

02161 Software Engineering 1 Spring 21
Rapport 1

Gruppe 27:

Anders Reher Christensen (s194587)

Anders Gad Lauridsen (s204496)

Magnus Kragh Siegumfeldt (s204472)

Victor Rasmussen (s204475)

Marts 2021

DTU



Indhold

1	Kravspecifikation	1
1.1	Indledning	1
1.2	Væsentlige begreber	1
1.3	Use case diagram	3
1.4	Detaljeret use cases	4
1.5	Diskussion af kravspecifikationer og opgaveformulering	7
2	Programdesign	9
2.1	Klassediagram	9
2.2	Sekvensdiagrammer	10
2.3	Diskussion af program-design	11

OBS. I højre margin vil der stå navn samt studienummer på vedkommende, der har skrevet følgende afsnit. Med *følgende afsnit* menes fra og med hvor marginnoten er placeret til den næste marginnote. Navnene vil se være angivet således: Magnus Siegumfeldt (s204472), Anders Lauridsen (s204496), Anders Christensen (s194587) og Victor Rasmussen (s204475). Bemærk at i figurer angives forfatteren i figurteksten.

1 Kravspecifikation

1.1 Indledning

Denne rapport er baseret på og dokumenterer de første faser i udviklingen af et program til Softwarehuset A/S. Vi vil i denne rapport se nærmere på problemformuleringen og diskutere dens uklarheder og de afgrænsninger, vi har sat for projektet. Yderligere forklares begreber og vendinger, som vil benyttes gennemgående i både rapporten og det program, der skal udvikles.

Vi ser efterfølgende nærmere på udkastet til programmets design og opbygning. Programmets design er specificeret og struktureret ved brug af en række værktøjer - nogle mere specifikke end andre. I første omgang ser vi nærmere på *Use Case*-diagrammer og opskriver på baggrund af disse en række detaljerede Use Cases som skal hjælpe den senere udvikling af programmet. Vi vil altså i stor grad benytte os af *Test-Driven-Development* under udviklingen af dette projekt.

Senere i denne rapport ser vi nærmere på programmets struktur og dets generelle design. Designet er modelleret ved brug af et klassediagram, der beskriver den overordnede struktur samt en række sekvensdiagrammer, som giver et indblik i, hvordan programmet skal udføre forskellige scenarier, samt hvilke klasser der arbejder sammen.

1.2 Væsentlige begreber

Project Management App / The system (Da.: *Project Management App / Systemet / Programmet*)

Project Management App, hvilken også kaldes systemet, betegner applikationen, som skal udvikles. Dette dækker over alt program logik og grænsefladen hertil. Systemet skal muliggøre oversigt og administration af forskellige projekter, som alle har forskellige tilknyttede opgaver. Yderligere skal systemet tilbyde administration af brugere med forskelligt ansvar. Systemet skal også tilbyde oversigt, registrering og administration af timeforbrug for de enkelte medarbejdere.

User interface (Da.: *Brugergrænseflade*)

Brugergrænsefladen betegner den interface, som muliggør interaktion med systemet. Denne er ikke nødvendigvis en grafisk brugerflade, og kan ske igennem en konsol.

Employee (Da.: *Medarbejder*)

En medarbejder betegner alle ansatte i firmaet, som skal have en bruger i systemet. En medarbejder kan tildeles opgaver af en projektleder.

Project Manager (Da.: *Projektleder*)

Projektleder betegner en udvalgt medarbejder, som er har ansvaret for et givet projekt. Det er projektlederens ansvar at oprette opgaver, uddelegeres opgaver og han/hun har mulighed for at se information om det projekt, han/hun er projektleder for.

Project (Da.: *Projekt*)

Et projekt betegner udviklingen af et stykke software - enten for en mulig kunde eller til intern brug. Et projekt har en række underopgave, som kan tildeles forskellige medarbejdere. Et projekt har typisk en tilknyttet projektleder. Et projekt har altid et genereret projektnummer og et angivet navn.

Project id (Da.: *Projektnummer*)

Et projektnummer er et unikt nummer, der automatisk tildeles et projekt, når det oprettes i systemet. Projektnummeret består af et årstal og et løbenummer. Hvis et projekt oprettes i 2021 og har løbenummer 0001, så bliver projektnummeret 210001.

Activity (Da.: *Aktivitet*)

En aktivitet er enhver form for beskæftigelse, der bruger arbejdstimer. Dette indeholder opgaver, ferie,

Følgende afsnit af:
Magnus
Siegmundfeldt
(s204472)
Victor Rasmussen
(s204475)
Anders Lauridsen
(s204496)
Anders Christensen
(s194587)

Følgende afsnit af:
Magnus
Siegmundfeldt
(s204472)
Victor Rasmussen
(s204475)
Anders Lauridsen
(s204496)
Anders Christensen
(s194587)

kurser og sygdom. Hver aktivitet har et starttidspunkt og et forventet sluttidspunkt, så det vides, hvornår medarbejder forventes at være ledig. Sluttidspunktet, kan ændres løbende.

Tasks (*Da.: Opgave*)

En opgave betegner et stykke arbejde, der kan udføres af en eller flere medarbejdere. En opgave er en underdel af et projekt - projektet består derfor af én eller flere opgaver, som skal udføres.

Estimated time (*Da.: Estimeret tidsforbrug*)

Det estimerede tidsforbrug betegner den tid, som en given opgave forventes at kræve. Et projekts estimerede tidsforbrug er givet ved summen af dets opgavers tidsforbrug.

Time spent (*Da.: Tid brugt*)

Tid brugt betegner den mængde af tid, der allerede er brugt på en given opgave. Et projekts tidsforbrug betegner summen af tid brugt på dets opgaver.

Remaining time (*Da.: Restarbejde*)

Restarbejde betegner den tid, der mangler, før opgaven forventes færdig. Et projekts restarbejde betegner tilsvarende den tid, der mangler, før et projekt forventes færdigt.

Reports (*Da.: Rapport*)

En rapport kan genereres af en projektleder og indeholder en oversigt over et projekts udvikling, altså oversigt over udførte og manglende opgaver, samt tid brugt på projektet og manglende arbejdstimer, før projektets opgaver er færdiggjort.

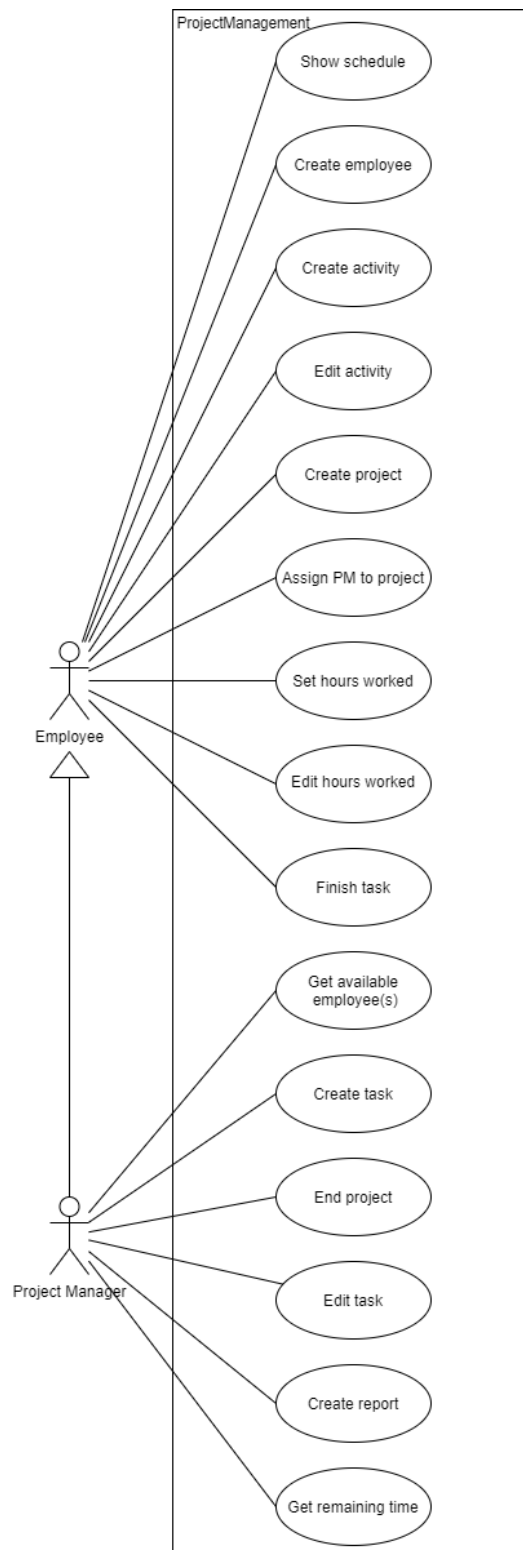
Time registration (*Da.: Timeregistrering*)

Medarbejdere skal kunne registrere de timer, de har arbejdet på forskellige opgaver, samt oprette ferie, kurser og lign. Denne funktion kaldes timeregistrering.

Available Employee (*Da.: Ledig medarbejder*)

En medarbejder er ledig, når han/hun ikke er tilknyttet en aktivitet på et givet tidspunkt.

1.3 Use case diagram



Figur 1: Use Cases

Følgende afsnit af:
Magnus
Siegumfeldt
(s204472)
Victor Rasmussen
(s204475)
Anders Lauridsen
(s204496)
Anders Christensen
(s194587)

1.4 Detaljeret use cases

Feature: Create project

Description: An employee is able create a project. The project is also added to the project list
Actors: Employee

Scenario: Create project successfully

Given there is an employee
When the active user creates a project with a name
Then the project is created
And the project is added to the list of projects

Scenario: An employee tries to create a project with no name

Given there is an employee
When the active user tries to create a project which does not have a name
Then the error message "A project needs a name" is given

Følgende afsnit af:
Anders Christensen
(s194587)

Feature: Create task

Description: A project manager is able create a task. The task is also added to the list of tasks.
Actors: Project manager

Background: The app has a project "project1" and two employees. One of the employees is project manager of "project1" and the other employee is not

Given these employees are contained in the app
| John |
| Hannibal |
And there is a project
| project1 |
And "John" is project manager of "project1"

Scenario: A project manager creates a task successfully

Given there is a project with the name "project1"
And an employee named "John" is project manager of "project1"
When "John" creates a task with the name "Refactoring"
Then the task "Refactoring" is created
And the task is added to the list of tasks

Scenario: An employee tries to create a task

Given there is a project with the name "project1"
And there is an employee with the name "Hannibal", who is not the project manager of "project1"
When "Hannibal" tries to create a task
Then the error message "You have to be a project manager to create a task" is given

Følgende afsnit af:
Anders Lauridsen
(s204496)

Feature: Edit task

Description: The project manager edits a task on a project
Actors: Project manager

Background: The app has a project "project1" and two employees. One of the employees is project manager of "project1" and the other employee is not

Given these employees are contained in the app
| John |
| Hannibal |
And there is a project
| project1 |
And "project1" has a task
| export java file |
And "John" is project manager of "project1"

Scenario: Edits a task successfully

Given there is an employee

Følgende afsnit af:
Anders Christensen
(s194587)

```
And "John" is the project manager of "project1"
And "John" is the active user
And there is a task with the name "export java file" in "project1"
When "John" edits that task
Then the task is edited succesfully
```

Scenario: Edits a task when active user is not the project manager

```
Given there is an employee
And "John" is the project manager of "project1"
And "Hannibal" is the active user
And there is a task with name "export java file" in "project1"
When "Hannibal" tries to edit that task
Then the error message "Active user is not project manager for this project" is given
```

Følgende afsnit af:
Magnus
Siegmundfeldt
(s204472)

Feature: Assign project manager

Description: The employees assign a project manager to a project
Actors: Employee

Background: The app has a project "project1" and two employees

```
Given these employees are contained in the app
| John |
| Hannibal |
And there is a project
| project1 |
```

Scenario: An employee is assigned as project manager

```
Given there is a project "project1"
And the project doesn't have a project manager
And there is an employee "John"
When the active user assigns "John" as project manager of "project1"
Then "John" is assigned as project manager of "project1"
```

Scenario: A project already has a project manager, and the employee tries to assign another project manager

```
Given there is a project "project1"
And "John" is the project manager of "project1"
And there is an employee "Hannibal"
When the active user assigns "Hannibal" as project manager for "project1"
Then the error message "The project already has a project manager" is given
```

Følgende afsnit af:
Anders Lauridsen
(s204496)

Feature: Create activity

Description: An employee is able create an activity in his/her schedule
Actors: Employee

Background: The app has an employee

```
Given these employees are contained in the app
| Hanne |
```

Scenario: Create activity successfully

```
Given there is an employee "Hanne"
And "Hanne" is the active user
When when the active user creates an activity with the name "holiday" and date "20.04-25.04"
Then the acty is created
And the activity is added to the active user's schedule
```

Scenario: An employee tries to create an activity with no name

```
Given there is an employee, who is the active user
When the active user tries to create an activity and provides no name
Then the error message "An activity needs a name" is given
```

Scenario: An employee tries to create multiple task in the same spot

Given an employee has an activity at "08:00, 20.04.2021"
When the employee tries to create an activity at "08:00, 20.04.2021"
Then the error message "Time slot not available" is given

Følgende afsnit af:
Victor Rasmussen
(s204475)

Feature: Edit activity
Description: An employee is able edit an activity in his/her schedule
Actors: Employee

Scenario: Edit activity successfully
Given there is an employee with the name "Hanne"
And "Hanne" is the active user
And in the active user's schedule, there is an activity
When the active user changes the activity
Then the activity is changed

Scenario: An activity is edited to have no name
Given there is an employee
When the active user tries to edit the name of an activity to ""
Then the error message "An activity needs a name" is given

Følgende afsnit af:
Victor Rasmussen
(s204475)

Feature: Get available employees
Description: An employee gets a list of all employees that are available during a duration
Actors: Employee

Background: The app has a number of employees with assigned tasks and some activities
Given these employees are contained in the app
| John |
| Bob |
| Harry |
And "John" and "Bob" have empty schedules
And "Harry" has a full schedule

Scenario: Get available employees successfully
Given there is an employee
When the active user provides a duration
And there is an employee who is available in the duration
Then the available employees "John" and "Bob" are given

Scenario: There are no available employees
Given there is an employee
And "John" and "Bob" are given full schedules
When the active user provides a duration
Then there are no employees who are available
And the message "No available employees at the given time" is given

Scenario: No duration provided
Given there is an employee
When the active user provides an empty duration
Then the error message "You have not selected a duration to find available employees" is given

Følgende afsnit af:
Magnus
Siegumfeldt
(s204472)

Feature: Get remaining time for a project
Description: An employee who is a project manager gets the remaining time assigned to a project
Actors: Project manager

Background: The app has a project with assigned tasks
Given there is a project
#| name | projectId |


```

    | Minesweeper | 20210001 |
And these tasks are contained in the project
# | name | timeSpent | estimatedTime |
| Refactoring | 6 | 6 |
| UI update | 14 | 12 |
| Tests | 5 | 12 |

Scenario: Get remaining time successfully
  Given there is an employee
  And the active user is the project manager of "Minesweeper"
  And the tasks "Refactoring", "UI update" and "Tests" have cumulatively taken 25 hours so far
  And the estimated project time is more than the time spent on tasks
  When the project manager requests the remaining time
  Then the remaining time "Estimated remaining time: 5 hours" is given

Scenario: Precisely all time is spent on tasks
  Given there is an employee
  And the active user is the project manager of "Minesweeper"
  And the tasks "Refactoring", "UI update" and "Tests" have cumulatively taken 25 hours so far
  And the tasks "Refactoring", "UI update" and "Tests" are edited to an total estimated time of
    25 hours
  When the active requests the remaining time of "Minesweeper"
  Then there is no more remaining time
  And the message "All currently budgeted time has been spent on tasks" is shown

Scenario: More time than estimated has been spent on tasks
  Given there is an employee
  And the active user is the project manager of "Minesweeper"
  And the tasks "Refactoring", "UI update" and "Tests" have cumulatively taken 25 hours so far
  And the tasks "Refactoring", "UI update" and "Tests" are edited to an total estimated time of
    20 hours
  When the project manager requests the remaining time
  Then there is no more remaining time
  And the message "The project has used 5 hours more than estimated" is shown

```

Følgende afsnit af:
 Magnus
 Siegmundfeldt
 (s204472)
 Victor Rasmussen
 (s204475)
 Anders Lauridsen
 (s204496)
 Anders Christensen
 (s194587)

1.5 Diskussion af kravspecifikationer og opgaveformulering

I dette afsnit nævnes og diskuteres kort evt. uklarheder og beslutninger taget i forbindelse med problemformuleringen. Evt. antagelser og overvejelser vil også blive nævnt.

Begreber og specifikationer

I problemformuleringen bruges begrebet 'Aktivitet' bredt og uspecifikt, hvorfor vi i dette projekt har valgt at dele dette begreb i to forskellige begreber, hhv. 'Opgave' og 'Aktivitet', hvilke begge er defineret i Afsnit 1.2. I programmet og rapporten er det essentielt at kunne formulere og udtrykke sig specifikt om disse områder, så vi fandt det nødvendigt at specificere de to begreber.

Yderligere møder vi i opgaveformuleringen en specifikation af et projektnummer som er baseret på de sidste 2 cifre af et årstal. Disse to cifre vil gå igen hvert 100. år, hvilket i teorien kan skabe problemer i systemet. I dette projekt er det ikke en problemstilling, vi har valgt at arbejde videre med og finde en løsning på. Det ville dog være en relevant problemstilling, da det vil give en oplagt udløbsdato på ens software, hvor det ville skulle opdateres.

Uspecifikke og ikke målbare krav

I problemformuleringen nævnes en række ikke målbare krav til programmet, herunder "Det skal være enkelt at foretage denne registrering, så medarbejderen ikke føler det er en byrde", og "Medarbejderen skal enkelt kunne se, om han/hun har registreret alle timer man har arbejdet i dag", hvilke det ikke har været muligt at skrive deciderede Use cases til. Generelt kan dette håndteres ved at opsætte specifikke mål for dette, fx "En medarbejder skal kunne registrere sit timeforbrug på under 1 minut, og der må højst opstå brugerfejl i 1 ud af 50 brugsgange.". Krav som disse er i dette projekt ikke håndteret med samme vægt som de detaljerede Use Cases, da disse ofte kræver en lang række brugertest, hvilke der

ikke vil være samme fokus på i dette projekt.

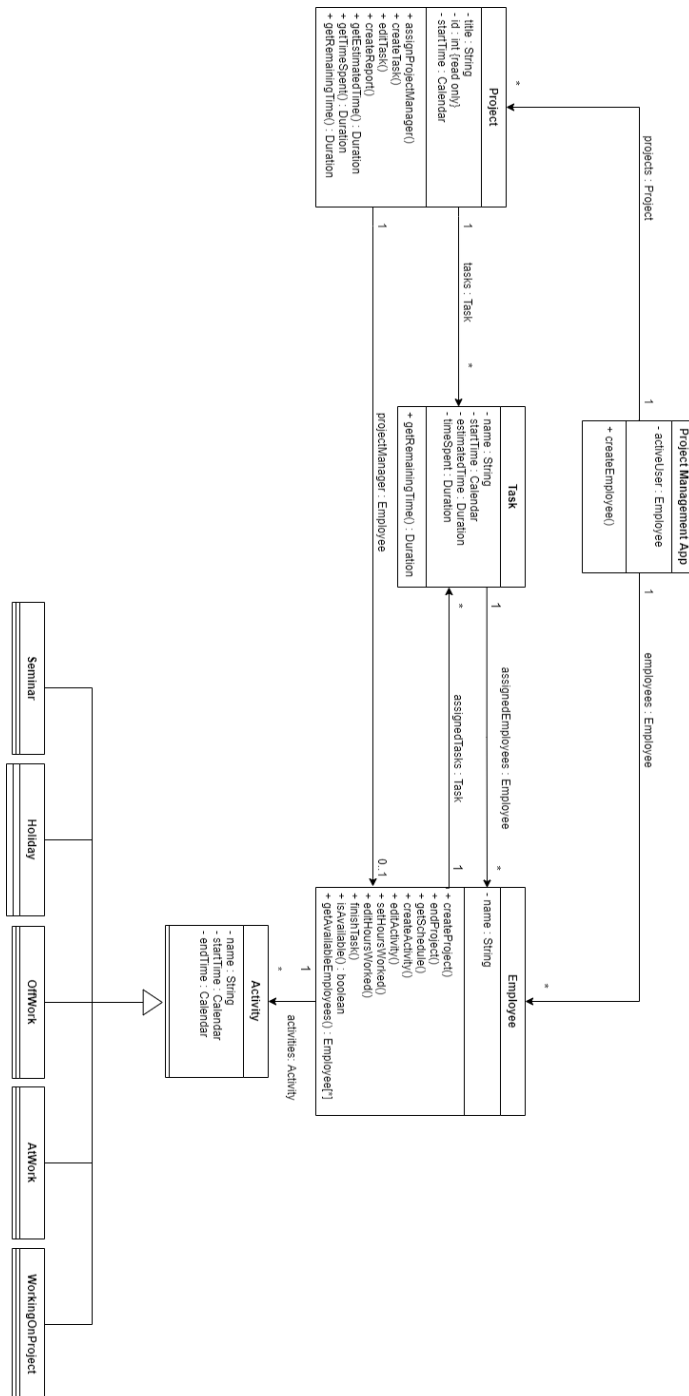
Programmets struktur og sikkerhed

I problemformuleringen støder vi på et stort problem, hvilket vil være en nødvendighed at forholde sig til inden programmet kan udvikles. I systemet er brugerene opdelt i to adgangsniveauer med forskellige muligheder, henholdsvis 'Medarbejdere' og 'Projektledere'. Projektlederne udvælges dog blandt medarbejderne af medarbejderne, og fra et sikkerhedsmæssige synspunkt kan dette skabe store problemer. Yderligere er det specificeret at adgangskontrol ikke er nødvendigt, hvilket på samme måde åbner for en lang række af måder systemet kan misbruges på. Disse sikkerhedsmæssige områder, er dog ikke noget vi har valgt og have fokus på i udviklingen af systemet - dette kunne man evt. have arbejdet videre med, hvis man havde haft et andet fokusområde i projektet.

2 Programdesign

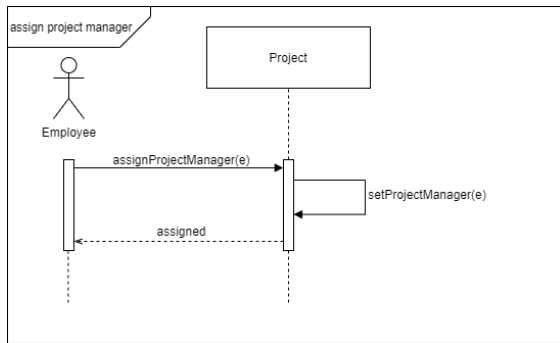
2.1 Klassediagram

Følgende afsnit af:
 Magnus
 Siegmufeldt
 (s204472)
 Victor Rasmussen
 (s204475)
 Anders Lauridsen
 (s204496)
 Anders Christensen
 (s194587)

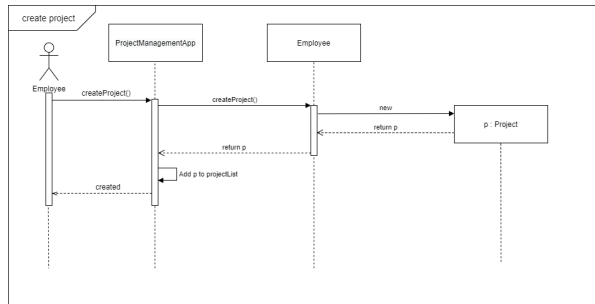


Figur 2: Klassediagram

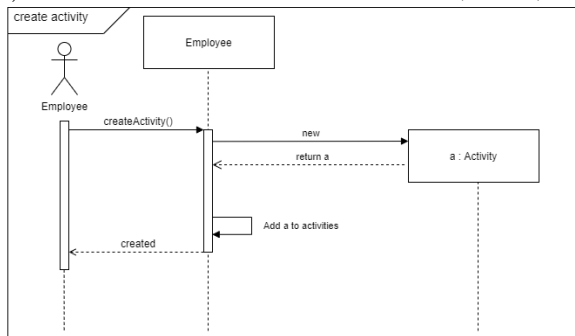
2.2 Sekvensdiagrammer



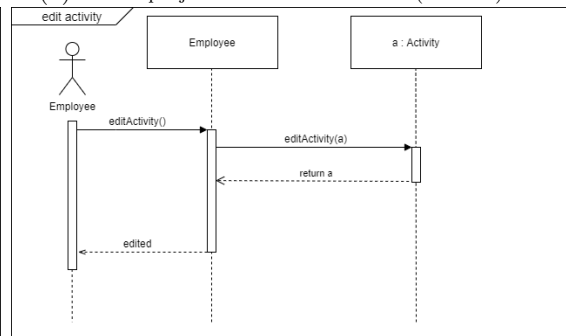
(a) Assign project manager: Magnus Siegumfeldt (s204472)



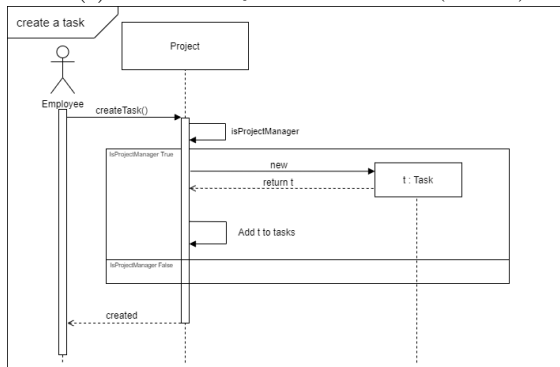
(b) Create project: Anders Christensen (s194587)



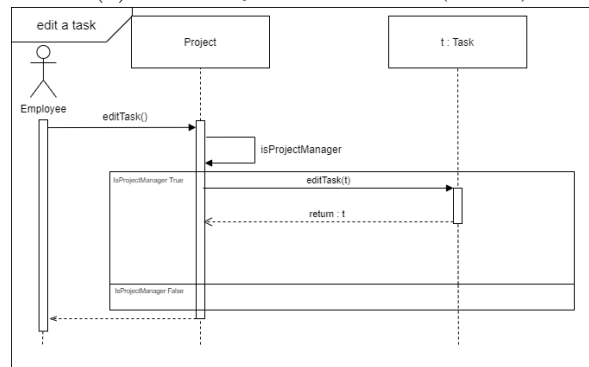
(c) Create activity: Anders Lauridsen (s204496)



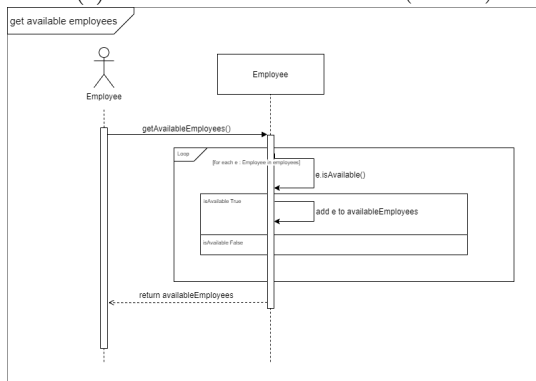
(d) Edit activity: Victor Rasmussen (s204475)



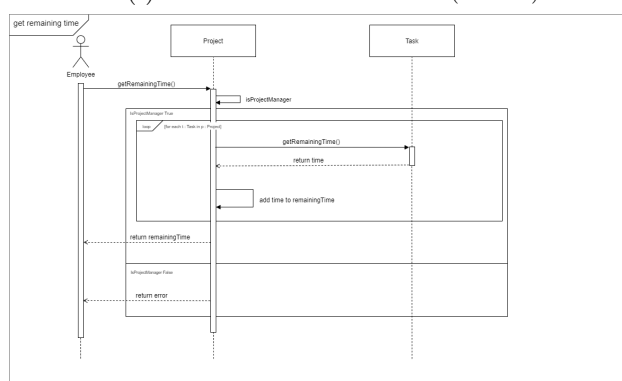
(e) Create task: Anders Lauridsen (s204496)



(f) Edit task: Anders Christensen (s194587)



(g) Get available Employees: Victor Rasmussen (s204475)



(h) Get remaining time: Magnus Siegumfeldt (s204472)

Figur 3: Sekvensdiagrammer

2.3 Diskussion af program-design

I dette afsnit forklares overvejelser og beslutninger foretaget omkring programmets design. Som set på klassediagrammet skal systemet opbygges af en række klasser, hvor 'Project', 'Employee' og 'Task' er de mest betydningsfulde.

Vi havde i gruppen flere overvejelser om, hvordan en projektleder for et givet projekt skulle noteres og opbevares. Vi havde især muligheden for nedrivning åben, men da en medarbejder kan være både medarbejder og projektleder samtidig, var dette ikke en mulighed. Løsningen som vi i gruppen har valgt at benytte er, at ethvert projekt har et felt 'projectManager' af typen 'Employee'. Dette felt beskriver, hvilken medarbejder der har rettigheder til at oprette og tildele opgaver til andre medarbejdere.

Vi havde yderligere en del overvejelser i forhold til, hvordan opgaver og aktiviteters starttidspunkt, sluttidspunkt og estimerede tid skulle repræsenteres. Den valgte løsning i gruppen er ved brug af 'Calendar' objekter, hvilke vi så implementeret i LibraryApp øvelserne. Yderligere bruges Java's 'Duration' objekt til at repræsentere et forventet tidsforbrug eller manglende tid - vi har i gruppen ikke arbejdet med dette objekt før, så der kan opstå evt. ændringer mht., hvordan tid og tidsforbrug implementeres.

Generelt er vi i gruppen opmærksomme på at klassediagram og sekvensdiagrammerne er værktøjer til, at vi i gruppen kan udvikle et godt og struktureret program, men vi er dog også klar over, at der kan ske ændringer i strukturen, når først udviklingen begynder, og vi støder på tekniske problemer.

Følgende afsnit af:
Magnus
Siegumfeldt
(s204472)
Victor Rasmussen
(s204475)
Anders Lauridsen
(s204496)
Anders Christensen
(s194587)