



KRAVSPECIFIKATION & PROGRAMDESIGN

02161 SOFTWARE ENGINEERING 1


Afleveringsgruppe 13:

Rasmus Wiuff **s163977**

Mathies Henriksen **s200747**

Max-Emil Scotten **s204633**

Kasper Sylvest **s205281**

github.com/rwiuff/02161ExamProject 

8. maj 2023

INDHOLD

	Side
1 Kravspecifikation	1
1.1 Indledning	1
1.2 Ordliste	1
1.3 Use case diagrammer	2
1.4 Detaljerede use cases	3
2 Diskussion: Kravspecifikation	9
2.1 Oplægsmæssige overvejelser og afgrænsninger	9
2.2 Tekniske overvejelser	9
3 Programdesign	10
3.1 Klassediagram af programdesign	10
3.2 Sekvensdiagrammer	10
4 Diskussion: Programdesign	11
4.1 Datastrukturer	11
4.2 Klassestrukturer	11
Figurer	12
Tabeller	12
Listings	12

1 KRAVSPESIFIKATION

1.1. INDLEDNING

Denne del af rapporten dokumenterer de usecases og begreber som med udgangspunkt i projektoplægget, danner kravspecifikationerne for projektet. Use cases udvikles til Cucumber scenarier som igennem Behavior- og Test Driven Development bliver til den implementation som findes i anden rapport.

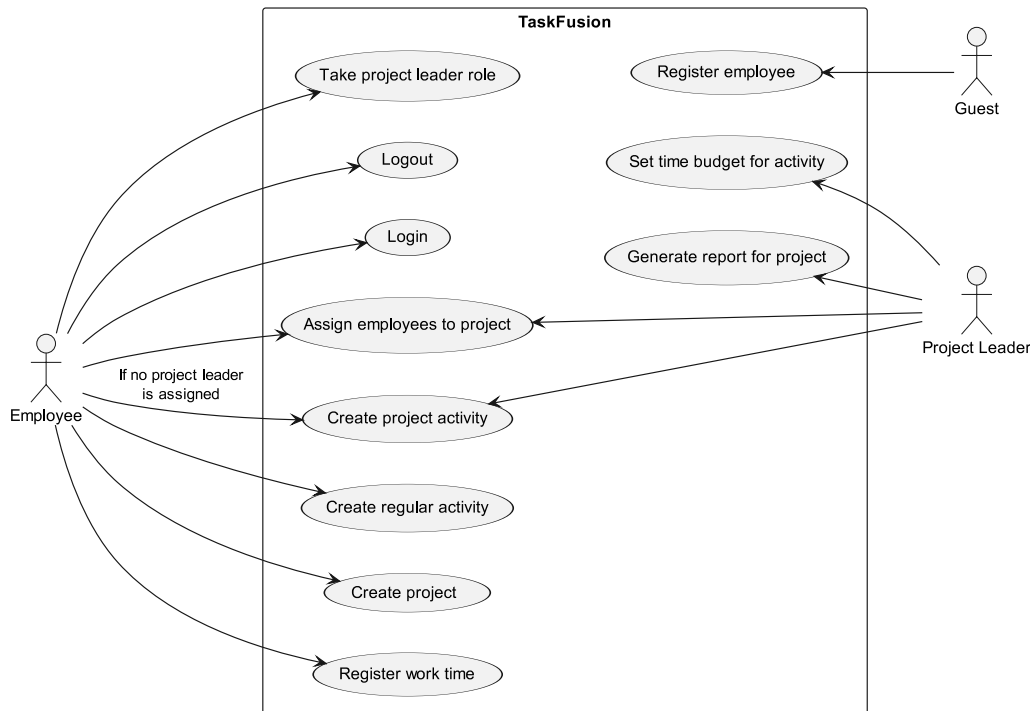
1.2. ORDLISTE

Medarbejder	<i>[Employee]</i> En medarbejder er en entitet ansat i Softwarehuset A/S, som har et unikt medarbejder ID (medarbejder initialer). En medarbejder kan påtage sig en projektleder rolle, oprette projekter, faste aktiviteter, projektaktiviteter og registrere arbejdstid på aktiviteter.
Projekt aktivitet	<i>[Project activity]</i> En delopgave af et projekt. Aktiviteter har en start- og slutuge, og kan få fastsat en budgetteret tid. Alle medarbejdere kan udføre arbejde på alle projektaktiviteter.
Fast aktivitet	<i>[Regular activity]</i> Aktivitet der ikke kan pålægges et projekt. Feks. ferie, sygdom, kurser. Disse har en start- og slutuge.
Projekt	<i>[Project]</i> Udviklingsarbejde udført for en kunde. Et projekt administreres af en projektleder og er inddelt i aktiviteter. Hvert projekt har et projektnummer.
Kunde	<i>[Customer]</i> En ekstern entitet som bestiller og er modtager af projekter.
Projektleder	<i>[Project leader]</i> En medarbejder der har ret til at oprette og tildele aktiviteter for et givent projekt samt generere projektrapporter.
Medarbejder initialer	<i>[Employee initials]</i> Unik identifikation for hver enkelt medarbejder, bestående af fire bogstaver. To første fra fornavn efterfulgt af to første fra efternavn. Feks. "rawi". Hvis initialer allerede er taget, vælges bogstav et og tre i efternavn, derefter et og fire, osv.
Projektnummer	<i>[Project number]</i> Identifikation for hvert enkelt projekt. Har formen årstal efterfulgt af et trecifret løbenummer. Feks. "23001"
Budgetteret tid	<i>[Time budget]</i> En aktivitets estimerede antal hele timer.
Arbejdstidsregistrering	<i>[Work time registration]</i> Mængde tid i inkrementer af halve timer, brugt på en aktivitet. Kan registreres af den medarbejder som har brugt arbejdstid på en given aktivitet.
Start- og sluttid	<i>[Start- and end week]</i> En periode med opløsning på uge-niveau til aktiviteter. Begge tider angives som år og uge, ved formatet "ÅÅUU". Feks. 2304 for uge 4 i 2023. En starttid afgrænser starten af en given uge, en sluttid afgrænser ved slutningen af en given uge. Start- og sluttid kan derfor godt være ens.
Projektrapport	<i>[Project report]</i> Et udskrift der viser info om et projekt, (Leder, kunde, projekt ID), samt aktiviteter, deres arbejdstidsregistreringer og samlede fremgang.

1.3. USE CASE DIAGRAMMER

Dette afsnit starter med Figur 1 som viser hvilke aktører der kan udfører hvilke cases. Herefter kommer en oversigt (Tabel 1) over de detaljerede use cases med henvisning til Cucumber features (Sider 3 til 8) og sekvensdiagrammer (Sider 10 til 10).

Figur 1: Use case diagram for programmet hvori de tre aktører inkluderet er Gæst, Medarbejder og Projektleder.



Som beskrevet i Figur 1 har vi tre roller: Guest, medarbejder og projektleder. Gæsten er en indgang til et system uden data og kan kun oprette medarbejdere i systemet. Medarbejdere har rettigheder til stort set alle funktioner i programmet, indtil en projektleder udpeges. Herefter er det kun projektlederen på et projekt, der kan tilføje medarbejdere til projektet, oprette projektaktiviteter, angive tidsbudgetter og generere rapporter.

Tabel 1: Use cases for programmet

Detaljeret use case	Cucumber feature	Sekvensdiagram
Opret medarbejder	Listing 1	Figur 2
Login	Listing 2	Figur 3
Logout	Listing 3	Figur 4
Opret projekt	Listings 4 og 5	Figur 5
Påtag projektlederrolle	Listing 6	Figur 6
Tildel medarbejder til projekt	Listing 7	Figur 7
Opret projektaktivitet	Listing 8	Figur 8
Opret fast aktivitet	Listing 9	Figur 9
Registrer arbejdstid	Listing 10	Figur 10
Anfør tidsbudget på projektaktivitet	Listing 11	Figur 11
Generér projektrapport	Listing 12	Figur 12

1.4. DETALJEREDE USE CASES

Listing 1: Cucumber feature: Opret medarbejder

```
1 Feature: Register employee
2 Description: A new employee is added to the application
3 Actors: Guest
4
5 #MAIN SCENARIOS
6 Scenario: 1. Register an employee
7     When the user registers an employee with first name "Michael", last name "Laudrup"
8     Then an employee with first
9         ↳ name "Michael", last name "Laudrup" and initials "mila" exists in the application
10
11 #ALTERNATIVE SCENARIOS
12 Scenario: 1a. First name is required to register an employee
13     When the user registers an employee with first name "", last name "Laudrup"
14     Then the error message "Fornavn mangler" is given
15
16 Scenario: 1b. Last name is required to register an employee
17     When the user registers an employee with first name "Michael", last name ""
18     Then the error message "Efternavn mangler" is given
19
20 Scenario: 1c. If initials exists for a new employee, next letter in last name is used
21     When the user registers an employee with first name "Michael", last name "Laudrup"
22     And the user registers an employee with first name "Mikado", last name "Laudrup"
23     Then an employee with first
24         ↳ name "Michael", last name "Laudrup" and initials "mila" exists in the application
25     And an employee with
26         ↳ first name "Mikado", last name "Laudrup" and initials "milu" exists in the application
```

Listing 2: Cucumber feature: Login

```
1 Feature: Employees can login
2 Description: An employee logs in to the application
3 Actors: employee
4
5 #MAIN SCENARIOS
6 Scenario: 1. Login using initials
7     Given the user registers an employee with first name "Michael", last name "Laudrup"
8     When the user logs in using initials "mila"
9     Then the user with initials "mila" is logged in as an employee
10
11 #ALTERNATIVE SCENARIOS
12 Scenario: 1a. Employee does not exist
13     Given the user logs in using initials "mila"
14     Then the error message "Ukendt medarbejder" is given
15
16 Scenario: 1b. Login is case insensitive
17     Given the user registers an employee with first name "Michael", last name "Laudrup"
18     When the user logs in using initials "MiLa"
19     Then the user with initials "mila" is logged in as an employee
```

Listing 3: Cucumber feature: Logout

```
1 Feature: Employees can log out
2 Description: An employee logs out of the application
3 Actors: employee
4
5 #MAIN SCENARIOS
6 Scenario: 1. Logout
7     Given the user registers an employee with first name "Michael", last name "Laudrup"
8     And the user logs in using initials "mila"
9     When the user logs out
10    Then none is logged in
```

Listing 4: Cucumber feature: Opret projekt (fortsætter på Listing 5)

```
1 Feature: Creating a project
2 Description: An employee creates a project in the application
3 Actors: employee
4
5 #BACKGROUND
6 Background:
7     Given the user registers an employee with first name "Michael", last name "Laudrup"
8     And the user logs in using initials "mila"
9
10 #MAIN SCENARIOS
11 Scenario: 1. Creating a project
12     Given the year is 2019
13     When the user creates a project with title "Projektplanlægning"
14     Then a project
15     ↪ with title "Projektplanlægning" with project number "19001" exists in the application
16     And the employee "mila" have 1 projects
17
18 Scenario: 2. A project can have a customer
19     Given the year is 2023
20     And the user creates a project with title "Projektplanlægning"
21     When the user sets customer "El-Giganten" on project "23001"
22     Then the project "23001" has customer "El-Giganten"
23     And the employee "mila" have 1 projects
24
25 Scenario: 3. A project in an internal project, if it does not have a customer
26     Given the year is 2023
27     And the user creates a project with title "Projektplanlægning"
28     Then the project "23001" is an internal project
29     And the employee "mila" have 1 projects
30
31 Scenario: 4. Project numbers increments with each new project for the same year
32     Given the year is 2023
33     And the user creates a project with title "Projektplanlægning"
34     When the user creates a project with title "Half-life 3"
35     Then a
36     ↪ project with title "Half-life 3" with project number "23002" exists in the application
37     And the employee "mila" have 2 projects
```

Listing 5: Cucumber feature: Opret projekt (fortsat fra Listing 4)

```
37 #ALTERNATIVE SCENARIOS
38 Scenario: 1a. A guest is not able to create a project
39     Given the user logs out
40     And the year is 2023
41     When the user creates a project with title "Projektplanlægning"
42     Then the error message "Login krævet" is given
43
44 Scenario: 1b. A title is required to create a project
45     When the user creates a project with title ""
46     Then the error message "En projekttitel mangler" is given
47     And the employee "mila" have 0 projects
48
49 Scenario: 3a. A project in an external project, if it has a customer
50     Given the year is 2023
51     And the user creates a project with title "Projektplanlægning"
52     When the user sets customer "El-Giganten" on project "23001"
53     Then the project "23001" is an external project
54     And the employee "mila" have 1 projects
55
56 Scenario: 5a. Project number increments is reset with each year
57     Given the year is 2021
58     And the user creates a project with title "Projektplanlægning"
59     And the user creates a project with title "Programdesign"
60     And the year is 2022
61     And the user creates a project with title "Half-life 3"
62     And the user creates a project with title "Implementering"
63
64     Then a project
65     ↪ with title "Projektplanlægning" with project number "21001" exists in the application
66     And a project
67     ↪ with title "Programdesign" with project number "21002" exists in the application
68     And a
69     ↪ project with title "Half-life 3" with project number "22001" exists in the application
70     And a project
71     ↪ with title "Implementering" with project number "22002" exists in the application
72     And there is 4 projects in the application
73     And the employee "mila" have 4 projects
```

Listing 6: Cucumber feature: Påtag projektlederrolle

```
1 Feature: Take on the role of project leader
2 Description: An employee can,
3   ↳ when no project leader is assigned to a project, appoint themselves as project leader.
4 Actors: Employees
5
6 #BACKGROUND
7 Background:
8   Given the user registers an employee with first name "Michael", last name "Laudrup"
9   And the user registers an employee with first name "Brian", last name "Laudrup"
10  And the user logs in using initials "mila"
11  And the year is 2023
12  And the user creates a project with title "Video game"
13  And the user logs out
14
15 #MAIN SCENARIOS
16 Scenario: 1. An employee can appoint themselves as project leader
17   Given the user logs in using initials "mila"
18   When the user takes the role as project leader on project "23001"
19   Then "mila" is the project leader on project "23001"
20
21 #ALTERNATIVE SCENARIOS
22 Scenario: 1a.
23   ↳ An employee appoints himself as project leader on a project where there is already one.
24   Given the user logs in using initials "brla"
25   Given the user takes the role as project leader on project "23001"
26   And the user logs in using initials "mila"
27   Given the user takes the role as project leader on project "23001"
28   Then the error message "Der kan kun være en projektleder" is given
29
30 Scenario: 1b. An employee appoints himself as project leader on a project that does not exist
31   Given the user logs in using initials "mila"
32   And the user takes the role as project leader on project "25001"
33   Then the error message "Projektet kunne ikke findes i samlingen af projekter" is given
```

Listing 7: Cucmber feature: Tildel medarbejder til projekt

```
1 Feature: Assign employees onto project
2 Description: Project leader assigns employees for a project
3 Actors: Project leader
4
5 #BACKGROUND
6 Background:
7     Given the user registers an employee with first name "Michael", last name "Laudrup"
8     And the year is 2023
9     And the user logs in using initials "mila"
10    And the user creates a project with title "Video game"
11    And the user logs out
12
13 #MAIN SCENARIO
14
15 Scenario: 1. Employee assigns employee to project
16     Given the user logs in using initials "mila"
17     And the user registers an employee with first name "Brian", last name "Laudrup"
18     And the user assigns "brla" to the project "23001"
19     Then the employee "brla" is assigned to the project "23001"
20     And the employee "brla" have 1 projects
21
22 Scenario: 2. Project manager assigns an employee
23     Given the user logs in using initials "mila"
24     And the user takes the role as project leader on project "23001"
25     And the user registers an employee with first name "Brian", last name "Laudrup"
26     When the user assigns "brla" to the project "23001"
27     Then the employee "brla" is assigned to the project "23001"
28     And the employee "brla" have 1 projects
29
30 #ALTERNATIVE SCENARIOS
31
32 Scenario: 1a. Employee assigns employee to project, with other employee as project leader
33     Given the user logs in using initials "mila"
34     And the user takes the role as project leader on project "23001"
35     And the user registers an employee with first name "Brian", last name "Laudrup"
36     And the user registers an employee with first name "Pape", last name "Poulsen"
37     And the user logs out
38     When the user logs in using initials "brla"
39     And the user assigns "papo" to the project "23001"
40     Then the error message "Kun projektleder kan tildele medarbejdere til projektet" is given
41     And the employee "papo" have 0 projects
42
43 Scenario: 1b. Employee doesn't exist
44     Given the user logs in using initials "mila"
45     And the user assigns "brla" to the project "23001"
46     Then the error message "Ukendt medarbejder" is given
```

Listing 8: Cucmber feature: Opret projektaktivitet

Listing 9: Cucmber feature: Opret fast aktivitet

Listing 10: Cucmber feature: Registrer arbejdstid

Listing 11: Cucmber feature: Anfør tidsbudget på projektaktivitet

Listing 12: Cucmber feature: Generér projektrapport

2 DISKUSSION: KRAVSPECIFIKATION

2.1. OPLÆGSMÆSSIGE OVERVEJELSER OG AFGRÆNSNINGER

2.1.1. En admin rolle Oplægget nævner som udgangspunkt ikke et krav om en administrator rolle, men derimod at det er et internt system og at der ikke er behov for et sikkerhedslag. Ofte er det oplagt at have en admin facilitet, og i dette projekt kunne det som eksempel give mening til oprettelse af medarbejdere. Men i forbindelse med en kortlægning af projektet samt afgrænsning af funktioner, ser vi større værdi for kunden i at fokusere på de ønskede funktioner i stedet for at *gætte* på at de vil have en admin. Desuden kan en sådan rolle laves senere, hvis nødvendigt. At en gæst kan oprette en medarbejder er dermed et eksempel på brugen af K.I.S.S. (*Keep It Simple Stupid*) ved at anskue denne funktionalitet som en mock admin-klasse, der eksisterer udelukkende i forbindelse med udviklingen af arbejdstidsregistrering og projekthåndtering, de centrale ønsker.

2.1.2. UI og fokus på business logik Oplægget har ingen krav til bruger interface, men derimod rigeligt med krav til funktionalitet. Vi ønsker at levere noget brugbart til kunden hurtigt, og have K.I.S.S. i fokus for vores beslutninger og afgrænsning. Fra tidlige leverancer kan vi bruge kundens feedback til at videreudvikle på softwareløsningen iterativt, og bedre kunne prioritere udvikling af funktioner løbende. Derfor har vi i første ikke noget grafisk bruger interface, men derimod et program der køres i et CLI (*Command Line Interface*).

2.1.3. Medarbejdere med samme navn For at tilgodese at medarbejdere kan hedde det samme vil medarbejder-objekter blive gemt ud fra deres initialer **som derfor skal være unikke**.

2.2. TEKNISKE OVERVEJELSER

2.2.1. Abstractions og compositions Disse to begreber bliver kun brugt i abstrakt forstand, da man i Java ikke har direkte kontrol over objekter. De kan konstrueres, men eksistere selv når objekter der peger på dem slettes. Java har en garbage collector der jævnligt analyserer pointers i programmet og fjerner objekter der ikke bliver peget på.

2.2.2. UI UI'en er som nævnt tekstbaseret. Dette betyder at vi til at starte med ikke har en *Viewer*-klasse, da alt printes til konsollen. Bliver UI-delen for kompliceret kan en *Viewer*-klasse let implementeres.

3

PROGRAMDESIGN

3.1. KLASSEDIAGRAM AF PROGRAMDESIGN

3.2. SEKVENSDIAGRAMMER

Figur 2: Sekvensdiagram: Opret medarbejder

Figur 3: Sekvensdiagram: Login

Figur 4: Sekvensdiagram: Logout

Figur 5: Sekvensdiagram: Opret projekt

Figur 6: Sekvensdiagram: Påtag projektlederrolle

Figur 7: Sekvensdiagram: Tildel medarbejder til projekt

Figur 8: Sekvensdiagram: Opret projektaktivitet

Figur 9: Sekvensdiagram: Opret fast aktivitet

Figur 10: Sekvensdiagram: Registrer arbejdstid

Figur 11: Sekvensdiagram: Anfør tidsbudget på projektaktivitet

Figur 12: Sekvensdiagram: Generér projektrapport

4 DISKUSSION: PROGRAMDESIGN

I dette afsnit bearbejdes to ting kort:

1. Valg af datastrukturer
2. Valg af klassestrukturer

4.1. DATASTRUKTURER

I valg af datastrukturer er det vigtigt hvorledes vi henter og gemmer data. I programmet bliver medarbejdere og aktiviteter defineret med en unik streng, mens projekter bliver defineret med et løbenummer. Hvis man for nemheds skyld konverterer løbenummeret til en streng, er der mulighed for at alle tre objekter kan gemmes i Map strukturer. Dette gør det nemt at hente objekter med `.get(key)`, udføre operationer på objekterne og overskrive objekterne i Map'et med `.put(key, Object)`. Er det nødvendigt at iterere over et Map, kan man også nemt bruge Java's `.stream()` metode. Ønsker man at gemme brugt arbejdstid på en aktivitet er det derimod nemmest at gemme denne i en List, da arbejdstiden kun akkumuleres.

4.2. KLASSESTRUKTURER

Programmet skal holdes simpelt og objekter skal nødvendigvis eje hinanden på en simpel måde. Det er besluttet at en *Controller/Model*-klasse varetager programmets busniessstruktur. Hvis klassen bliver for kompliceret kan der senere indsættes en *Viewer/Controller*-klasse som udelukkende varetager UI. Controller klassen indeholder Maps med projekter, medarbejdere og rapporter. Aktiviteter eksistere som en abstrakt klasse der nedarves til en fast aktivitet (f.eks. ferie), og projekt aktiviteter. De faster kan så ejes af et medarbejder objekt. Projektaktiviteter ejes af projekt objekter. Dermed bliver aktivitetsobjekter så ens som muligt, men ejes af de objekter der skal bruge dem. Med valget af MVC (Model-View-Controller)-arkitektur er der fare for at man som udvikler kan gøre controller-objekter til "gudeklasser", dog egner denne abstraktion sig særligt godt til planlægningsarbejdet, da man ved opmærksomhed på denne fare tvinges til at betragte cohesion og coupling endnu nærmere, end man måske ville have gjort uden den. Ligeledes deler den data og handlinger på de data naturligt (og ansvar), hvilket er en central idé i objektorienteret programmering. Endeligt er det et arkitekturmønster der naturligt passer på denne opgave jf. Softwarehuset's requirements; modellen er i vores tilfælde objekter der repræsenterer projektdata, viewet er vores brugergrænseflade og lader bl.a. Softwarehuset skabe overblik gennem rapporter, og kan vise hvilke medarbejdere er ledige til et projekt. Controlleren udbyder muligheden for at ændre objekterne (f.eks. til registrere arbejdstid, registrere ferie/sygdom og modifikation af projekter).

FIGURER

1	Use case diagram for programmet hvori de tre aktører inkluderet er Gæst, Medarbejder og Projektleder	2
2	Sekvensdiagram: Opret medarbejder	10
3	Sekvensdiagram: Login	10
4	Sekvensdiagram: Logout	10
5	Sekvensdiagram: Opret projekt	10
6	Sekvensdiagram: Påtag projektlederrolle	10
7	Sekvensdiagram: Tildel medarbejder til projekt	10
8	Sekvensdiagram: Opret projektaktivitet	10
9	Sekvensdiagram: Opret fast aktivitet	10
10	Sekvensdiagram: Registrer arbejdstid	10
11	Sekvensdiagram: Anfør tidsbudget på projektaktivitet	10
12	Sekvensdiagram: Generér projektrapport	10

TABELLER

1	Use cases for programmet	2
---	--------------------------	---

LISTINGS

1	Cucumber feature: Opret medarbejder	3
2	Cucumber feature: Login	3
3	Cucumber feature: Logout	4
4	Cucumber feature: Opret projekt (fortsætter på Listing 5)	4
5	Cucumber feature: Opret projekt (fortsat fra Listing 4)	5
6	Cucumber feature: Påtag projektlederrolle	6
7	Cucumber feature: Tildel medarbejder til projekt	7
8	Cucumber feature: Opret projektaktivitet	7
9	Cucumber feature: Opret fast aktivitet	7
10	Cucumber feature: Registrer arbejdstid	7
11	Cucumber feature: Anfør tidsbudget på projektaktivitet	8
12	Cucumber feature: Generér projektrapport	8