Danmarks Tekniske Universitet

02161 - Software Engineering 1

Report 1 - Group 3

Dejan Joel Künnemeyer
s235470

Mads Amtoft Pedersen
s235472

Bertram Fink-Jakobsen
s235474

Sebastian Tobias Holdt
s235475

Karl Gustav Kofoed Petersen
s235481

AI tools have been lightly used for grammar correction and polish.

March 24, 2025

Contents

1 Requirement specification

1.1 Introduction

We have been tasked with developing a time registration and project management system to replace an outdated solution currently used by 'Softwarehuset A/S'. Our goal is to improve the processes for registering projects, activities, and employee work hours, as well as enabling efficient review of these records for employees.

This report focuses primarily on documenting detailed use cases and presenting a preliminary program design. The report concludes with a brief discussion that explains and supports the choices made in both the use-case definitions and system design.

Ultimately, this report aims to provide a solid foundation for the future implementation of the proposed system by 'Softwarehuset A/S'.

1.2 Glossary

- Project: A structured work initiative with a specific goal, consisting of multiple activities. It can be for an external client or an internal task within Softwarehuset A/S. A project has a name, an id, a time frame, a list of activities and potentially a project leader.
- Activity: A task within a project, such as design, programming, or testing, with an estimated duration and required effort. Employees register time spent on activities. An activity has a name, a time frame, an estimated amount of hours and a list of assigned employees and their worked hours per day.
- Employee: A developer working on multiple projects and activities, identified by a unique set of initials (up to four letters). Employees register their time daily.
- Project leader: A special employee responsible for structuring a given project, assigning employees to activities, estimating work effort, and tracking progress.
- Firm activity: An activity that is not related to a project and can usually be determined further in advance.
- Start- and end date: Describing the time frame of projects and activities. They can be changed by project leaders or employees on projects without project leaders.
- Project ID: A unique identifier assigned to each project, consisting of the year and a sequential number (e.g., 22001 for the first project in 2022).

1.3 Use case-diagram

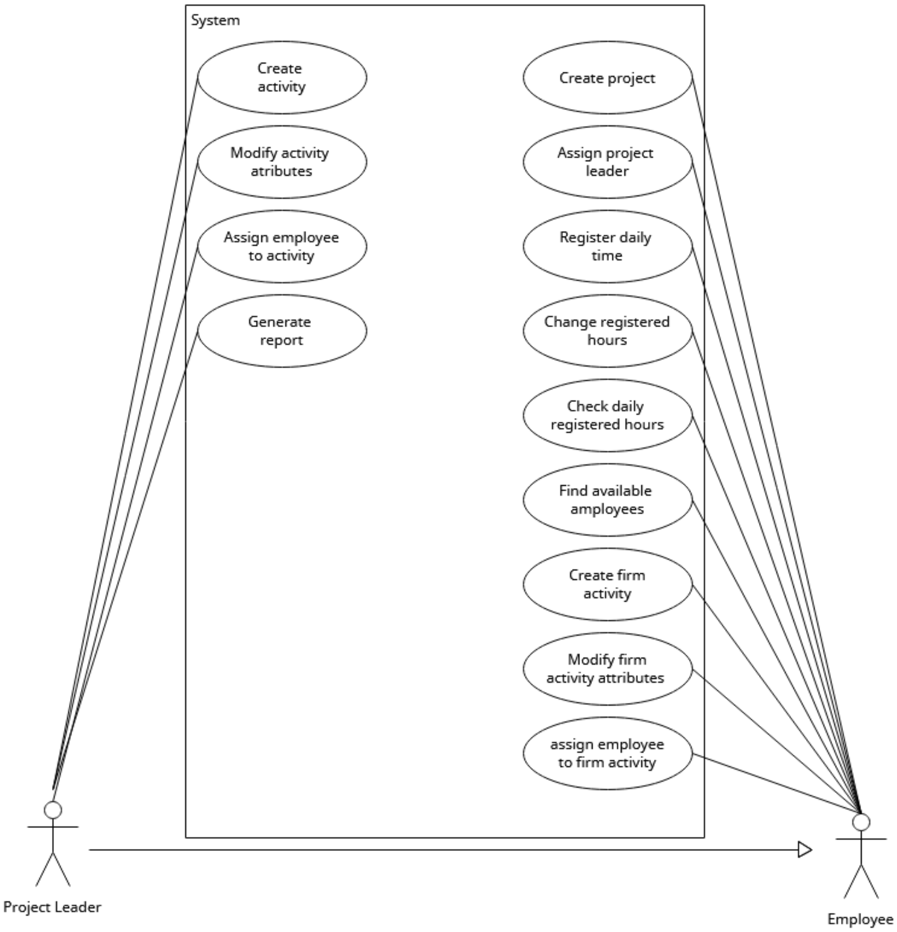

Figure 1: Use case-diagram

1.4 Detailed use cases

1.4.1 Register time daily (Karl)

```
Feature: Register time Daily
  Description: Employee registers time used for the day
  User: Employee

  Scenario: Employee registers 5 hours to existing activity
    Given a project
    And it has an activity
    When the user tries to register daily time to 5 for activity
    Then 5 hours has been registered to the activity

  Scenario: Employee registers 5 hours to non-existing activity
```

```
    Given a project
    When the user tries to register daily time to 5 for activity
    Then error message "No Activity To Registers Hours To" is given
```

1.4.2 Check registered daily (Karl)

```
Feature: Check registered Daily
  Description: Employee checks registered hours of the day
  User: Employee

  Scenario: Employee has registered 5 hours to a activity
    Given a project
    And it has an activity
    And the employee has already registered 5 hours to a activity
    When an employee checks daily registered hours
    Then a map with the name of the activity and 5 hours is returned

  Scenario: Employee has registered 0 hours
    Given a project
    And it has an activity
    When an employee checks daily registered hours
    Then an empty map should be returned

  Scenario: Employee has registered 5 hours to a activity and 4 to another activity
    Given a project
    And it has two activities
    When an employee checks daily registered hours
    Then a map with the name of the first activity and 5 hours is returned
    And a map with the name of the second activity and 4 hours is returned
```

1.4.3 Generate report (Sebastian)

```
Feature: Generate report
  Description: A project leader or employee generates a report of the current activities in a project
  User: Employee

  Scenario: Report generated by project leader
    Given a project
    And the user is the leader of the project
    When the user generates a report
    Then a report is generated

  Scenario: Report generated of project without a project leader
    Given a project
    When the user generates a report
    Then a report is generated

  Scenario: Report generated by different employee than project leader
```

```
    Given a project with project leader "Different Employee"
    When the user generates a report
    Then error message "You are not the project leader of this project" is given
```

1.4.4 Find Available Employees (Sebastian)

```
Feature: Find Available Employees
  Description: Employee gets a list of all employees who are not currently working on activities
  User: Employee

  Background:
    Given these employees registered in the app
      | bert | deja | karl | mads | seba |
    And a project
    And an activity in the project

  Scenario: All employees are found
    When searching for available employees
    Then employees found are
      | bert | deja | karl | mads | seba |

  Scenario: Some employees are found
    Given employee with name "bert" is assigned to activity
    And employee with name "karl" is assigned to activity
    When searching for available employees
    Then employees found are
      | deja | mads | seba |

  Scenario: No employees are found
    Given employee with name "bert" is assigned to activity
    And employee with name "deja" is assigned to activity
    And employee with name "karl" is assigned to activity
    And employee with name "mads" is assigned to activity
    And employee with name "seba" is assigned to activity
    When searching for available employees
    Then no employees are found
```

1.4.5 Create Project (Bertram)

```
Feature: Create Project
  Description: A project leader or employee creates a project
  User: Employee

  Scenario: The creation of a new project succeeded
    When creating a new project named "success"
    And no projects of the same name was found
    Then a project with name "success" is created
  Scenario: The creation of a new project failed
    When creating a new project named "failed"
    And a project named "failed" already exists
    Then error message "Project Name Is Taken" is given
```

1.4.6 Change Project (Bertram)

```
Feature: Change Project
  Description: A project leader or employee makes changes to a project
  User: Employee

  Scenario: Start- and end date changed for a project
    Given a project
    And the user is the leader of the project
    When the start date is set to "1.1.2021"
    And the end date is set to "1.5.2021"
    Then the start date is "1.1.2021"
    And the end date is "1.5.2021"

  Scenario: Start- and end date changed by non-project leader for a project
    Given a project
    And the user is the not the leader of the project
    When the start date is set to "1.1.2021"
    And the end date is set to "1.5.2021"
    Then error message "Not Project Leader" is given

  Scenario: set new name for a project as project leader
    Given a project
    And the user is the leader of the project
    When setting project name to "new"
    Then project name is "new"

  Scenario: set new name for a project as non-project leader
    Given a project
    And the user is not the leader of the project
    When setting project name to "new"
    Then error message "Not Project Leader" is given

  Scenario: set new customer for a project as project leader
    Given a project
    And the user is the leader of the project
    When setting project customer to "good customer"
    Then project name is "good customer"

  Scenario: set new customer for a project as non-project leader
    Given a project
    And the user is not the leader of the project
    When setting project customer to "good customer"
    Then error message "Not Project Leader" is given

  Scenario: set new project leader for a project
    Given a project
    Given "per" exists as employee
    When setting leader as "per"
    Then leader is "per"

  Scenario: set a non existing employee as project leader
    Given a project
    Given "John" does not exist as employee
    When setting leader as "John"
    Then error message "Not A Valid Employee" is given
```

1.4.7 Create Project Activity (Mads)

```
Feature: Create Project Activity
  Description: A project leader or employee creates a project activity
  User: Employee

  Scenario: Create a new activity for a project as project leader
    Given a project
    And the user is the leader of the project
    When creating a new activity "A1"
    Then the activity "A1" is a part of the project

  Scenario: Create a new activity for a project as non-project leader
    Given a project
    And the user is not the leader of the project
    When creating a new activity "A1"
    Then error message "Not Project Leader" is given

  Scenario: Create activity with missing name
    Given a project
    When creating a new activity ""
    Then error message "Invalid Activity Name" is given

  Scenario: Creating activity with duplicate name
    Given a project
    And it has an activity "A1"
    When creating a new activity "A1"
    Then error message "Activity Name Already Taken" is given
```

1.4.8 Change Project Activities (Mads)

```
Feature: Change Project Activities
  Description: A project leader or employee makes changes to a project activity
  User: Employee
  Scenario: Start- and end week set by project leader for an activity
    Given a project
    And it has an activity
    And the user is the leader of the project
    When the start week is set to "1"
    And the end week is set to "12"
    Then the start week is "1"
    And the end week is "12"

  Scenario: Start- and end week set by non-project leader for an activity
    Given a project
    And it has an activity
    And the user is not the leader of the project
    When the start week is set to "1"
    And the end week is set to "12"
    Then error message "Not Project Leader" is given

  Scenario: Set estimated hours for an activity as project leader
    Given a project
    And it has an activity
    And the user is the leader of the project
    When setting the estimated hours of an activity to "100"
    Then the estimated time of the activity should be "100"

  Scenario: Ser estimated hours for an activity without being project leader
    Given a project
    And it has an activity
    And the user is not the leader of the project
    When setting the estimated hours of an activity to "100"
    Then error message "Not Project Leader" is given

  Scenario: Add user to activity
    Given a project
    And it has an activity
    When i add myself to the project
    Then i am a part of the project
```

1.4.9 Assign Employee to Activity (Dejan)

```
Feature: Assign Employee
 Description: Assign an Employee to an Activity
 User: Employee

    Scenario: Succesfully add employee to an activity
        Given Some employee "e"
        And Some existing project "p" with an activity "a"
        And "e" is assigned to 0 activities
        And "e" is not assigned to "a" in "p"
        When "e" is assigned to "a" in "p"
        Then "e" is succesfully assigned to "a" in "p"

    Scenario: The employee is already assigned to 20 activities
        Given Some employee "e"
        And Some existing project "p" with an activity "a"
        And "e" is assigned to 20 activities
        And "e" is not assigned to "a" in "p"
        When "e" is assigned to "a" in "p"
        Then "e" wont be assigned to "a" in "p"
        And error message "Too Many Activities" is given

    Scenario: The employee is already assigned to the given activity
        Given Some employee "e"
        And Some existing project "p" with an activity "a"
        And "e" is assigned to 0 activities
        And "e" is already assigned to "a" in "p"
        When "e" is assigned to "a" in "p"
        Then "e" is succesfully assigned to "a" in "p"
```

1.4.10 Assign Employee to Firm Activity (Dejan)

```
Feature: Assign an employee to a firm activity
 Description: Assign an Employee to a Firm Activity
 User: Employee

    Scenario: Succesfully add employee to a firm activity
        Given Some employee "e"
        And Some existing firm activity "a"
        And "e" is not assigned to "a"
        When "e" is assigned to "a"
        Then "e" is succesfully assigned to "a"

    Scenario: The employee is already assigned to the given firm activity
        Given Some employee "e"
        And Some existing firm activity "a"
        And "e" is already assigned to "a"
        When "e" is assigned to "a"
        Then "e" is succesfully assigned to "a"
```

1.5 Discussion of Requirement specification

There were uncertainties in the project description regarding the difference between how Employees and Project leader are expected to be handled in the system. We have come to decision, that we will let each Project to have a field for the project leader of the type Employee. This lets the employee use employee functions and also access project leader functions in that specific project. Firm activities was another topic of ambiguity, as they are described similar to normal activities, but with other properties. Therefore we decided to handle them with the same class but without having them assigned to a project, and not counting them towards the max of 20 activities for an employee.

2 Program design

2.1 Class Diagram - Solution Domain

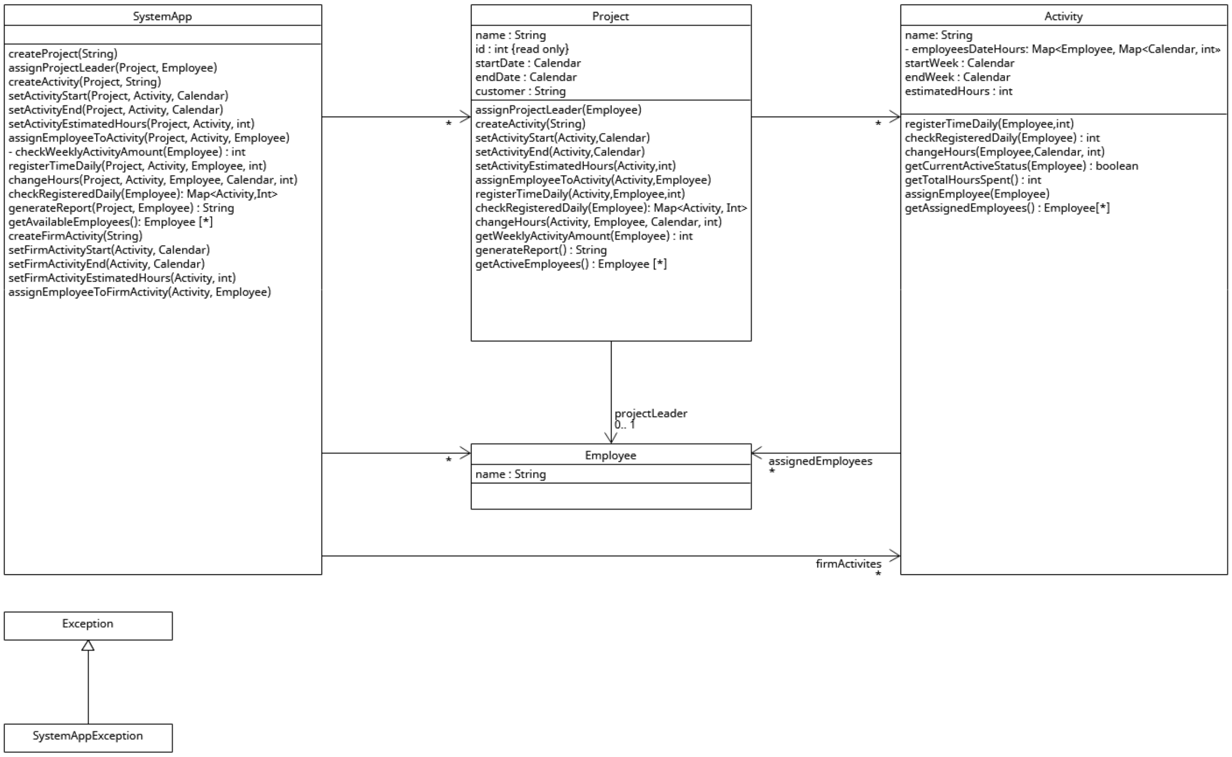
SystemApp
createProject(String)
assignProjectLeader(Project, Employee)
createActivity(Project, String)
setActivityStart(Project, Activity, Calendar)
setActivityEnd(Project, Activity, Calendar)
setActivityEstimatedHours(Project, Activity, int)
assignEmployeeToActivity(Project, Activity, Employee)
- checkWeeklyActivityAmount(Employee) : int
registerTimeDaily(Project, Activity, Employee, int)
changeHours(Project, Activity, Employee, Calendar, int)
checkRegisteredDaily(Employee): Map<Activity,Int>
generateReport(Project, Employee) : String
getAvailableEmployees(): Employee [*]
createFirmActivity(String)
setFirmActivityStart(Activity, Calendar)
setFirmActivityEnd(Activity, Calendar)
setFirmActivityEstimatedHours(Activity, int)
assignEmployeeToFirmActivity(Activity, Employee)
*
Project
name : String
id : int {read only}
startDate : Calendar
endDate : Calendar
customer : String
assignProjectLeader(Employee)
createActivity(String)
setActivityStart(Activity,Calendar)
setActivityEnd(Activity,Calendar)
setActivityEstimatedHours(Activity,int)
assignEmployeeToActivity(Activity,Employee)
registerTimeDaily(Activity,Employee,int)
checkRegisteredDaily(Employee): Map<Activity, Int>
changeHours(Activity, Employee, Calendar, int)
getWeeklyActivityAmount(Employee) : int
generateReport() : String
getActiveEmployees() : Employee [*]
*
Activity
name: String
- employeesDateHours: Map<Employee, Map<Calendar, int»
startWeek : Calendar
endWeek : Calendar
estimatedHours : int
registerTimeDaily(Employee,int)
checkRegisteredDaily(Employee) : int
changeHours(Employee,Calendar, int)
getCurrentActiveStatus(Employee) : boolean
getTotalHoursSpent() : int
assignEmployee(Employee)
getAssignedEmployees() : Employee[*]
projectLeader
0.. 1
*
Employee
name : String
assignedEmployees
*
firmActivites
*
Exception
SystemAppException

Figure 2: Class Diagram - Solution Domain

2.2 Sequence Diagrams

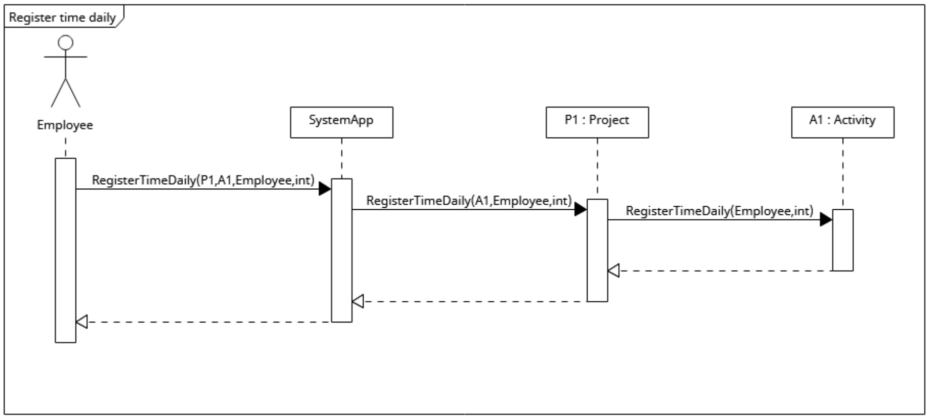
Register time daily
Employee
SystemApp
P1 : Project
A1 : Activity
RegisterTimeDaily(P1,A1,Employee,int)
RegisterTimeDaily(A1,Employee,int)
RegisterTimeDaily(Employee,int)

Figure 3: Sequence Diagram for Register time daily (Karl)

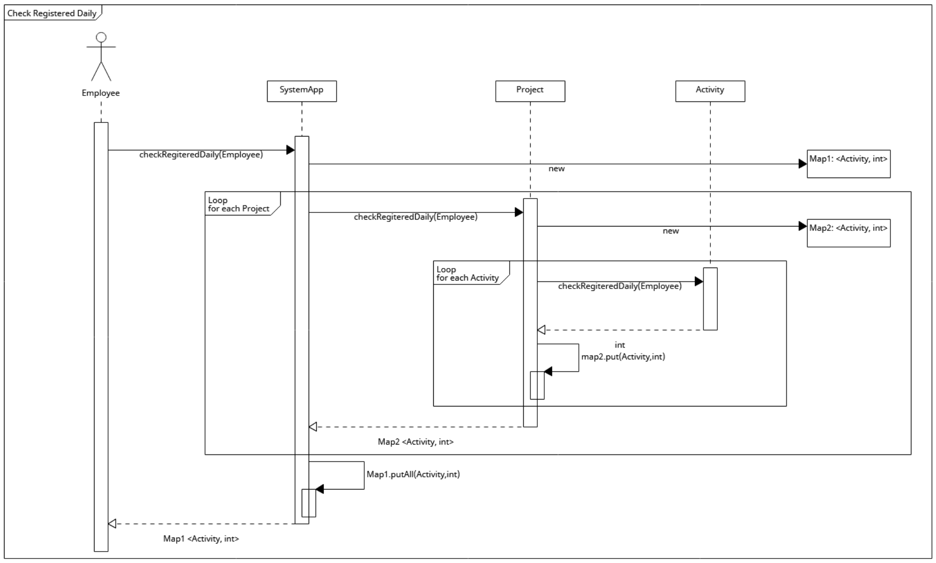

Figure 4: Sequence Diagram for Check Registered Daily (Karl)

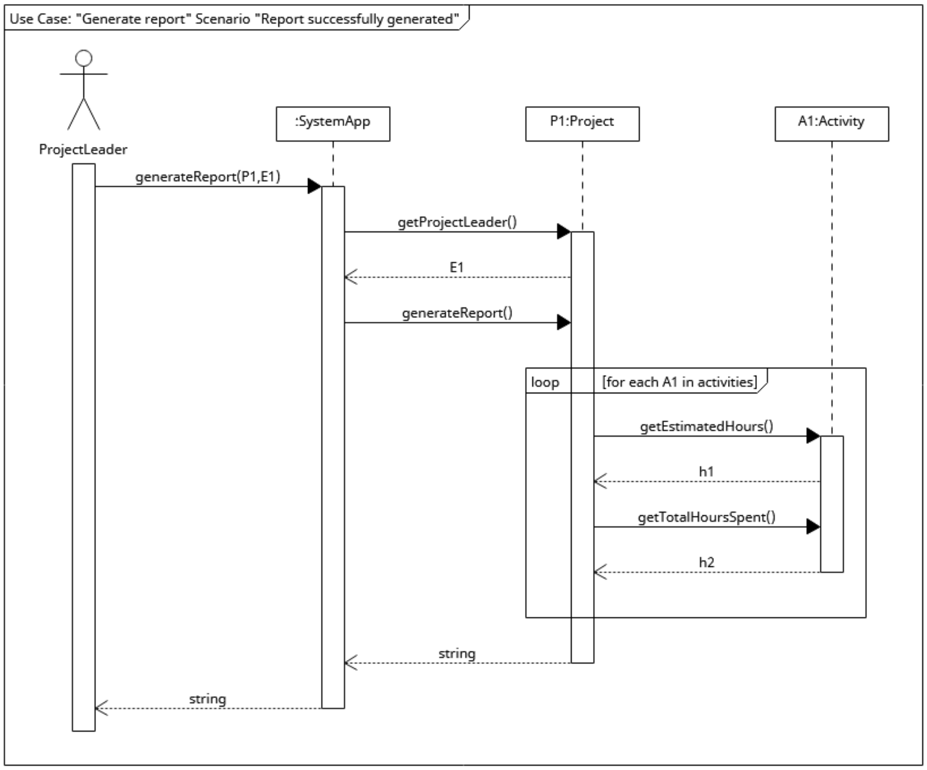

Figure 5: Sequence Diagram for Generate Report (Sebastian)

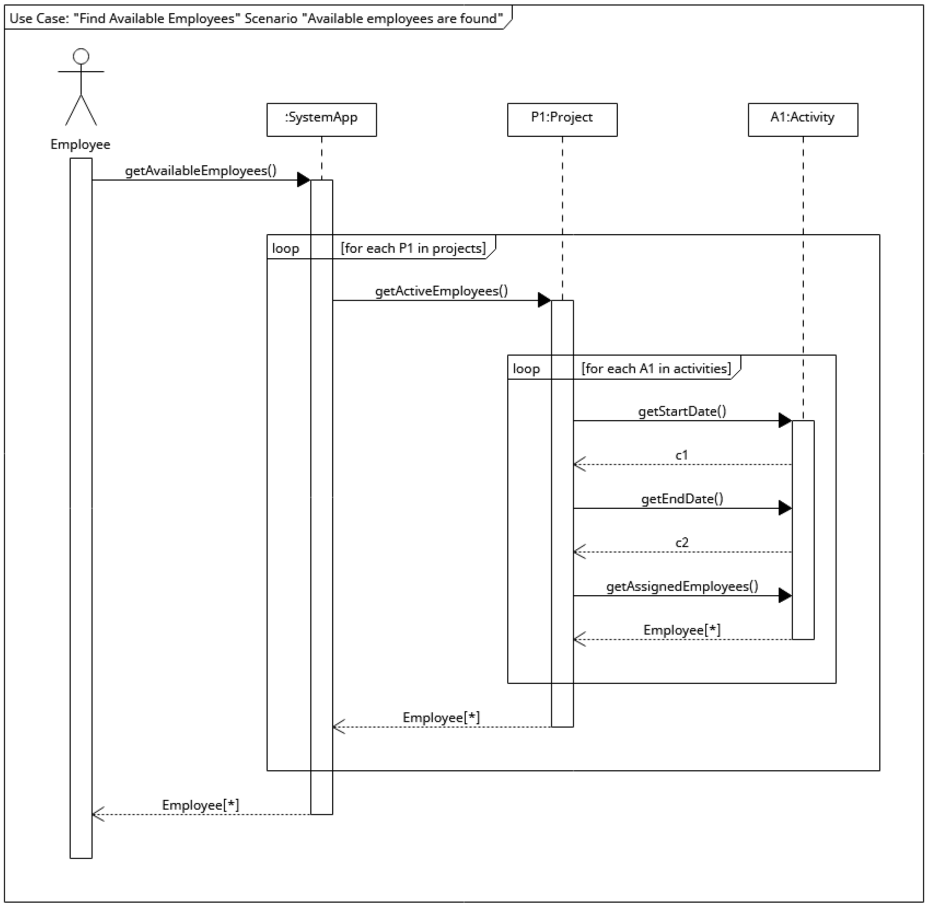

Figure 6: Sequence Diagram for Find Available Employees (Sebastian)

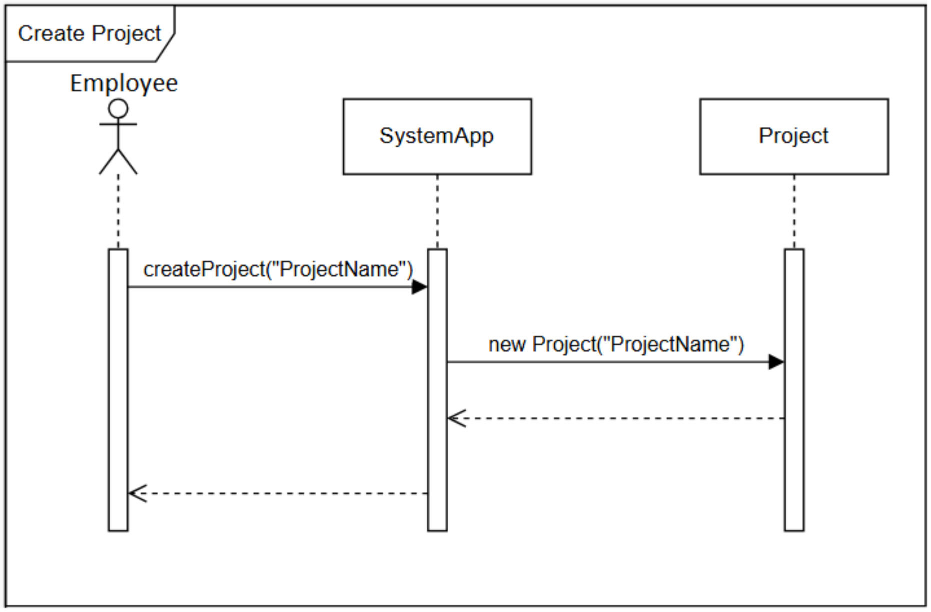

Figure 7: Sequence Diagram for Creating a Project (Bertram)

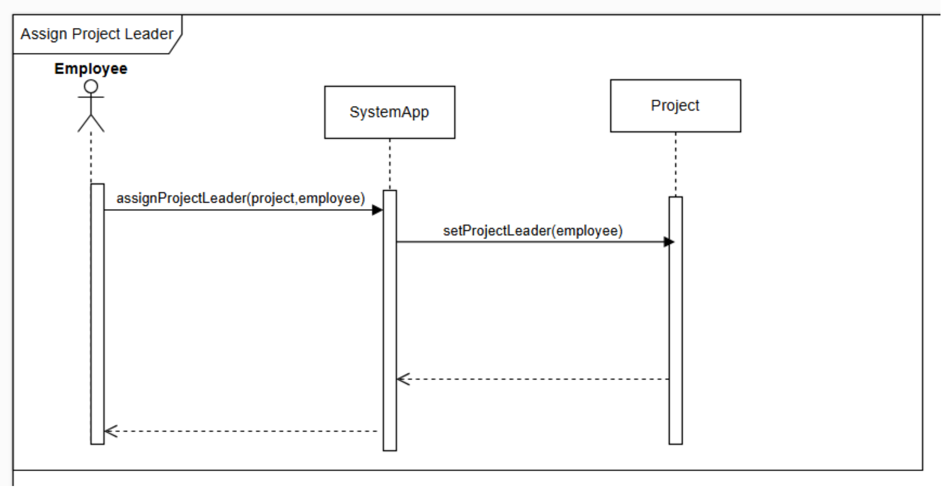
Assign Project Leader
Employee
SystemApp
Project
assignProjectLeader(project,employee)
setProjectLeader(employee)

Figure 8: Sequence Diagram for assigning a Project leader (Bertram)

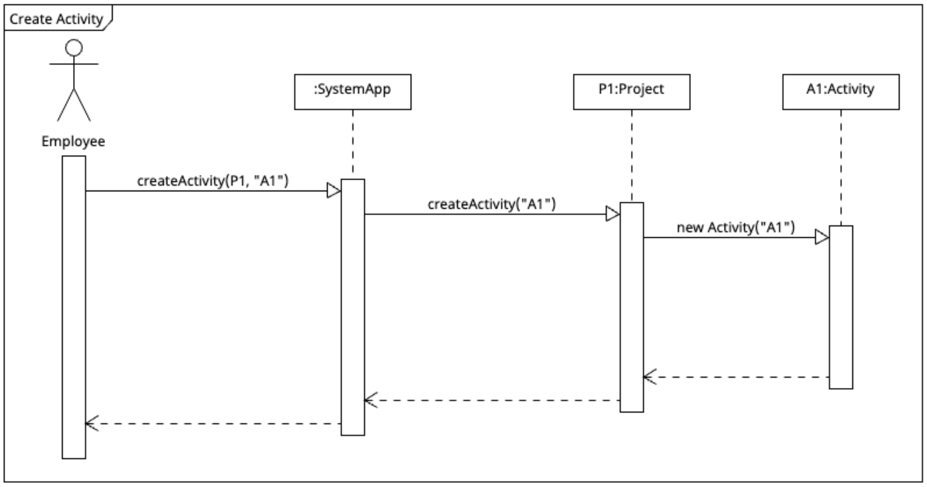

Figure 9: Sequence Diagram for Creating an Activity (Mads)

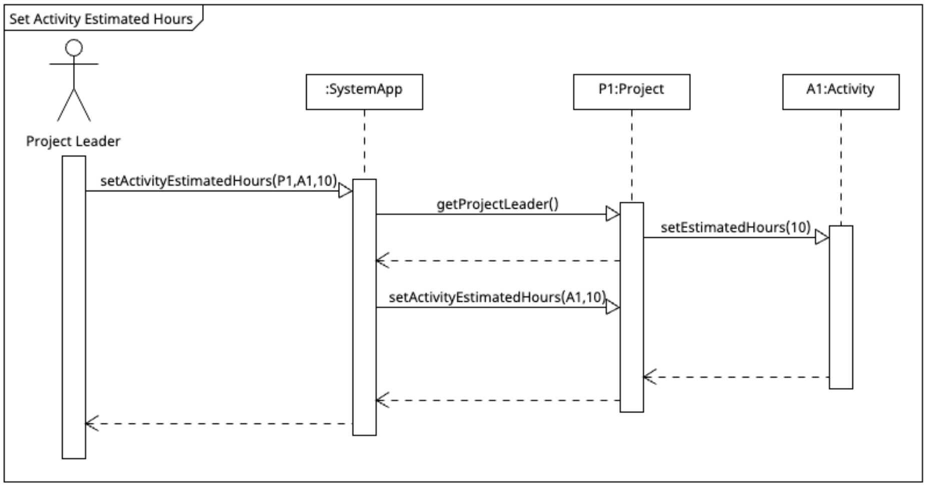
Set Activity Estimated Hours
Project Leader
:SystemApp
P1:Project
A1:Activity
setActivityEstimatedHours(P1,A1,10)
getProjectLeader()
setEstimatedHours(10)
setActivityEstimatedHours(A1,10)

Figure 10: Sequence Diagram for setting the estimated hours of an Activity (Mads)

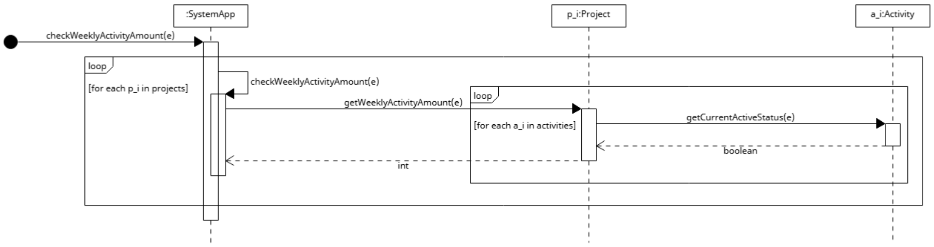
:SystemApp
p_i:Project
a_i:Activity
checkWeeklyActivityAmount(e)
loop
[for each p_i in projects]
checkWeeklyActivityAmount(e)
getWeeklyActivityAmount(e)
loop
[for each a_i in activities]
getCurrentActiveStatus(e)
boolean
int

Figure 11: Sequence Diagram for checking the amount of Activities an Employee is working on the given week (Dejan)

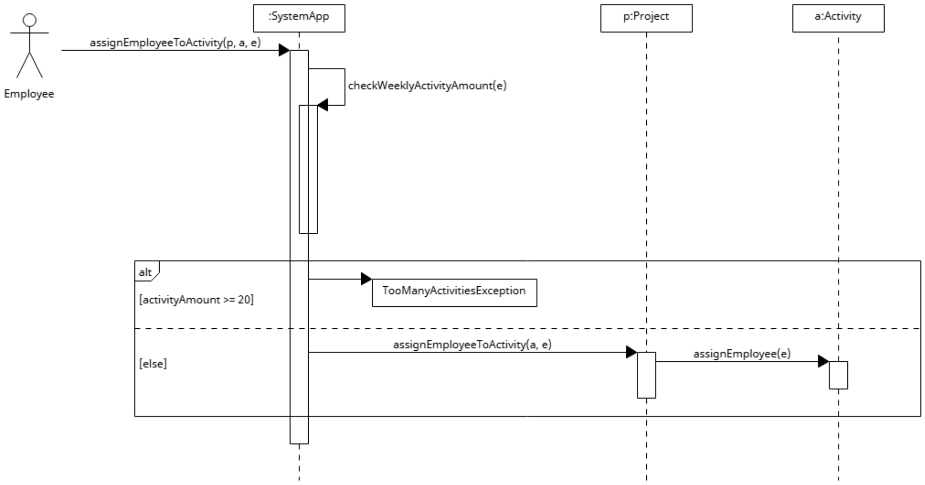
:SystemApp
p:Project
a:Activity
Employee
assignEmployeeToActivity(p, a, e)
checkWeeklyActivityAmount(e)
alt
[activityAmount >= 20]
TooManyActivitiesException
[else]
assignEmployeeToActivity(a, e)
assignEmployee(e)

Figure 12: Sequence Diagram for assigning an Employee to an Activity (Dejan)

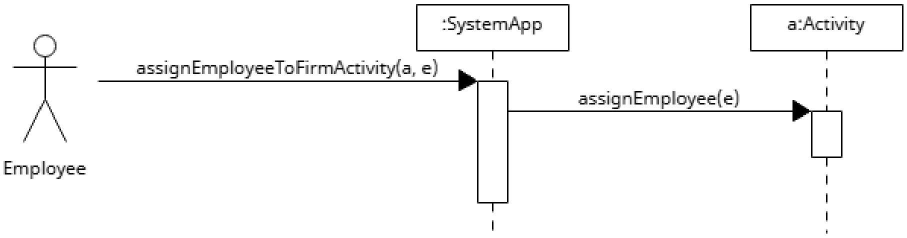

Figure 13: Sequence Diagram for assigning an Employee to a Firm Activity (Dejan)

3 Discussion

Besides that few uncertainties mentioned in Section 1.5, we have made the following decisions. We chosen to store the registered hours for each employee on each activity in a nested map, mapping the employee and the date of the registered hours to an integer holding the amount of halfhour-intervals registered. Furthermore we have decided to make the "id" for each project read only, as it will be assigned automatically with the creation of the project.