*Faculty of Natural Science*

*Department of Computer Science*

*Kristianstad University*

*Course: DA122B*

*Program: TGDU1*

*Authors: Shpat Krasniqi, Felix Karlsson, Martin Månsson*

HR-Application

Table of Contents

[1. Introduction 4](#_Toc511170667)

[2. Requirements 5](#_Toc511170668)

[2.1 Use Case Overview 5](#_Toc511170669)

[2.1.1 Manage users 6](#_Toc511170670)

[2.1.2 Remove users 6](#_Toc511170671)

[2.1.3 Add users 6](#_Toc511170672)

[2.1.4 Salary report 6](#_Toc511170673)

[2.1.5 Save as pdf 6](#_Toc511170674)

[2.1.6 Salary information 6](#_Toc511170675)

[2.1.7 Login information 6](#_Toc511170676)

[2.1.8 Forgot password 6](#_Toc511170677)

[2.1.9 Confirm login 6](#_Toc511170678)

[2.1.10 Job information 6](#_Toc511170679)

[2.1.11 Search 6](#_Toc511170680)

[2.1.12 Smart filter 6](#_Toc511170681)

[2.1.13 Personal information 6](#_Toc511170682)

[2.1.14 Edit personal information 6](#_Toc511170683)

[3. Design and Implementation 7](#_Toc511170684)

[3.1 Classes 7](#_Toc511170685)

[3.1.1 <Name of Class> 7](#_Toc511170686)

[3.1.2 <Name of Class> 7](#_Toc511170687)

[3.2 Class Interactions and Use Case mappings 7](#_Toc511170688)

[3.2.1 <Name of Use Case> 7](#_Toc511170689)

[3.3 Database 7](#_Toc511170690)

[4. Test Results 8](#_Toc511170691)

[5. Summary and Conclusion 9](#_Toc511170692)

[5.1 Weekly Progress 9](#_Toc511170693)

[5.1.1 Week 1 9](#_Toc511170694)

[5.1.2 Week 2 9](#_Toc511170695)

[5.1.3 Week 3 9](#_Toc511170696)

[5.1.4 Week 4 9](#_Toc511170697)

[5.1.5 Week 5 9](#_Toc511170698)

[5.1.6 Week 6 9](#_Toc511170699)

[5.1.7 Week 7 9](#_Toc511170700)

[5.2 Difficulties and challenges 9](#_Toc511170701)

[5.2.1 GitHub 9](#_Toc511170702)

[5.3 Correctness of time estimates 10](#_Toc511170703)

[5.4 Priority decisions 10](#_Toc511170704)

[5.5 Conclusion 10](#_Toc511170705)

[6 References 11](#_Toc511170706)

# Introduction

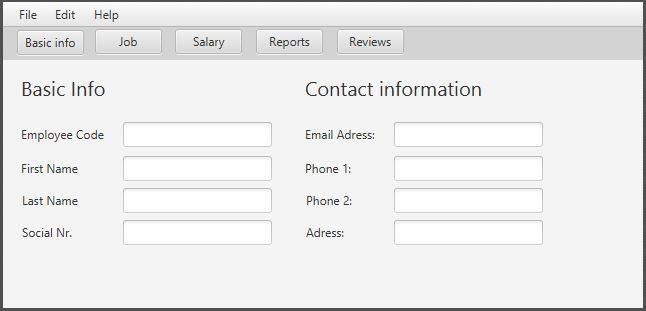
HR manager is a simple application aimed toward small to medium sized businesses to store, manage and access employee information including personal details and contact information, company roles, salary information and performance reviews.

The goal of this application is to help businesses’ HR departments to keep track of relevant employee information to increase efficiency by making everyday HR processes more manageable.

The applications functionality allows users to view and edit their personal and job information as well as view their compensation details, performance reviews and print related reports.

Admins are afforded all user privileges as well as the ability to add and remove users and edit their salary information. Admins also review employee performance.

The UI makes all navigation easy through several buttons that stand out at the top, dividing the information into logical categories and making it easily accessible and understandable.

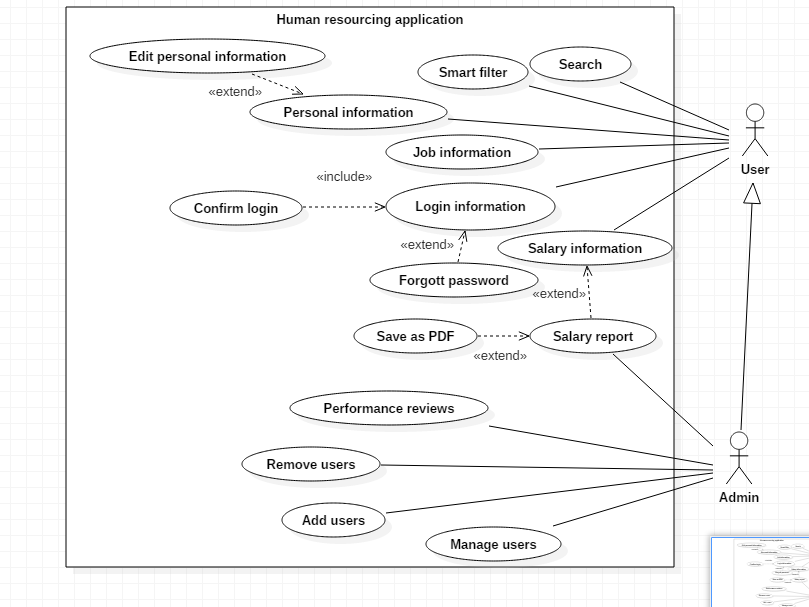
**Figure 1. Design example**

# Requirements

Table 1 - Requirements

|  |  |
| --- | --- |
| **Req.**  **No** | **Req. Name: Description** |
| 1 | Manage users: add/remove information as well as edit |
| 2 | Remove user: remove user from database |
| 3 | Add user: add user to the database |
| 4 | Salary report: be able to view salary information and history |
| 5 | Save as pdf: be able to save individual salary report as pdf |
| 6 | Salary information: view salary information |
| 7 | Login information: be able to login with username and password |
| 8 | Forgot password: be able to retrieve password from database by sending email |
| 9 | Confirm login: database shall confirm logins with information in database |
| 10 | Job information: view job information |
| 11 | Search: be able to search for information inside the database |
| 12 | Smart filter: be able to filter certain information |
| 13 | Personal information: view users own information |
| 14 | Edit personal information: be able to edit own information for user |

## Use Case Overview



### Manage users

In Manage users, users can be edited. Information can be removed or added.

### 2.1.2 Remove users

In Remove users, Admin can delete users.

### 2.1.3 Add users

In Add users, Admin can create users.

### 2.1.4 Salary report

In Salary report, Admin can view salary specification.

### 2.1.5 Save as pdf

In Save as pdf, Admin can save the Salary report as pdf to then later print it.

### 2.1.6 Salary information

In Salary information, the user can view things like wage, employee time and salary history.

### 2.1.7 Login information

In login information, the user shall enter their SSN and personal password.

### 

### 2.1.8 Forgot password

In Forgot password, the user will get a random number to be able to edit his forgotten password.

### 2.1.9 Confirm login

In Confirm login, SSN and password will be checked if correct.

### 2.1.10 Job information

In Job information, the user can view

### 2.1.11 Search

In Search, the user can search after

### 2.1.12 Smart filter

In Smart filter, the user can do more advanced searching

### 2.1.13 Personal information

In Personal information, the user can access profiles of other personnel or his own information.

### 2.1.14 Edit personal information

In Edit personal information, the user can edit personal information.

# Design and Implementation

< Instructions: Describe your design in this chapter. >

## Classes

<Description of classes. List one class per sub chapter and add some class diagrams to illustrate relations (inheritance and/or associations) between the main classes. The UML does not need to be extremely detailed, but the most important attributes and methods shall be shown.>

### 3.1.1 <Name of Class>

< Description of this class, including UML. >

### 3.1.2 <Name of Class>

…

## Class Interactions and Use Case mappings

<Instructions: the sub chapters here shall correspond to the use cases in chapter 2, and each use case shall contain a UML **sequence diagram** of the classes that are involved in that use case, and how they interact to implement the use case, including method calls. >

### 3.2.1 <Name of Use Case>

<A sequence diagram of the classes involved in this use case, and how they interact. You may write some explaining text here, and/or you may use notes in the diagram itself.>

## Database

<Show your database design with ER diagram(s). >

# Test Results

Table 2 below contains the current status of implemented and tested requirements.

<Instructions: This table shall map 1-1 to the table in Chapter 2. The test result for each requirement shall be one of the following: NOT IMPLEMENTED, PASSED or FAILED.>

Table 2 - Test Results

|  |  |  |
| --- | --- | --- |
| **Req.**  **No** | **Req. Name** | **Test Result** |
| 1 | <Requirement 1 name> | <NOT IMPLEMENTED/PASSED/FAILED> |
| 2 | <Requirement 2 name> | … |
| .. | .. | .. |

# Summary and Conclusion

This chapter contains a summary and conclusion of the work that was carried out in this project as well as reflections and thoughts about working methods and challenges.

## Weekly Progress

Below is a short summary of what was done each week.

### Week 1

We have formed the current project group which we all feel comfortable about. We had 2 meetings in which we discussed the preliminary features of the application and what the contents of this report should be. We agreed upon the chosen application, we have created a use case diagram and our ideas and presented there. GitHub was downloaded and we did have a few problems setting everything up but it was solved during a help session. We are all very eager to start this project up.

### 5.1.2 Week 2

TODO

### 5.1.3 Week 3

TODO.

### 5.1.4 Week 4

TODO.

### 5.1.5 Week 5

TODO.

### 5.1.6 Week 6

TODO.

### 5.1.7 Week 7

TODO.

## Difficulties and challenges

Below is a list of notable challenges that came up during this project and that took a long time to solve.

### 5.2.1 GitHub

Setting up GitHub was a challenge, we accidentally used git init in a cloned project and therefore could not push it to the master on Github since technically it was a new project because of git init. It took a while to solve the issue with the help of Andreas.

## Correctness of time estimates

<Instructions: Look back on your time estimates and discuss your results. How accurate were they? What have you learned about time estimates and how can you get better in next project?>

## Priority decisions

<Instructions: Look back on your feature priority settings. Did you prioritize the right features? Did you succeed to deliver the highest prioritized features? Have you learned anything about setting priorities>

## Conclusion

<Instructions: Look back on the whole project. Here you can write a bit more freely about your thoughts on this project. What was your overall experience? How was the teamwork? What did you learn? Can you list some points that you will do better in next project? Other thoughts. >

# References

<Instructions: In this chapter, you shall list references to external sources, books, web sites, etc. In this document we use the **Vancouver Referencing System**, also called the author-number system. >