vores kravliste har rækkefølgen fra vigtigst og realistisk til uvigtigt og urealistiske.

2.1 Functional Requirements

Critical priority:

1. Som projekt creator, skal jeg kunne oprette nye projekter i systemet så der kan holdes styr på hvad der bliver arbejdet på.
2. Som projekt creator, skal jeg kunne tilsætte og fjerne ansatte på projekter så det er klart hvad de ansatte skal arbejde på.
3. Som projekt creator, skal jeg kunne fjerne, redigere eller sætte en rolle på de ansatte, så de ved hvad deres opgave er (Som Product Owner og Scrum Master).
4. Som projekt owner, skal jeg kunne redigere projekternes krav. Her skal jeg kunne give et krav en ‘approved’, eller ‘rejected’.
5. Som projekt owner, skal jeg kunne tilføje, fjerne og prioriterer kravene, samt tildele funktionel eller non-funktionel rolle.
6. Som kunde af hjemmesiden, skal jeg have adgang til oplysninger, såsom opgavebeskrivelse og alle dets krav med status, på min bestilte opgave.

High priority:



7. Som Scrum master, vil jeg kunne få oplysning om hvorvidt arbejdet på et krav er færdig, for at kunne rapportere til product owner at kravet skal valideres.
8. Som Scrum master, skal jeg kunne rapportere når et krav er færdig så project owner kan validere kravet.
9. Som bruger af systemet skal jeg kunne se vigtige informationer om projekterne såsom start og slut dato, deadline, proces, vurderet antal tid brugt på projektet, individuelle opgaver og medlemmer.
10. Som kunde skal hjemmesiden være lavet i et responsivt design, så jeg som kunde har adgang til hjemmesiden, på alle mine enheder.

Low priority:

11. Som bruger af administrator programmet, er det vigtigt for mig, at kunne se tidligere færdiggjorte projekter, ved søgning.
12. Som Scrum master af administrator programmet, er det vigtigt for mig at team members tid bliver registreret så jeg kan hjælpe arbejdes processen.
13. Som bruger af administrator programmet, er det vigtigt at kunne søge på medarbejdere, og hans relaterede projekter, produktivitet, hvem de har arbejdet mest med, tid brugt på et projekt og deres status.
14. Som bruger af administrator programmet, er det nødvendigt for mig, at jeg kan søge efter projekterne, og få det korrekte resultat.

2.2 Non-Functional Requirements

15. Hjemmesiden skal være supportet af browserne: Microsoft Internet Explorer 9, 10 and 11, Google chrome 28-47, Mozilla Firefox 32, Microsoft Edge 20.
16. Kun 1 person adgangen, skal kunne bruge administrator programmet.
17. GUI skal være lavet med JavaFX, så Colour IT senere selv kan redigere det.
18. Systemet skal automatisk gemme dataet ved ændring



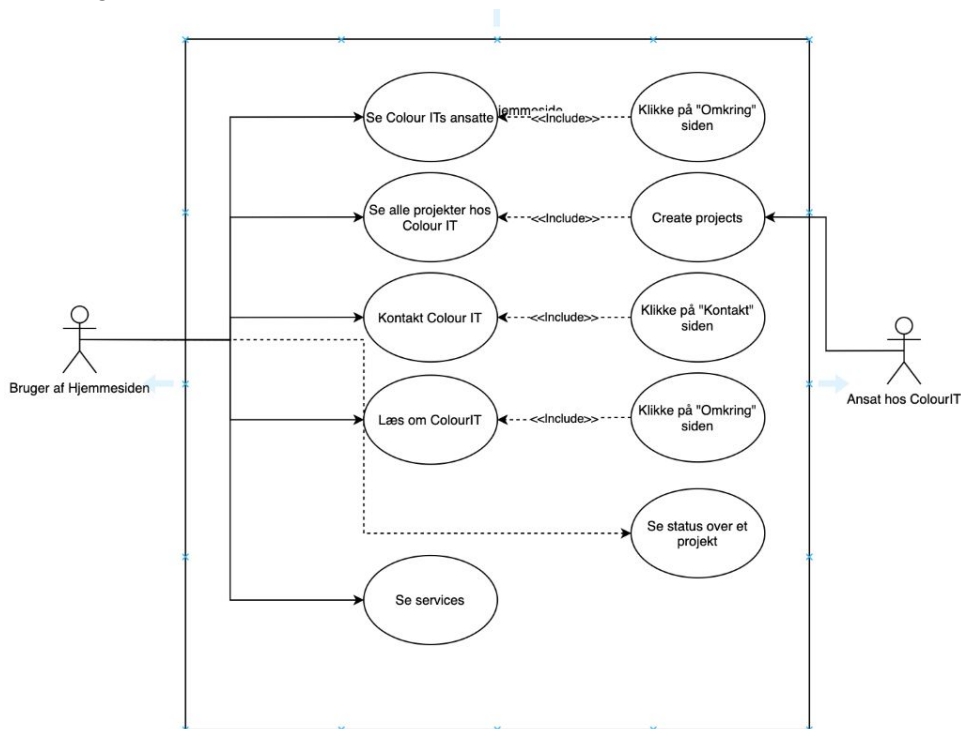
19. Hjemmesiden skal ikke bestå af for meget tekst, men nok til at kunden får en ide om hvad Colour IT arbejder med.
20. Systemet ændre staten af et krav når alle opgaver er færdigt i kravet til 'ended' staten.
21. Hjemmesiden skal være i flere forskellige farver, have et organiseret layout med god overskuelighed, det skal være nemt at finde sin ønskede 'under'-side.

3 Analysis

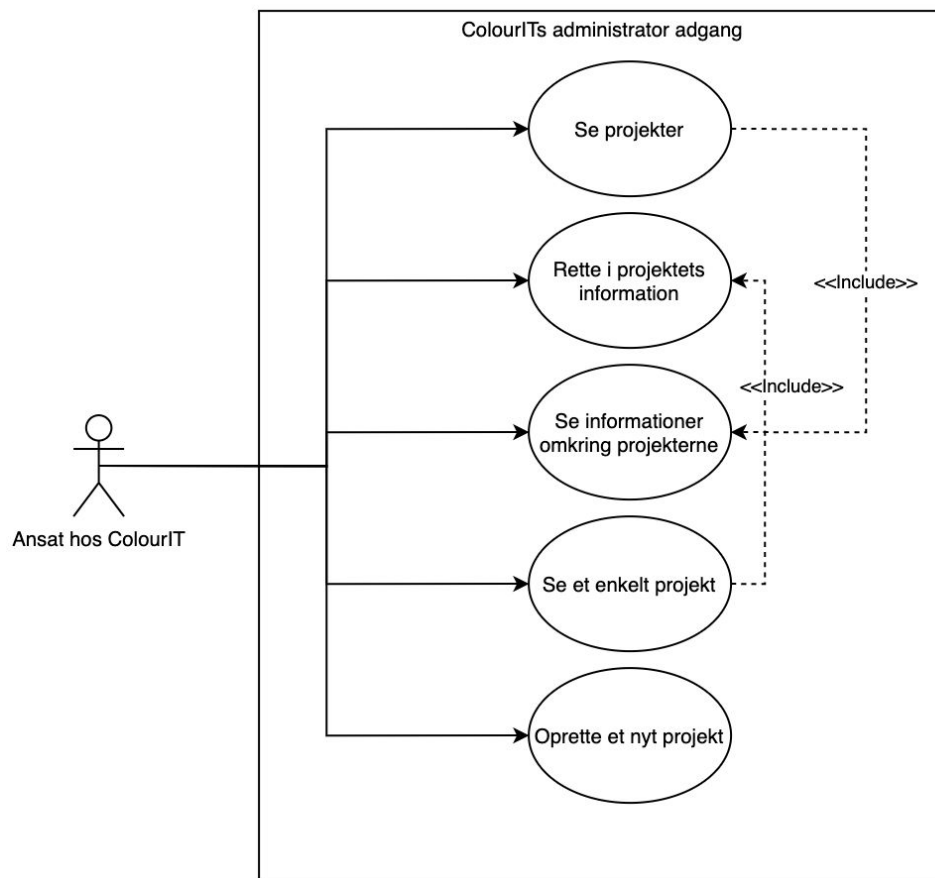
Vi vil her kigge på hvordan programmet virker fra en brugers perspektiv. Vi vil her komme ind og kigge på nogle af de aktiviteter en bruger kan komme ud for.

3.1 Use Case Diagram

Use case diagrammer er brugt til at visuelt repræsentere de forskellige typer af brugere, her er det Bruger af hjemmeside, og administrativ på JAVA produktet. I JAVA produktet er alene yderligere 4 roller fra SCRUM, 'Scrum master', 'Project Creator', 'Product Owner' og 'Team member'. Dette besidder produktet dog ikke, da vi nedprioriterede rolle specifikke handlinger.



Se bilag "UseCaseDescription" for flere



Se bilag "UseCaseDescription" for flere



3.2 Use Case Descriptions

Use Case Descriptions simple tabeller der viser vejledning mellem A til B med specifikke handlinger og instruktioner. Nedenunder har vi vist to af de grundlæggende handlinger for vores produkt.

Alle Use Case Descriptions kan ses i Appendix "UseCaseDescription".

3.2.1 Hjemmesiden

Use Case	Se Colour ITs ansatte
Summary	Åben hjemmesiden, klik på "Our Team", og se der de ansatte.
Actor	Bruger af hjemmesiden
Precondition	At hjemmesiden kører optimalt
postcondition	
Base sequence	<ul style="list-style-type: none">• Åben hjemmesiden• Klik på "Our Team"• Se her de ansatte
Exception sequence	
Note	

Figur X, Use Case Description



3.2.2 Java delen

JAVA DELEN:

Use Case	Opret et projekt
Summary	Opret et projekt, opret krav, og ansæt kollegaer.
Actor	Project creator
Precondition	Projekt creatoren har snakket med kunden om ønskede krav og funktionaliteter.
postcondition	Et nyt projekt er oprettet, med deadline og kravene fra samtalen med kunden.
Base sequence	<ul style="list-style-type: none"> • Åben systemet • Vælg rolle • Åben 'Projekter' • Klik på "Tilføj projekt" • Udfyld inputs med Projekt navn, deadline, projekt ejer, og de kendte start krav. • Tilføj kollegaer på projektet og vælg de tilsvarende roller. • Klik tilføj projekt, i bunden af siden
Exception sequence	
Note	

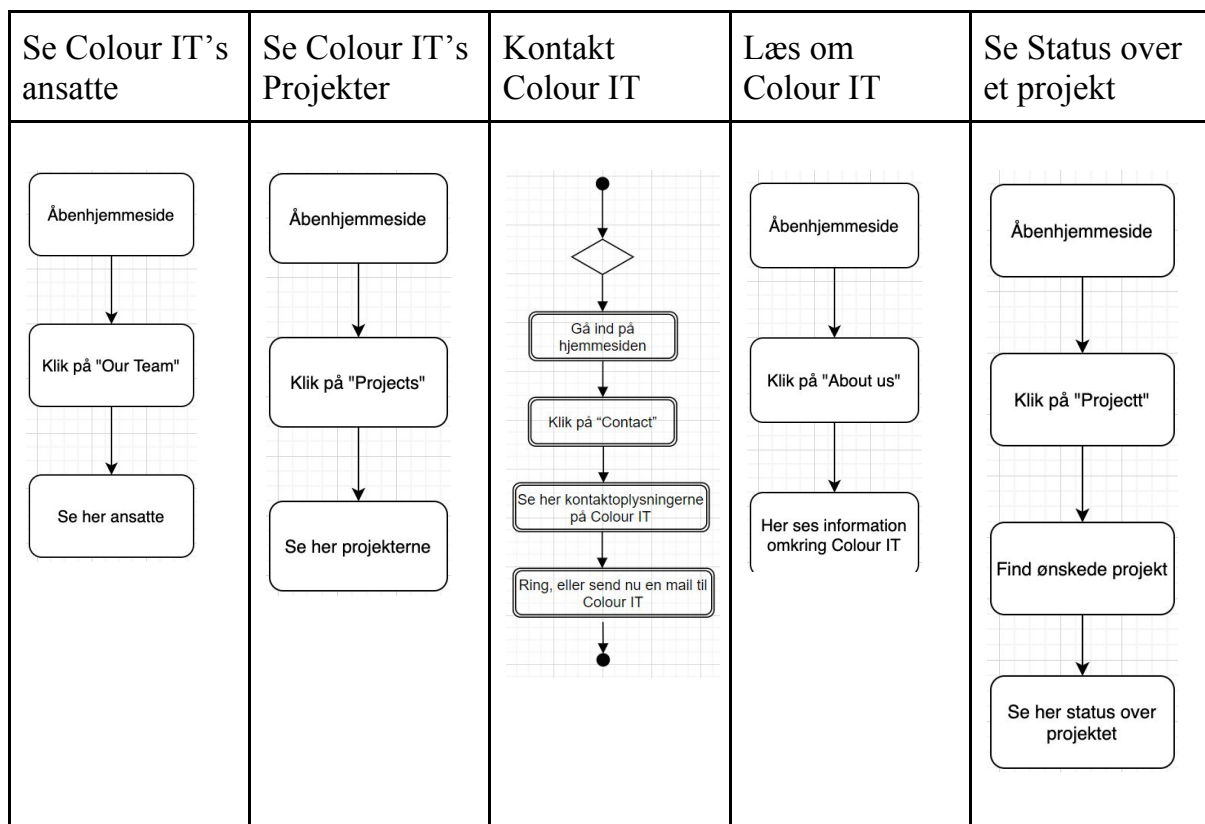
Figur X, Use Case Description



3.3 Activity Diagram

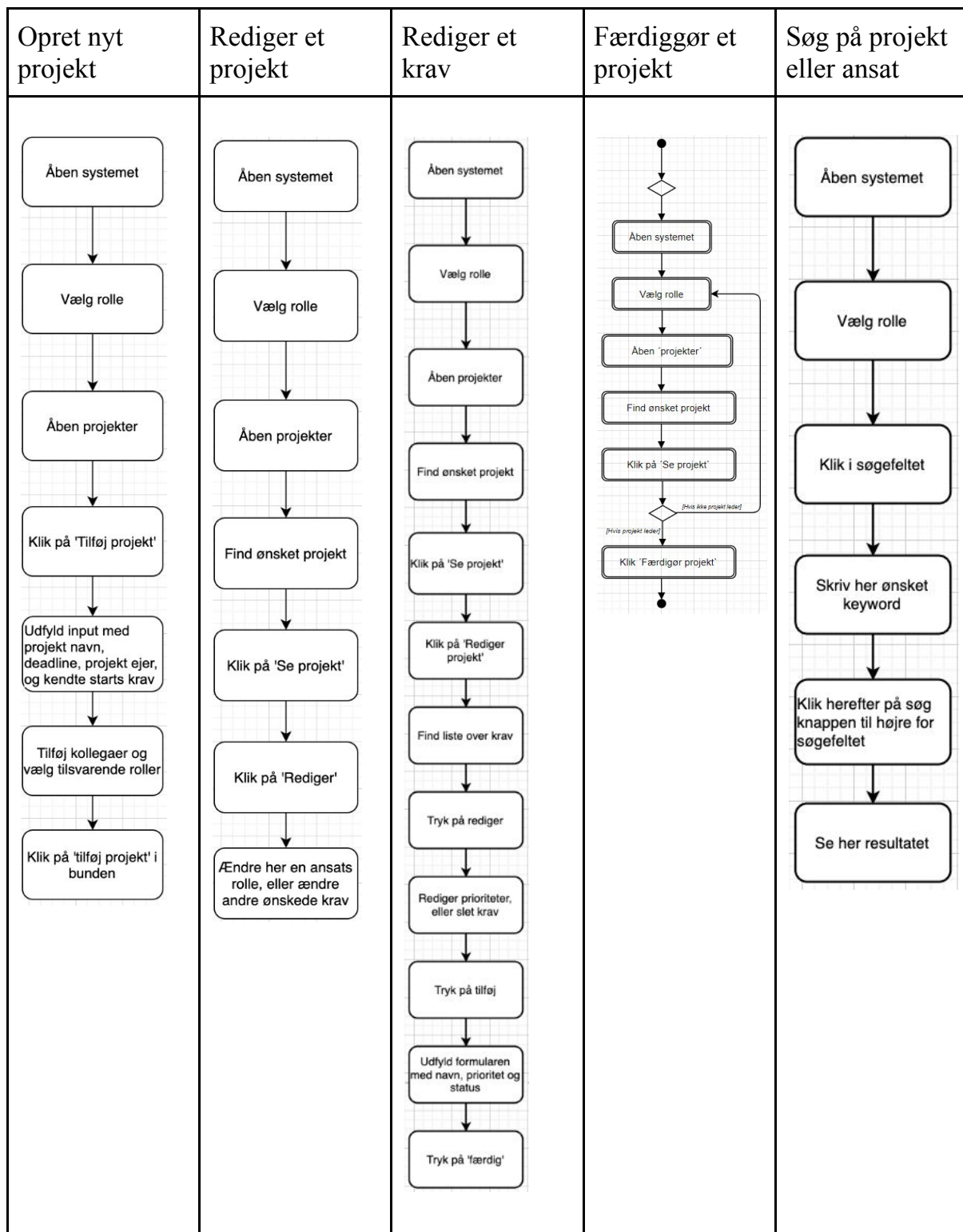
Aktivitetsdiagrammer for projektet ses under. Aktivitetsdiagrammer er en visuel repræsentation af hvad Use case Descriptions beskriver. Dette kan gøre det mere forståeligt for kunden at se, frem for tekst. Aktivitets diagrammerne viser hvordan man kommer fra A til B i det gyldige program.

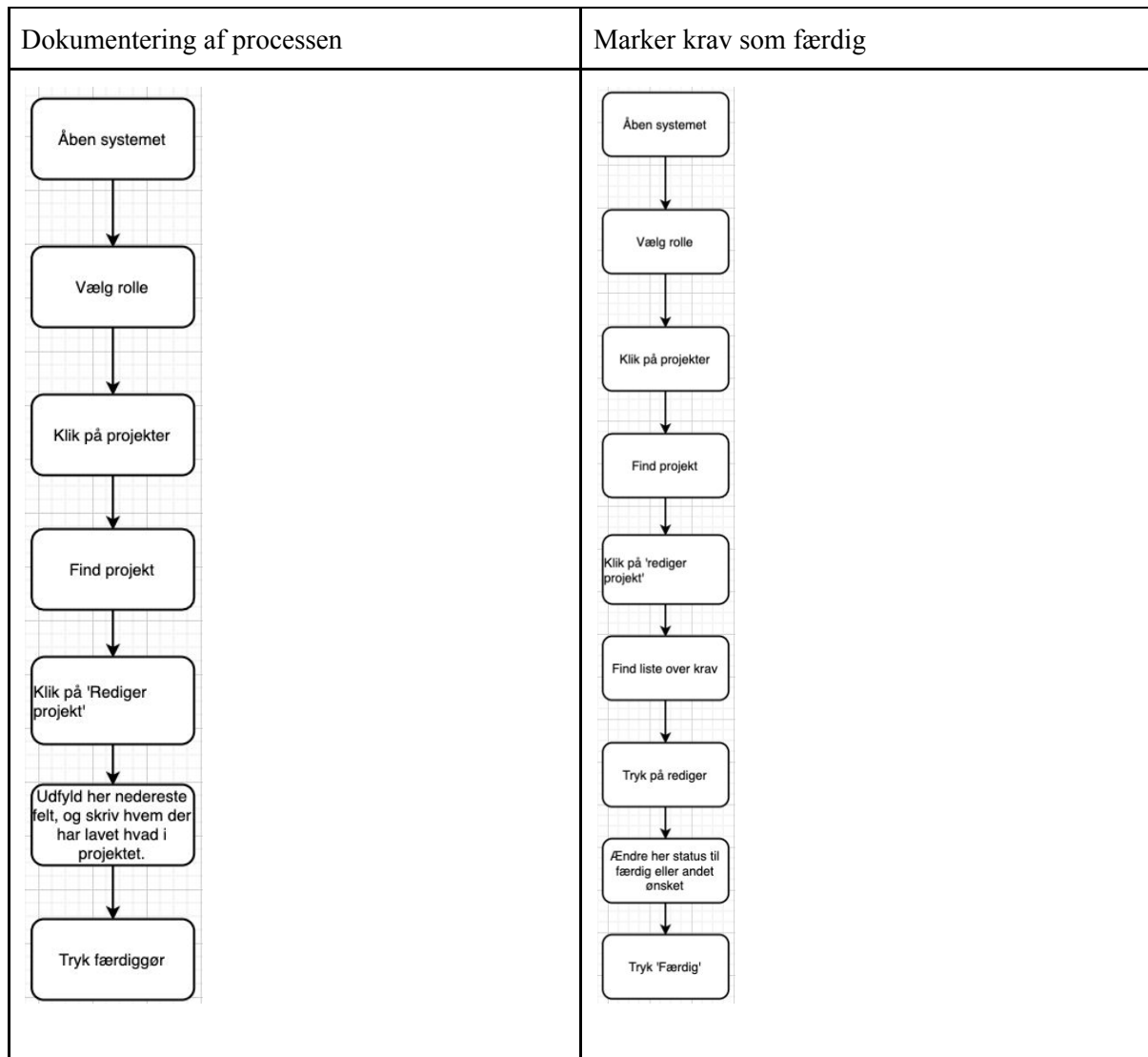
Herunder ses hjemmesiden





SDJ Systemet

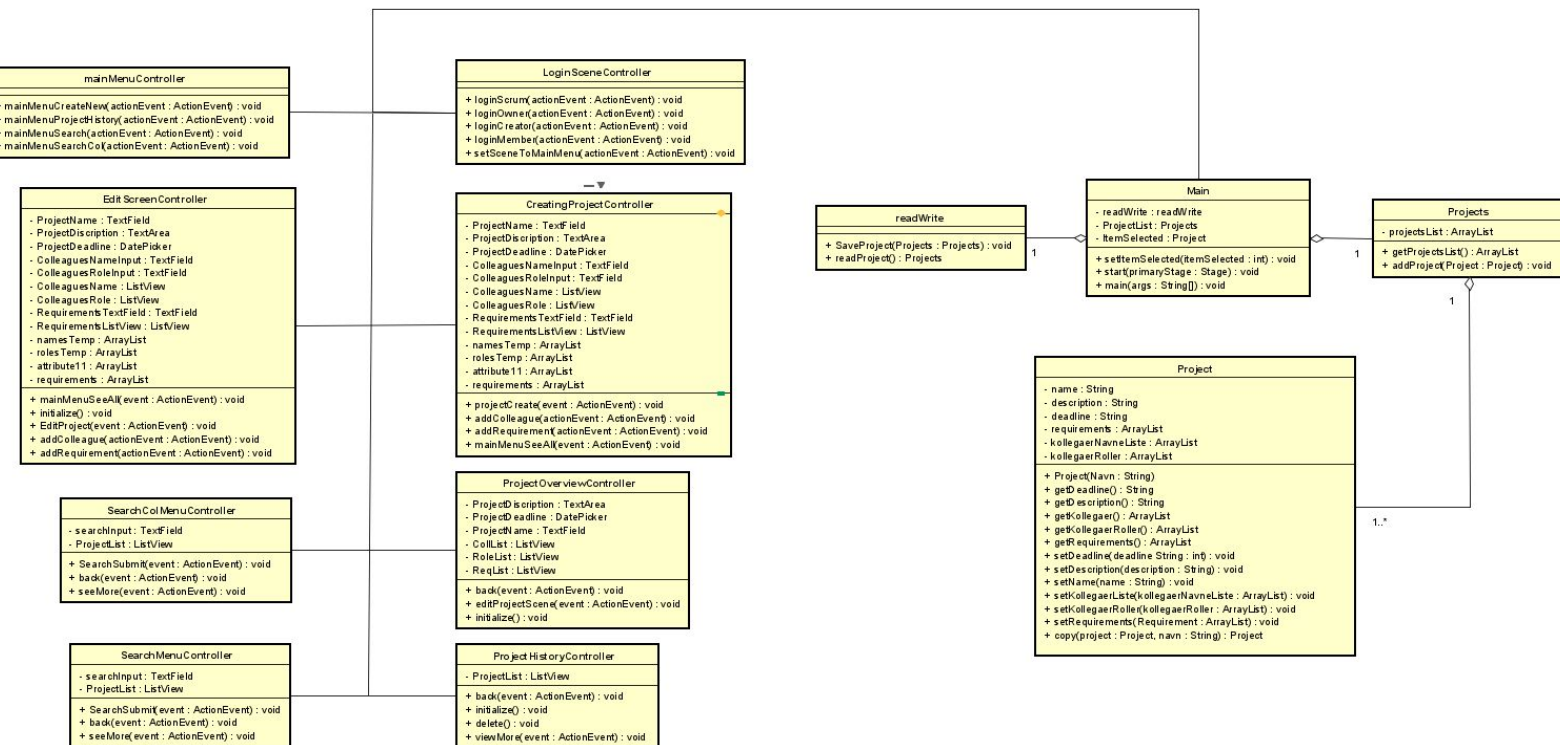






3.4 UML Diagram

Herunder ses vores UML program for det færdige JAVA produkt. En større version af UML diagrammet kan findes i Appendix 1



Figur 1, færdige UML diagram.



4 Design

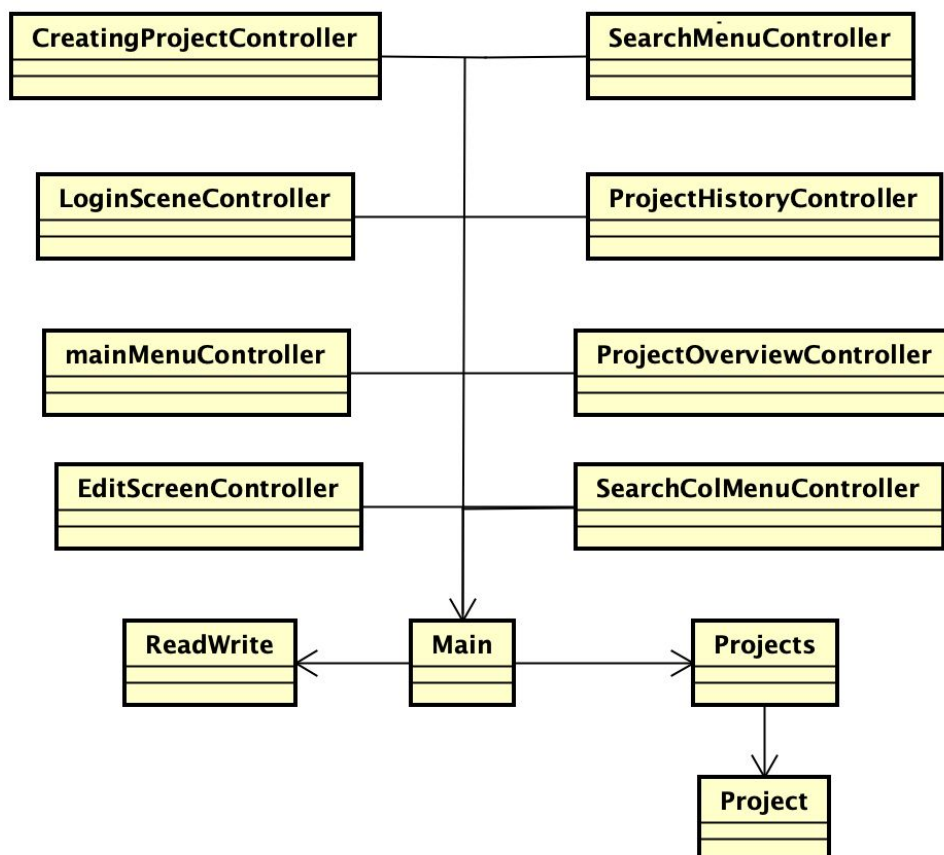
I det endelige produkt er navigationen opbygget af en halv MVC model.

MVC står for Model, View, Controller. Vores struktur er en halv MVC model, idet der mangler ViewHandler eller View delen af MVC modellen. Dette er skyld vi først fandt ud af det sent i processen. Vores Controller del af MVC er vores individuelle Controller's for hver af vores scener, her produktet er de også indbyrdes ansvarlig om View funktionerne. Model delen, er vores Project, hvilket er alt logikken bag programmet.

4.1 Class Diagram / Domain model

Class diagrammet el. klasse diagrammet kan ses under (figur 2) uden dens felt værdier og metoder. Klasse diagrammet viser relation mellem klasserne, og hvordan de påvirker hinanden.

Alt brugeren har behov for at se ligger samlet i vores 'project' klasse, hvor så alle vores dem er samlet i 'projects' som reelt set bare er en ArrayList.

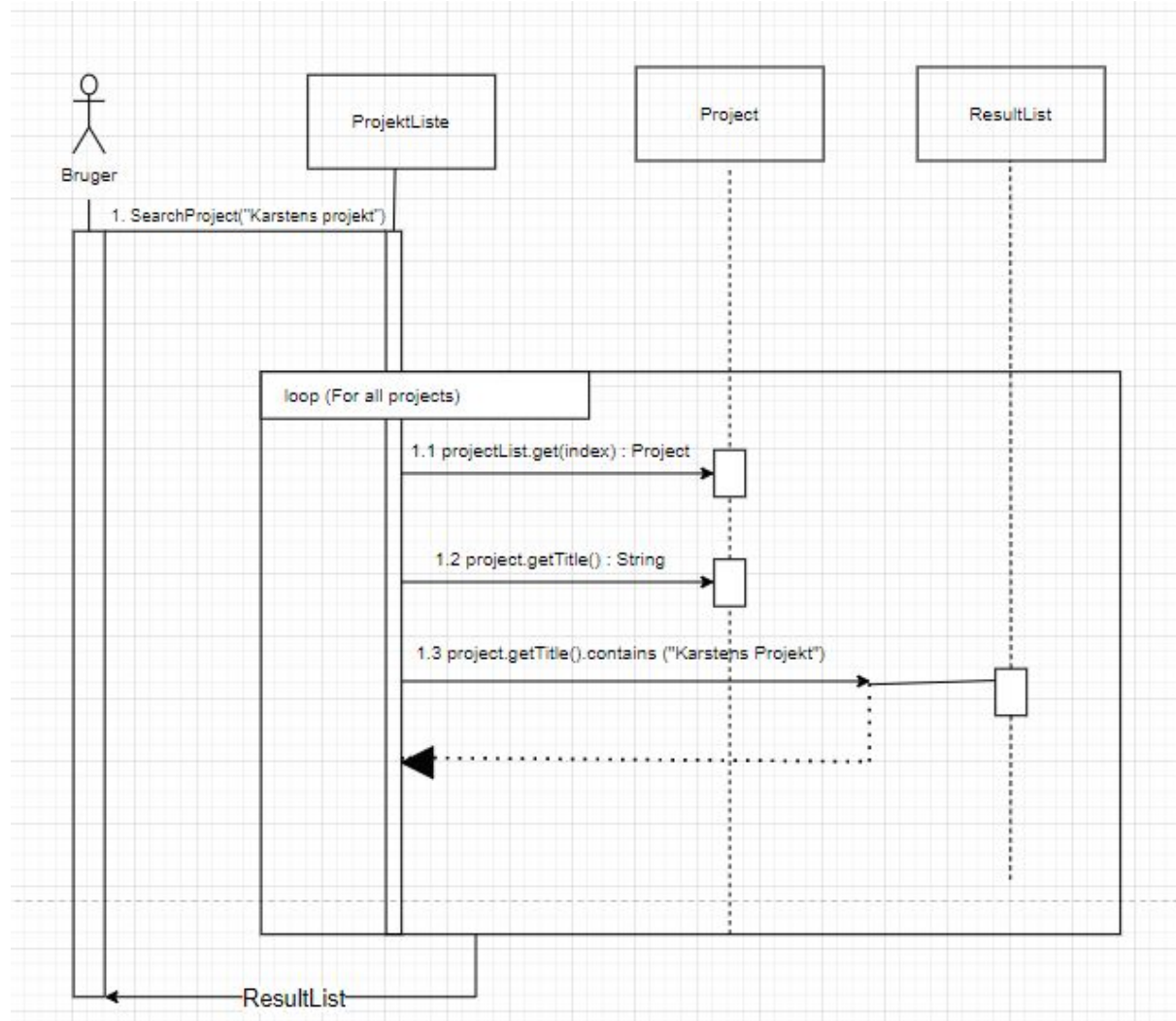


Figur 2. Class Diagram (ingen metoder)

4.2 Sekvensdiagram



Sekvens diagrammet viser hvordan klasserne og objekterne handler sammen i den rigtige rækkefølge. Vi kigger her på hvordan søgning i vores program vil virke. Vi kan se på billedet under, at brugeren indtaster et søgeord, en String. Denne bliver nu sammenlignet alle navnene på projekterne. Hvis et projekts navn indeholder søgeordet, tilføjes dette projekt til "ResultList" arrayet. Når dette loop har kørt igennem alle Projekterne, sidder brugeren tilbage med et array over alle de matchende værdier.



Figur 3 Sekvensdiagram

Denne søgefunktion bruges også når vi søger mellem kollegaer på et projekt. Her er forskellen blot listen der sammenlignes med. Her bruges nemlig en liste med alle kollegerne fra alle projekterne.

4.3 Hjemmesidens design

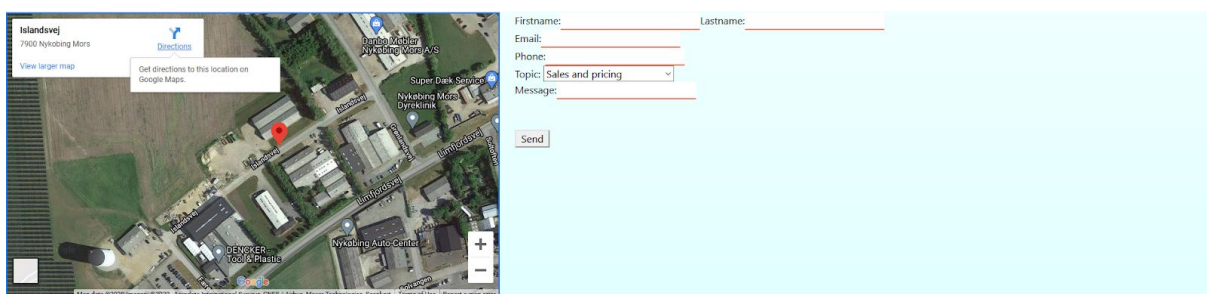
Designet af hjemmesiden er baseret på firmaet navn 'Colour'. Vi har prøvet at skabe et brand design til dem, og prøvet at holde det konsekvent gennem hele hjemmesiden. Dette har vi gjort for var det et rigtigt firma, ville man også gerne have genkendelighed gennem f.eks. design. Nedenunder ses et



billede af hvordan landingssiden (Den side man først ser) ser ud. Lignende sider finder man også på hver af hjemmesidens undersider, dog med andet billede men samme farverige tema kørende.



Kontakt siden på hjemmesiden, har vi gjort et lidt interessant valg i at implementere et google maps overblik. Nålen på kortet er placeret på et tilfældigt sted, dette er gjort da Colour IT er funktionelt, og derfor ikke har en lokation. Dog beholder vi kortet, da hvis det var et Reelt firma, kunne dette være meget relevant for både bruger og Colour IT. Ved siden af Kortet har vi en meget simpel formel, og igen da vi ikke har nogle steder at sende dataet hen som f.eks. en email, er dette også kun opsat for eventuel grund, og kan altid fjernes hvis ønskes. Det at vi ikke har opsat dette er en stor del af arbejdet, så dette betegner vi ikke som en virkende kontaktside. Alt dette er selvfølgelig også lavet med bootstrap, så det virker funktionelt på alle skærmstørrelser.



Vores projekt side på hjemmesiden understøtter brugen af billeder. Derudover er der også adgang til general information omkring projekterne så som krav, noter, og status pr. individuell krav, men også en til projektets helhed. Alt data er i den færdige version placeholder tekst, og ville i praksis blive udfyldt af den administrative bag Colour IT. Denne side er statisk, i det at når man ændre noget i JAVA programmet, ændres hjemmesiden ikke. Der er dog mulighed for at lave en sådan løsning, men vi har her i produktet det ikke.



COLOUR IT

[About us](#) [Our Team](#) [Service](#) [Projects](#) [Contact](#)

Project #1

Lorem ipsum dolor sit amet consectetur adipiscing elit. Ut illum expedita et assumenda laborum! Non praesentium fugiat aspernatur, doloremque minima accusantium fugit molestiae vel rerum ipsam veniam voluptatem, est natus.

50%

Krav	Notes	Status
X1	No current notes	50% done
X2	No current notes	100%



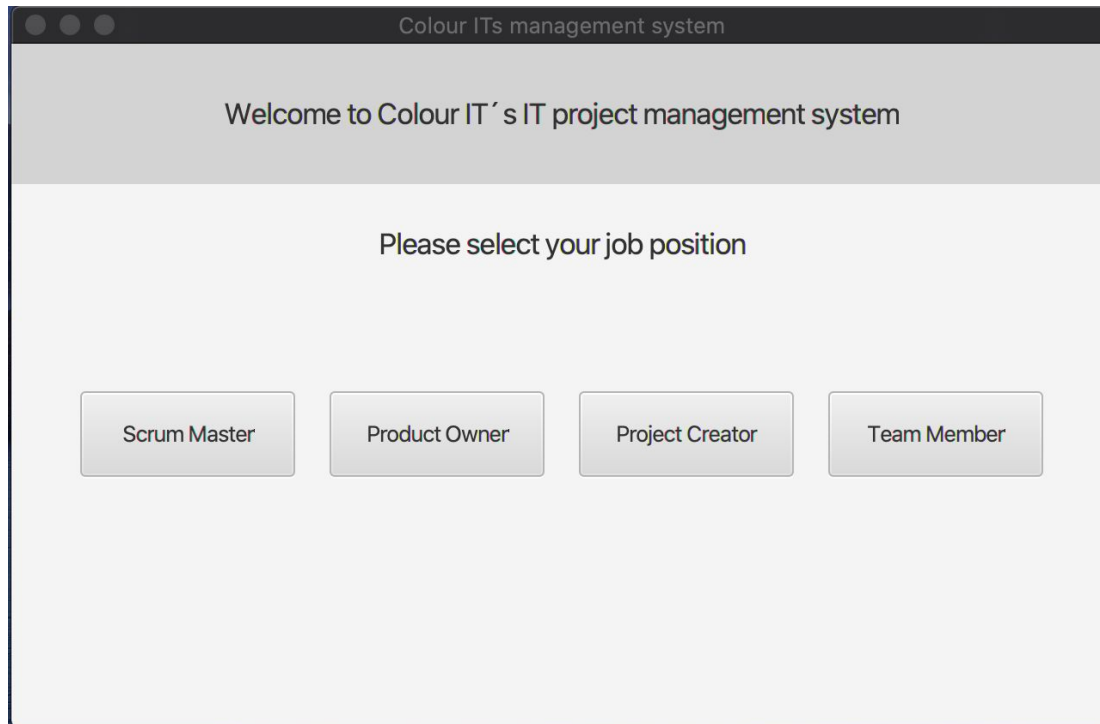
Project #2

Lorem ipsum dolor sit amet consectetur adipiscing elit. Ut illum expedita et



4.4 GUI design

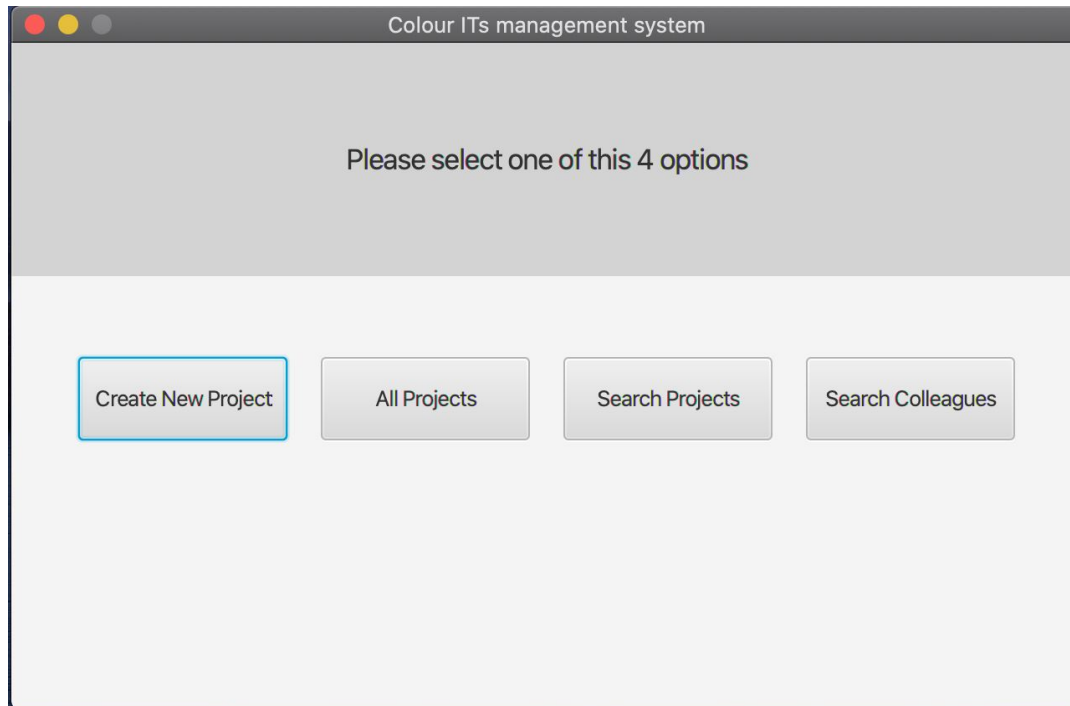
Når man først åbner programmet, bliver man sat på denne “login” skærm. Her skal man blot vælge sin funktion i firmaet. Alle disse 4 knapper fører en videre til Main scenen, eller home scenen. Selve designet på denne side, er meget simpelt og minimalistisk. Der er ikke nogen farver, og siden kan godt virke lidt kedelig. Dog har vi her, og generelt igennem hele SDJ programmet valgt at fokusere på det funktionelle frem for det design mæssige aspekt.



På Main scenen får man her mulighed for at vælge imellem 4 knapper. Her er knapperne følgende;



Create New Project, All Projects, Search Projects og Search Colleagues.



Klikker man på Create New project, kommer man til følgende scene. Her har man mulighed for at udfylde detaljer som;

- Project Name
- Deadline
- Colleagues
- Colleagues Roles
- Requirements
- Description


Klikker man herefter “Create New Project” vil dette projekt blive gemt, i en binær fil, så projektet bliver gemt på trods af genstart.



Colour ITs management system

Creating a new Project

Project name:

Deadline: 

Colleagues

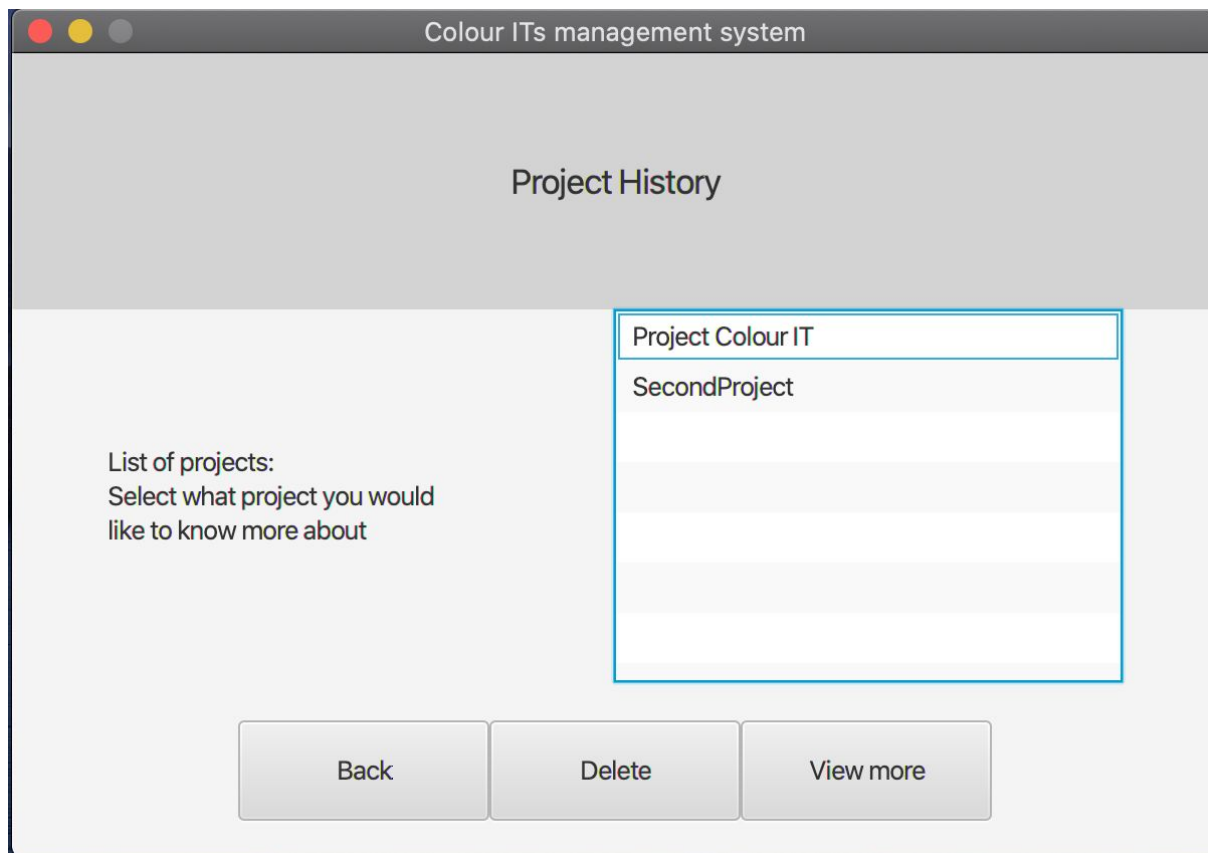
Colleague Name	Colleague Role	
<input type="text"/>	<input type="text"/>	<input type="button" value="Add Colleague"/>

Requirements

Requirement	
<input type="text"/>	<input type="button" value="Add requirement"/>

Project description:

Klikker man herefter Back, kommer man til hovedmenuen. Klikker man her på næste knap, "All Projects" Kommer man til følgende scene;



Her ses alle programmets projekter, i en tabel. Tilføjes der så mange projekter at disse køre længere ned en tabellen, tilføjes der en scroll mulighed. Det betyder altså at man her kan se alle projekter. Herudover er der også mulighed for at fjerne programmer herunder. Hvis dette skal ske, vælges et projekt og man klikker dernæst på “Delete” knappen. Her har vi en fejl i programmet, så få at projektet visuelt fjernes, skal man klikke back, og tilbage til denne scene. Dette er en detalje, men alligevel en detalje der er vigtig for brugeroplevelsen.

Vælger man et projekt og klikker på “View More” knappen, kommer man til følgende side;



Project Overview

Project name:
Colour IT Project

Project deadline:
2020-12-06

Project description:
Colour IT er et firma m...

Back

Configure

Colour ITs management system

All colleagues	Role	All requirements
Kasper Falk Mikkelsen	The supr	Lav rapport
Andreas Østergaard	Soldat	Lav process rapport
Maja	ingen ko	Lav program

Total progress

Her kan man se yderligere informationer om det valgte projekt. Her ses;

- Project Name
- Project Deadline
- Project Description
- Alle kollegaer og deres tilsvarende roller
- Og alle projektets krav

Ønsker man at redigere i disse oplysninger, klikker man på “Configure” knappen nede til venstre. Denne knap bringer en videre til følgende side.



Colour ITs management system

Edit only if necessary

Edit only what you want to change! Leave empty if you do not wish to change.

Project name: Colour IT project

Deadline: (must select) 2020-12-23

Project description: Colour it bla bla bla

Colleagues

Add a Colleague

Colleague Name: Colleague Role: Add C...

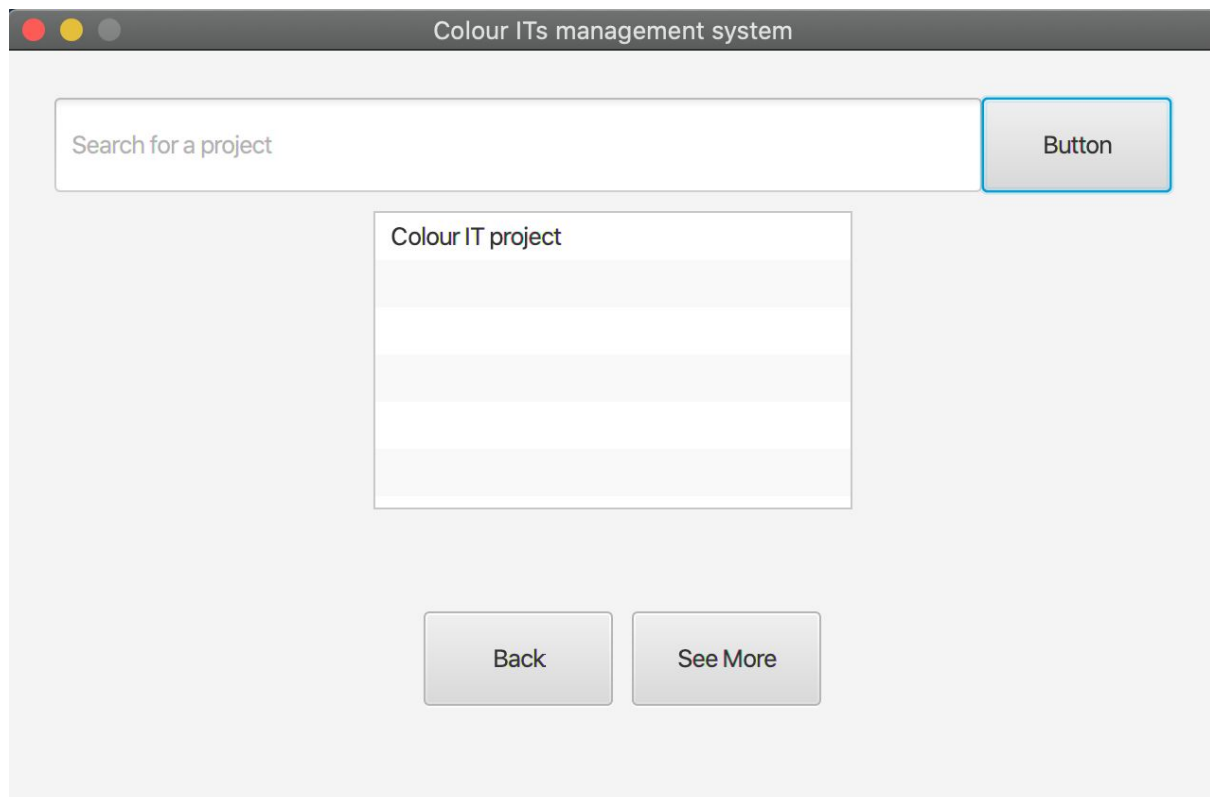
Requirements

Requirement: Add requirement

Back Edit

Her beskrives der i toppen hvad man skal gøre for at ændre noget. Der bliver her skrevet, at hvis man ikke vil ændre et felt, så lad det være tomt. Det vil sige, at hvis man for eksempel ikke vil ændre kollegerne, og deres rolle, skal man ikke udfylde den tabel.

Klikker man fra startside i stedet på "Search a project", kommer man til følgende siden;



Skriver man her et søgeord og klikker på knappen til højre, vises resultatet i tabellen under.

Klikker man på “See More” knappen kommer man ind og ser tidligere nævnte “project overview”, hvor man ser en oversigt over det valgte projekt.

Hvis man, på startsiden, klikker på “Search employee” har man her mulighed for at søge efter ansatte. Her vil programmet kigge i alle projekterne, og tager her alle ansatte på det projekt. Denne ligner dog fuldstændig billedet over.



5 Implementation

JavaFX biblioteket er brugt i vores kode for at kunne supportere vores scener bygget med Scene Builder. Biblioteket giver os adgang til at vise et grafisk interface.

```
import java.util.ArrayList;
```

Vi importer ArrayList i det vi ikke på forhånd kender hvor mange projekter, kollegaer, eller krav systemet skal kunne håndtere. ArrayList giver os muligheden for altid at udvide hvis nødvendigt.

```
import java.io.Serializable;
```

Serializable bliver brugt til at gemme vores objekter gennem vores ReadWrite.java fil. Serializable siger at klassen den er brugt i skal bruges til Serialization, hvilket er et krav for at ObjectOutputStream og ObjectInputStream fra ReadWrite.java skal virke.

Noget lidt interessant vi har gjort omkring vores gemning af data, har været at vi har udelukkende brugt en metode, og kun et slags objekt. Vores 'ReadWrite' fil består af af skærbilledet under. Eftersom vores program er opbygget omkring en ArrayList i projects, med ArrayList inde i projekt, giver det os muligheden for bare at gemme det ene ArrayList, fordi det indeholder objekter med yderligere ArrayList relevant for objektet det er i. Dette samler alt relevant information i et objekt, og gør det mere overskueligt at finde andre relevante værdier hurtigt, frem for hvis vi gemte en liste for hver forskellig (feks. SaveKollega, SaveRequirement). Kort forklaret virker vores saveProject, og readProject næsten ens. Vi specificere først filnavnet her hedder den "Database.bin", konstruere et File objekt. Derefter åbner vi en 'stream' der tillader os at sende data frem til filen, det er et output fra vores klasse, til Databasefilen. Efter vi har sendt vores objekt lukker vi streamen.

```
5 public class ReadWrite {
6
7     public void SaveProject(Projects projects) throws IOException {
8         String filename = "Database.bin";
9         File file = new File(filename);
10
11         FileOutputStream fos = new FileOutputStream(file);
12         ObjectOutputStream out = new ObjectOutputStream(fos);
13
14         out.writeObject(projects);
15         out.close();
16     }
17
18     public Projects readProjects() throws IOException, ClassNotFoundException {
19         String filename = "Database.bin";
20         File file = new File(filename);
21
22         FileInputStream fis = new FileInputStream(file);
23         ObjectInputStream in = new ObjectInputStream(fis);
24
25         Projects project1 = (Projects) in.readObject();
26
27         in.close();
28         return project1;
29     }
30 }
```



Search funktionen

```
@FXML
private TextField searchInput;
@FXML private ListView ProjectList;

public void SearchSubmit(ActionEvent event) throws IOException, ClassNotFoundException {
    ArrayList<Project> result = new ArrayList<>();
    if(searchInput.getText() != "") {
        System.out.println("Search Input:" +searchInput.getText());
        for (int i = 0; i < Main.ProjectsList.getProjectsList().size(); i++) {
            if(Main.ProjectsList.getProjectsList().get(i).getName().contains(searchInput.getText())) {
                result.add(Main.ProjectsList.getProjectsList().get(i));
            }
        }

        for (int i = 0; i < result.size(); i++) {
            System.out.println(result.get(i).getName());
        }

        //System.out.println(Main.ProjectsList.getProjectsList().get(4).getName());
        searchInput.setText("");
    }
    ProjectList.getItems().clear();
    for(int i = 0; i < result.size(); i++){
        ProjectList.getItems().add(result.get(i).getName());
    };
}
```

På billedet over, ses den funktion vi bruger når vi søger efter et projekt. Denne funktion bliver kørt, når brugeren klikker på søg knappen på den scene.

I starten af funktionen laver vi en ny array liste til vores resultater. Derefter tjekker vi om vores søgeord ikke bare er en tom string. Hvis dette ikke er tilfældet, fortsættes programmet ind i et loop, som kører igennem alle projekterne. Hver projekts titel, bliver sammenlignet med søgeordet. Hvis der her er et match, tilføjes dette projekt, til resultats listen. Herefter printes de først ud i console. Herefter sættes søgefeltet til en tom string.

Til sidst tilføjes alle projekterne til den tabel der ligger under søgefeltet. Dette gøres i et for loop, der kører gennem hele resultats listen. Her tilføjes hvert projekt til tabellen.



For at få hjemmesiden til at virke skal vi inkludere nogle absolut nødvendige tags, for at fortælle browseren hvilken fil det er og hvordan den skal læses (). I vores head tag, har vi dog også tags som script, som tillader os at bruge biblioteket fra andre steder fra. Her bruger vi <Script> til at kunne bruge bootstrap og ajax på vores hjemmeside.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="stylesheet" href="css/style.css" />
    <link href="https://fonts.googleapis.com/css?family=Raleway" rel="stylesheet"/>
    <link href="css/bootstrap.min.css" type="text/css" rel="stylesheet"/>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no"/>
    <title>COLOR IT</title>
  </head>
```

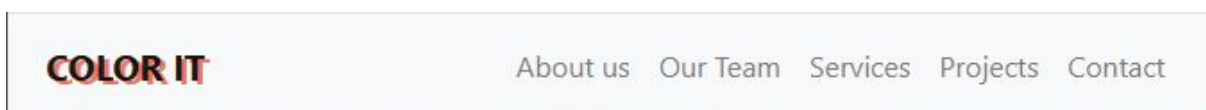
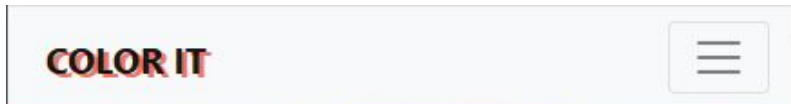
Vi har brugt bootstrap på hjemmesiden for at hurtigt kunne gøre den responsiv (ie. at den virker på alle skærmstørrelser), resultatet kan også opnås med ren HTML og CSS, men dette ville være et kæmpe stykke arbejde uden bootstrap. Derudover får vi også smarte og enkle løsninger til vores navbar, og billedkarrusel med bootstrap.

Vi har også lavet et lidt interessant valg og inkluderet en font, dette er selvfølgelig ikke essentielt, men beslutningen har udelukket været baseret efter subjektiv smag.

```
14 <!-- ANDREAS -->
15 <body>
16   <nav class="navbar navbar-expand-sm bg-light navbar-light">
17     <a class="navbar-brand" href="index.html">COLOR IT</a>
18     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#myNavbar">
19       <span class="navbar-toggler-icon"></span>
20     </button>
21     <div class="collapse navbar-collapse" id="myNavbar">
22       <ul class="navbar-nav">
23         <li class="nav-item">
24           <a class="nav-link" href="about.html">About us</a>
25         </li>
26         <li class="nav-item">
27           <a class="nav-link" href="DevTeam.html">Our Team</a>
28         </li>
29         <li class="nav-item">
30           <a class="nav-link" href="Service_at_Colour_IT.html">Services</a>
31         </li>
32         <li class="nav-item">
33           <a class="nav-link" href="Projects.html">Projects</a>
34         </li>
35         <li class="nav-item">
36           <a class="nav-link" href="Contact_and_opening.html">Contact</a>
37         </li>
38       </ul>
39     </div>
40   </nav>
```



Vores navbar består som sagt af bootstrap, <button> tagget kommer kun frem når ‘data-toggle’ bliver aktiveret, og det gør det når brugeren skalere hans browservindue langt ned, eller han er på mindre enheder. Alt dette tager bootstrap sig af, så vi behøver ikke makke i @media tags i css’en. Øverst i min nav tag, skriver jeg klassen, og det er ikke en opdigtet men en ret specifik klasse liggende i bootstrap, som har prædefinerede farver og størrelser. Billede nedenunder er det endelige resultat af ovennævnte kode.



```
<!--Kasper Making Carousel-->
<div id="carouselForEmployees" class="carousel slide center-block" data-ride="carousel" data-interval="5000">
  <ol class="carousel-indicators">
    <li data-target="#carouselForEmployees" data-slide-to="0" class="active"></li>
    <li data-target="#carouselForEmployees" data-slide-to="1"></li>
    <li data-target="#carouselForEmployees" data-slide-to="2"></li>
  </ol>

  <div class="carousel-inner">
    <div class="carousel-item active">
      
      <div class="carousel-caption d-none d-md-block">
        <h3 class="text-align-left">Maja Møller</h3>
        <p>Has been in the company for 4 years</p>
      </div>
      <div id="Info">
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption d-none d-md-block">
        <h3 class="text-align-left">Andreas Østergaard</h3>
        <p>Has been in the company for 1 years</p>
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption d-none d-md-block">
        <h3 class="text-align-left">Kasper Falk Mikkelsen</h3>
        <p>Has been in the company for 5 years</p>
      </div>
    </div>
  </div>
</div>
```

I vores projekt, lavede vi en Bootstrap “Carousel”, som er et slide show, vi brugte til at vise alle firmaet ansatte. På billedet over kan vi se hvordan denne er lavet. Vi sætter den yderste div til at have klassen “carousel”, for at danne rammen for vores carousel. Det er også i denne div vi kan bestemme hastigheden på karrusellen, ved brug af “data-interval”. Hvis vi nu kigger på den først div, inde i vores



karrusel, har vi vores “Carousel Indicators”. Disse kommer også med bootstrap, og bruges til manuelt at skifte mellem billederne. Disse laver vi 3 forskellige af, da vi har 3 forskellige billeder. Her giver vi dem hver især deres tilsvarende værdi i “data-slide-to” variabelen, som bestemmer hvilket billede hver knap.

Under dette laver vi tre forskellige Divs, som vi bruger som containere for vores billeder. Heri lægger vi blot vores billeder, giver dem størrelser ved hjælp af klassen “img-fluid”. Derudover laves der også h3 felter, og p felter for at tilføje en forklarende tekst i bunden af billedet.

6 Test

Black box testing er set ud fra et end-user perspective, i dette tilfælde er det Colour IT, altså nogle der ikke kender til koden bag. Eftersom vi gennemføre alt testing selv, og ved hvordan det hele er sat sammen, er det ikke muligt for os at køre black box testing. Derfor er der her brugt white box testing (Guru99, ukendt). Denne metode brugte vi i endelsen af vores produktudvikling, for at tjekke at alle kravene er opfyldt. Vi har testet systemet, ved input af mange Kollegaer, roller og krav. Det er muligt at lave lange navne (Projekt navn, kollega navne, roller, krav, m.m.), dog når det gøres, genereres automatisk en scrollbar i bunden af det label, så det er stadig muligt at få læst alt teksten.

Requirements

Da vi føler vi har været under pres, da vi kun har været to, har vi valgt hovedsageligt at fokusere på Critical priority

Krav	Virker Ikke	Virker	Noter
1. Som projekt creator, skal jeg kunne oprette nye projekter i systemet så der kan holdes styr på hvad der bliver arbejdet på.		X	
2. Som projekt creator, skal jeg kunne tilsætte og fjerne ansatte på projekter så det er klart hvad de ansatte skal arbejde på.		X	
3. Som projekt creator, skal jeg kunne fjerne, redigere eller sætte en rolle på de ansatte, så de ved hvad deres opgave er (Som Product Owner og Scrum Master).		X	



4. Som projekt owner, skal jeg kunne redigere projekternes krav. Her skal jeg kunne give et krav en 'approved', eller 'rejected'.	X		Dette er en af de ting vi har nedprioriteret, da vores 3. gruppemedlem forlod gruppen
5. Som projekt owner, skal jeg kunne tilføje, fjerne og prioriterer kravene, samt tildele funktionel eller non-funktionel rolle.	X	X	Efter projektet er lavet først gang, er det muligt at redigere kravene - Dog er det ikke muligt at prioritere kravene
6. Som kunde af hjemmesiden, skal jeg have adgang til oplysninger, såsom opgavebeskrivelse og alle dets krav med status, på min bestilte opgave. (RWD)		X	

Non-functional Requirements

Krav	Virker ikke	Virker	Noter
15. Hjemmesiden skal være supportet af browserne: Microsoft Internet Explorer 9, 10 and 11, Google chrome 28-47, Mozilla Firefox 32, Microsoft Edge 20.		X	
16. Kun 1 person adgangen, skal kunne bruge administrator programmet.		X	
17. GUI skal være lavet med JavaFX, så Colour IT senere selv kan redigere det.		X	



18. Systemet skal automatisk gemme dataene på lukning		X	
19. Hjemmesiden skal ikke bestå af for meget tekst, men nok til at kunden får en ide om hvad Colour IT arbejder med.		X	Hjemmesiden indeholder ikke overvældende meget tekst, men nok til at brugeren får en forståelse af firmaet og dets projekter
20. Systemet ændre staten af et krav når alle opgaver er færdigt i kravet til 'ended' staten.	X		Vi har undladt kravenes status.
21. Hjemmesiden skal være i flere forskellige farver, havet et organiseret layout med god overskuelighed, det skal være nemt at finde sin ønskede 'under'-side.		X	Efter vores vurdering er dette opfyldt. Dette er dog subjektivt



6.1 Test Specifications

Nedenunder dette ses nogle forskellige aktiviteter, og om hvorvidt de kan lykkes i produktet

SDJ

Use case description	Works	Does not work
Opret et projekt	X	
Rediger et projekt	X	
Redigere Krav	X	
Færdiggør et projekt		X
Søg efter projekt	X	
Søg efter ansat	X	
Dokumentering af processen		X
Marker Krav som færdige		X

RWD

Use case description	Works	Does not work
Se Colour IT's ansatte	X	
Se Colour IT's Projekter	X	
Kontakt Colour IT		X
Læs om Colour IT	X	
Læs om services at Colour IT	X	
Se status over et projekt	X	



7 Results and discussion

Vi fik alt det kritiske med i vores projekt. Systemet vi har lavet, kan gemme og læse data fra forrige sessioner programmet har været brugt, vi har en simpel overskueligt GUI, og så har vi også implementeret projekter, krav, og kollegaer. Dog funktioner som rolle specificerede aktioner (såsom roller kan noget, andre ikke kan) mangles. Dette skyldes udelukket at arbejdsprocessen har været baseret ud fra vores krav, og så vidt muligt gået fra toppen og ned.

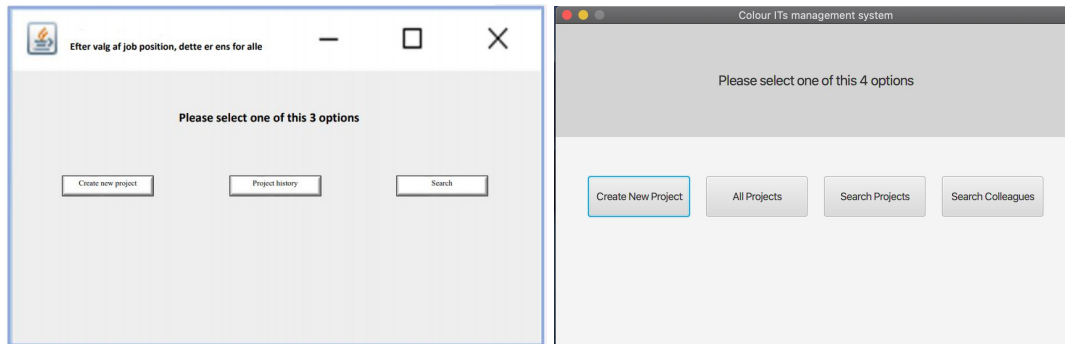
Havde vi haft mere tid, eller flere i vores gruppe havde vi med sikkerhed fået de sidste par features med, og det er en skam vi ikke har det.

Oprindeligt var det planlagt at vores hjemmeside og java program kunne snakke sammen, dette ville vi have opnået gennem AJAX, XML, og en parser. Vi oplevede dog mange problemer under at få dette implementeret, så vi valgte at det ville tage for lang tid frem for andre vigtige krav vi manglede at implementere. Så selvom hjemmesiden ikke opdatere automatisk, er det stadig muligt for den administrative side, at se alt data på vores JAVA program, og derefter skrive det manuelt ind i HTML.

Baseret på vores design valg i JAVA, er der plads til forbedring. Generelt er strukturen vedligeholdet i JAVA programmet, dog nogle sider som 'create project' (se figur 4) endte ud i at blive langt mere rodet end planlagt. Der skulle have brugt mere tid på Wireframing af vores JAVA program (se figur 5, for alle udkast se appendix "Sketches_for_Scenebuilder"), og ikke mindst diskuteret med hinanden noget mere om det, da det primært var en persons ansvar at lave det. Nedenunder ses 2 eksempler på hvordan vores program ser ud, til højre ses vores forventede sketches, og til venstre ses vores faktiske program.

Figur 5

Figur 4



En simpel ting som Search scenen blev udeladt da det virkede u-konstruktivt at lave et nyt vindue med kun to knapper i, imens vi sagtens kunne implementere det på den første 'main screen'. 'Creating new Project' scenen ser væsentligt anderledes ud end den forventede sketch. Hovedårsagen til dette er at implementations problemerne ved at indsætte tabeller i tabeller. Derudover er layoutet til 'Team members & Job position' i vores sketch til øverst venstre ikke særlig intuitivt, når det kunne i praksis stå som en String ved siden af navnet.

8 Conclusion

Hvis man kigger tilbage på vores endelige produkt, sidder man tilbage med et funktionelt produkt, der kan håndtere et firmaet projekter. Produktet man sidder tilbage med, er funktionelt med de mest grundlæggende elementer for lignende programmer, men mangler få low priority krav. Det endelige produkt består af en Hjemmeside, og en Java System. Java systemet kan kreere og redigere projekter, krav og kollegaerne, samt søge på forskellige relevante informationer gemt i produktet. Systemet gemmer automatisk input, og ændringer, så åbnes det på ny er sidste sessions informationer allerede i programmet. Det endelige Java produkt halter på pænere mere opskalerer kode, og få funktionelle funktioner.

Hjemmesiden giver brugeren et overskueligt overblik af Colour IT's projekter, programmørerne bag hjemmesiden, kontakt side, samt vigtige informationer omkring firmaet. Hjemmesiden er statisk, og information bliver ikke automatisk hentet efter ændringer i Java systemet.



9 Project future

I fremtidige projekter, skal der kigges på alle de nævnte funktionelle krav. Vi har i vores projekt ikke arbejdet på alle kravene. Her har vi i stedet kigget på dem vi mente var “Critical priorities”. Dette skyldes at vi i dette projekt, kun har været to medlemmer i gruppen. Vi mangler altså derfor, de fleste af vores, “high priority” og “low priority” krav.

Vi kunne have lavet et interface controller, for alle de metoder vores controllere har som minimum. Dette kunne have gjort delegering nemmere, og givet os en bedre mulighed for at se hvad var minimum i controlleren. Et andet smart valg vi kunne have gjort var at skrive vores kontrollere om, i det at vi gengiver meget kode (skærbillede eksempel under), når vi skal skifte scene. Man ville kunne have lavet en mere elegant løsning til det her, ved hjælp af for eksempel en switch eller en metode, der checker input og matcher controllerne med inputtet.

```
23 @ public void back(ActionEvent event) throws IOException {
24     Parent andetWindowParent = FXMLLoader.load(getClass().getResource( name: "mainMenu.fxml"));
25     Scene andetWindow = new Scene(andetWindowParent);
26
27     //Stage information
28     Stage window = (Stage) ((Node)event.getSource()).getScene().getWindow();
29
30     window.setScene(andetWindow);
31     window.show();
32 }
```

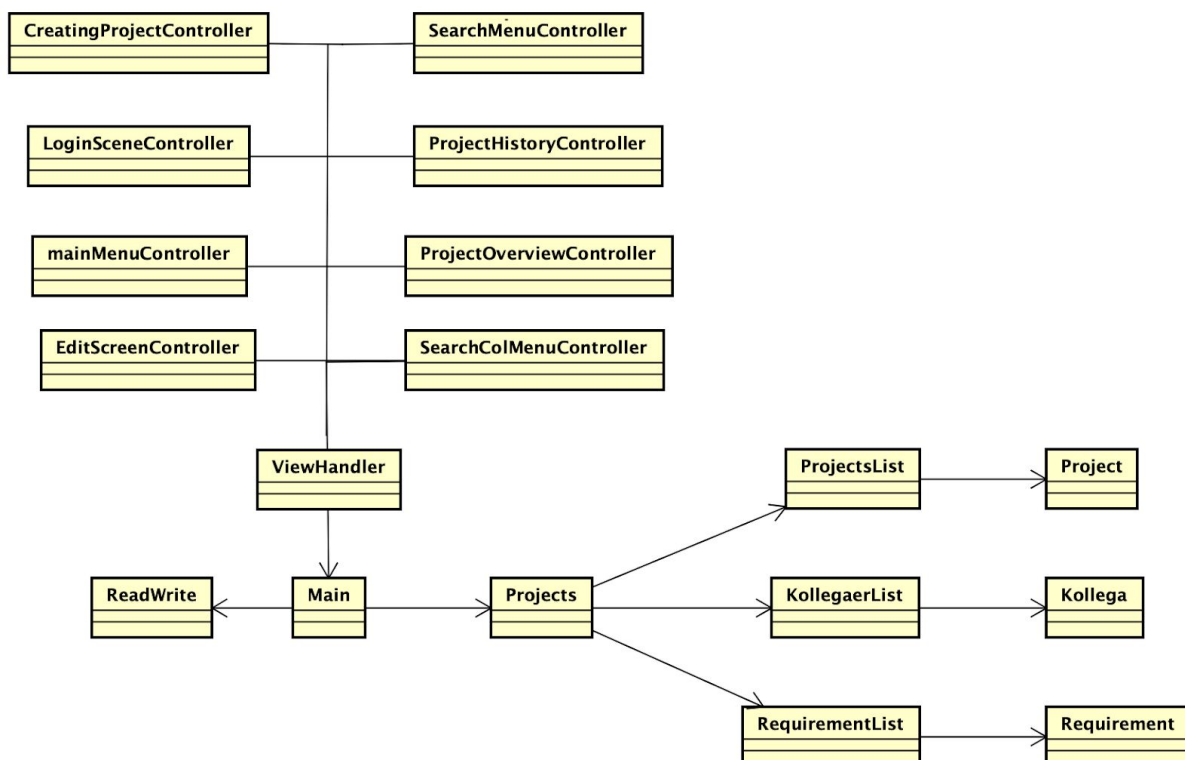
En anden ting vi også klart skal have lavet om i fremtiden, er måden vi håndterer de forskellige klasser. Ikke nok med det ikke er sorteret ordentligt mellem filerne, men også i stedet for vi har næsten alt data samlet i ‘project’ klasse, have det fordelt ud. Her er det relevant med en klasse for krav og kollegaer for sig selv. I den færdige version af programmet er alt omkring Krav og kollegaer opsat af String ArrayList i project klassen, og det tillader os ikke lave metoder på kollegaerne eller manipulere / give dem flere funktioner. Under ses et skærbillede af vores kode som det ser ud færdiglavet, og under er den mest optimale arkitektur, burde programmet have i fremtiden (figur 6 (inkludere også ovennævnte ViewHandler løsning)).



```

7   public class Project implements Serializable{
8       private String name;
9       private String description;
10      private String deadline;
11
12      private ArrayList<String> requirements = new ArrayList<>();
13      private ArrayList<String> kollegaerNavneListe = new ArrayList<String>();
14      private ArrayList<String> kollegaerRoller = new ArrayList<>();

```



Figur 6, den ideelle arkitektur

10 Sources of information

ItsLearning. Astrid Hanghøj. RWD fag

ItsLearning. Michael Viuff. SDJ fag

Oracle. Alla Redko. Sep. 2013. Using JavaFX UI Controls.

https://docs.oracle.com/javafx/2/ui_controls/list-view.htm

Guru 99. Ukendt. Black Box Testing Vs. White Box Testing: Key Differences.

<https://www.guru99.com/back-box-vs-white-box-testing.html>



11 Appendices

Appendix A Project Description



Se bilag "Project Description"

Se bilag "Sketches_for_Scenebuilder.pdf"

Se bilag "Use Case Description Tutorials.docx"

Appendix X UML Diagram

Se bilag "UML Diagram.docx"

Pågældende	Underskrift
Kasper Falk Mikkelsen - 304712	
Andreas Østergaard - 304546	

Ved underskrift accepteres følgende; 'Jeg godkender hermed at min projektgruppe og jeg har lavet projektet her, og at alle kilder af information er givet respektivt ud.'