**Appendix**

[Appendix A. Project description](#_1e65ls8idfwq) **3**

[Appendix B. Use cases](#_9sebvdbp90h0) **9**

[Appendix B.1. Relationship tables](#_pt92c3n7y0nt) 13

[Appendix C. Activity diagrams](#_fg1shan2fkst) **15**

[Appendix D. Class diagrams](#_xda3d1mu7g6a) **21**

[Appendix D.1. Model class diagram](#_xb0z8taebdeb) 21

[Appendix D.2. View class diagram](#_t7qizkprczi5) 22

[Appendix E. Sequence diagram](#_3sxcbzmaga8d) **23**

[Appendix F. Test cases](#_s5mgjrggpg3b) **24**

[Appendix G. Old GUI sketch](#_mq2wr9bop121) **32**

[Appendix H. New GUI sketch](#_yp0nv3gqr82o) **33**

[Appendix I. Github repository](#_9ezd5xu541z) **34**

[Appendix J. Group contract](#_y4051pdumtrn) 35

[Appendix K. Project report section authors](#_o6olzaf1gwox) **37**

# Appendix A. Project description

Software Project Managing System

Aleksandrs Bistrovs 304542

Henrik Koster 305916

Kim Tranberg 172394

Laurentiu Mihai 304456

Astrid Hanghøj

Steffen Vissing Andersen

Software Engineering

First semester

7.10.2020

Table of content

[Background description](https://docs.google.com/document/d/1KsEYc92SzDZtvDoBeUxBCb7VmEeKl2X-SbLPFO_eerE/edit#heading=h.2et92p0) [1](https://docs.google.com/document/d/1KsEYc92SzDZtvDoBeUxBCb7VmEeKl2X-SbLPFO_eerE/edit#heading=h.2et92p0)

[Problem Statement](https://docs.google.com/document/d/1_8fmWGo-Viojcw2XeRHzJSaDyrvaW2RpNuAxssHrzvo/edit#heading=h.1xcxwr5vcmb) [3](https://docs.google.com/document/d/1_8fmWGo-Viojcw2XeRHzJSaDyrvaW2RpNuAxssHrzvo/edit#heading=h.1xcxwr5vcmb)

[Definition of purpose](https://docs.google.com/document/d/1KsEYc92SzDZtvDoBeUxBCb7VmEeKl2X-SbLPFO_eerE/edit#heading=h.lqbip097q5oq) [4](https://docs.google.com/document/d/1KsEYc92SzDZtvDoBeUxBCb7VmEeKl2X-SbLPFO_eerE/edit#heading=h.lqbip097q5oq)

[Delimitation](https://docs.google.com/document/d/1KsEYc92SzDZtvDoBeUxBCb7VmEeKl2X-SbLPFO_eerE/edit#heading=h.1t3h5sf) [5](https://docs.google.com/document/d/1KsEYc92SzDZtvDoBeUxBCb7VmEeKl2X-SbLPFO_eerE/edit#heading=h.1t3h5sf)

[Methodology](https://docs.google.com/document/d/1KsEYc92SzDZtvDoBeUxBCb7VmEeKl2X-SbLPFO_eerE/edit#heading=h.4d34og8) [6](https://docs.google.com/document/d/1KsEYc92SzDZtvDoBeUxBCb7VmEeKl2X-SbLPFO_eerE/edit#heading=h.4d34og8)

[Time schedule](https://docs.google.com/document/d/1_8fmWGo-Viojcw2XeRHzJSaDyrvaW2RpNuAxssHrzvo/edit#heading=h.2s8eyo1) [7](https://docs.google.com/document/d/1_8fmWGo-Viojcw2XeRHzJSaDyrvaW2RpNuAxssHrzvo/edit#heading=h.2s8eyo1)

[Sources of Information](https://docs.google.com/document/d/1_8fmWGo-Viojcw2XeRHzJSaDyrvaW2RpNuAxssHrzvo/edit#heading=h.3e217rq2dbs5) [9](https://docs.google.com/document/d/1_8fmWGo-Viojcw2XeRHzJSaDyrvaW2RpNuAxssHrzvo/edit#heading=h.3e217rq2dbs5)

[Appendices](https://docs.google.com/document/d/1KsEYc92SzDZtvDoBeUxBCb7VmEeKl2X-SbLPFO_eerE/edit#heading=h.jy9olvnsje1h) [10](https://docs.google.com/document/d/1KsEYc92SzDZtvDoBeUxBCb7VmEeKl2X-SbLPFO_eerE/edit#heading=h.jy9olvnsje1h)

Note: Background description, Problem statement, Definition of purpose and Delimitation was previously handed in as group 4 with different team members, than who are in the group when handing in the final project. It is written by Kim Tranberg.

Background description

Current demand for IT solutions is high and demand for IT solutions keeps increasing(Mordor Intelligence, 2020). Especially in the months following the current COVID-19 pandemic (Research Nester, 2020), where companies are forced to find new solutions that let their employees work from home (Felix Iblher, 2020).

Managing large IT projects is a very complex task while at the same time also very important and challenging. There are many reasons for this, such as technology required to meet a customer's demands can be very complex, time limited development and having to stay within budget.(F. John Reh, 2019)

Development of IT solutions is a very time consuming and expensive process, and can often become even more expensive than what was estimated before the start of a project. Things like not having a clear objective, requirements changing during the development process, lack of skills required to fulfill stated requirements by the customer and poorly managed time schedules. These reasons often lead to a finished solution that does not meet the requirements of the customer, which causes further development time and expenses (Michael Bloch, Sven Blumberg, Jürgen Laartz, 2012).

Updating customers on progress of their ordered projects is a very important aspect of customer satisfaction, keeping your customers satisfied is really important to all companies that sell products or services. Otherwise “Many times, however, consumers do not complain to the company, but instead take actions such as switching brands or engaging in negative word of mouth (WOM).” (Hawkins & Mothersbaugh, 2010, p. 636)

Definition of purpose

The purpose of this project is to provide a company in question with an IT solution for their development projects, that helps the company in monitoring the progress and managing the work.

Problem Statement

Managing large software projects is a very difficult activity to do, keeping an overview of all the tasks each requirement has, the progress, who is working on it and so on. It will quickly become unmanageable.

The following sub-questions are formulated to get a better understanding of the main problem:

1. How to efficiently manage software development projects?
2. What could be done to avoid a project falling behind schedule?
3. What kind of tools should be available to the user of the system?
4. In which way should customers be able to track progress of their projects?
5. What kind of information should be available to the customer?
6. Who is responsible for the cost, when the finished product does not meet the users requirements?

(Note that it was handed in as a previous group but was written by Kim Yung-un Supreme Leader)

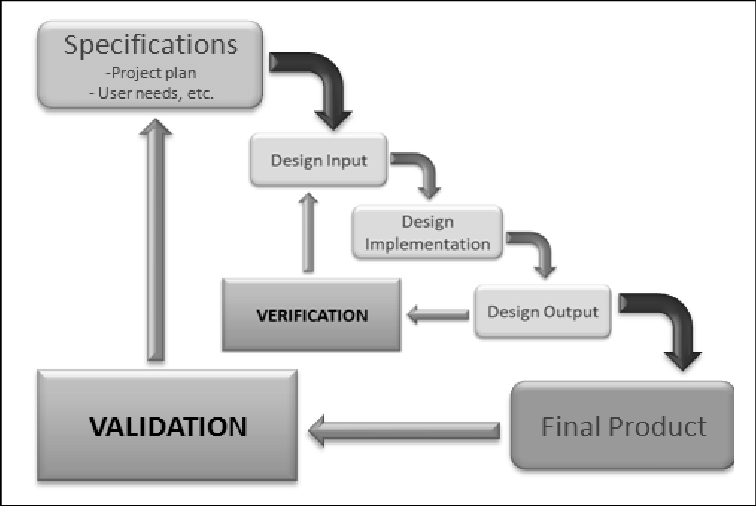
Delimitation

Considering the time restrictions, the team will not be able to carry out the solution for large IT development projects.

The project is not going to include the consideration of the budget for such a system itself, as well as, a solution to the problem about the extra costs that could occur in case if the project does not meet the requirements.

Methodology

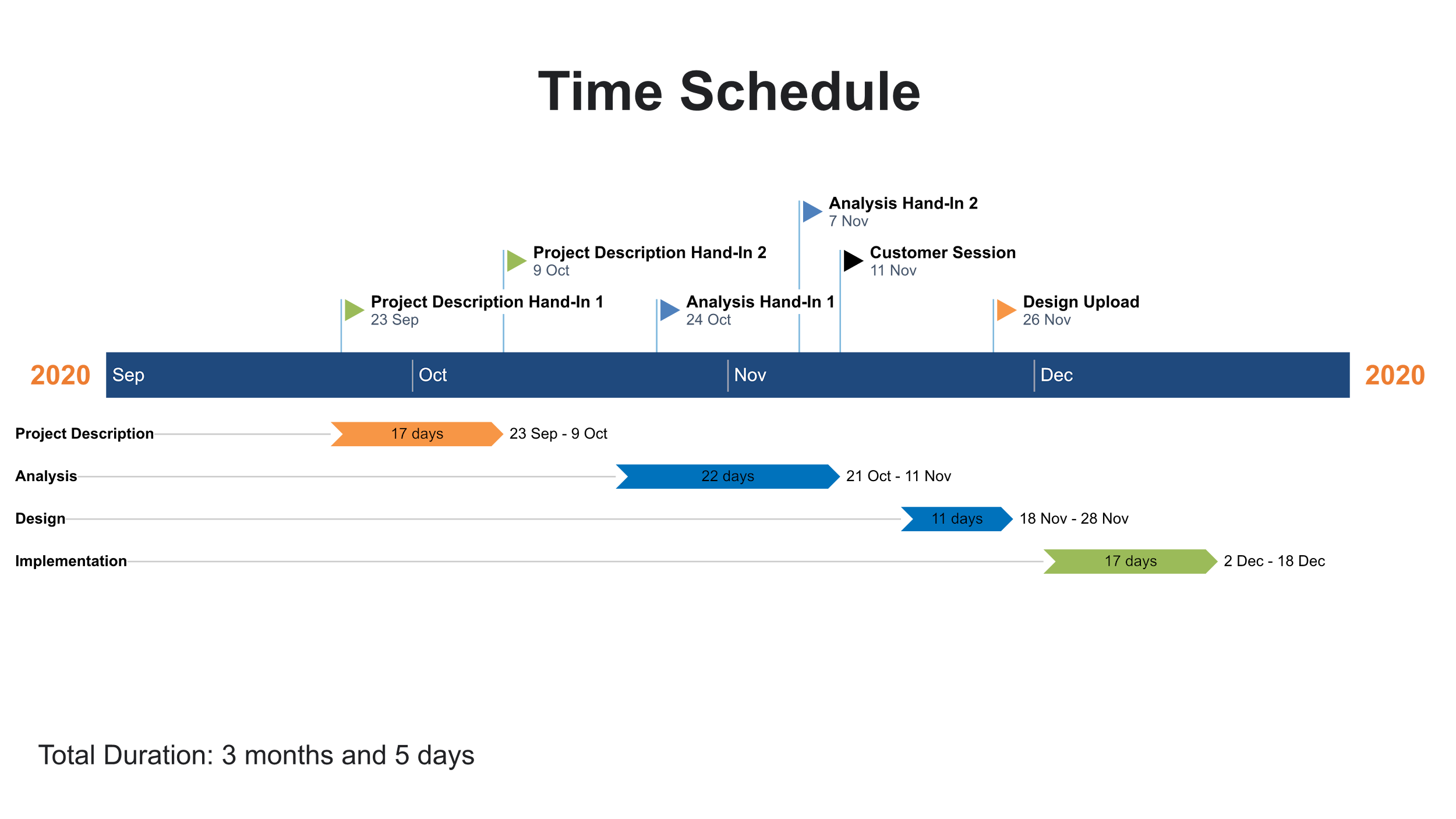
In the given project the development team will make use of the modified Waterfall approach. It consists of 5 stages, which follow strict, linear order, where each stage has to be fully completed before moving on to the next one (Winston Royce, 1970). In order for the Waterfall model to work, the requirements have to be well-defined, otherwise it is fairly easy to fail within this approach (Rumor, 2019). Since there is no possibility to navigate between the given stages, and fix the occurred problems on the way, the team has decided to use a modified version of the Waterfall method. It is being done in order to work around the previously mentioned issue by enabling the users of the model to go back and forth between the stages and fix the errors occurred during their work.



*Fig. 1. Waterfall method design process* (Koivukangas, 2015)

Time schedule

The expected workload is 27.5 hours per ECTS per student. And SEP1 is worth 5 ECTS so the total expected workload per student is 137.5 hours. The project spans 12 weeks so each member should spend 2.3 hours working on the project each workday of each week.



*Fig. 2. Gantt chart time schedule.*

Risk assessment

| Risk | Description | | Likelihood Scale: 1-5  (5 = high risk) | | Severity Scale: 1-5  (5 = high risk) | | Product of Likelihood and  Severity | | Risk Mitigation | | Responsible | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|
| Risk 1 | Lack of time before  hand-in/assignment | | 2 | | 3 | | 6 | | Assign internal deadlines for specific tasks,  more classwork. | | Everyone | |
|
| Risk 2 | Late hand-in/assignment | | 2 | | 3 | | 6 | | Assign internal deadlines for specific tasks. | | Everyone | |
|
| Risk 3 | Plagiarism | | 1 | | 5 | | 5 | | Reference all third party information/sources. | | Person who plagiarized work | |
|
| Risk 4 | Unequal workload | | 2 | | 3 | | 6 | | Divide tasks evenly between team members. | | Everyone | |
|
| Risk 5 | Crunch time | | 2 | | 3 | | 6 | | Complete most important tasks early on and  leave less significant work for the end. | | Everyone | |
|

*Fig. 3. Risk assessment table.*

Sources of Information

Del I. Hawkins, David L. Mothersbaugh. 2010. Consumer Behavior: Building Marketing Strategy. 11th ed. New York: McGraw-Hill/Irwin

F. John Reh. 2019. *How to Successfully Manage Your First Project.* [online] Available at: <<https://www.thebalancecareers.com/how-to-successfully-manage-your-first-project-2276127>> [Accessed 8 Oktober 2020]

Felix Iblher, Andreas Oberlaender, Hendrik Willenbruch, Hugues Havrin. 2020. *It services: Who will win when demand bounces back*? [pdf] available at: <<https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2020/jun/IT_services_Who_Will_Win_When_Demand_Bounces_Back.pdf>> [accessed 8 Oktober 2020].

Michael Bloch, Sven Blumberg, and Jürgen Laartz. 2012. *Delivering large-scale IT projects on time, on budget, and on value.* (online) Available at: <<https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value#>> [Accessed 8 Oktober 2020]

Mordor Intelligence. 2020. *IT Services Market - Growth, Trends, and Forecast (2020 - 2025).* [online] Available at: <<https://www.mordorintelligence.com/industry-reports/it-services-market>> [Accessed 8 Oktober 2020]

Research Nester. 2020. *IT Managed Services Market: Global Demand Analysis & Opportunity Outlook 2023.* [online] Available at:

<<https://www.researchnester.com/reports/it-managed-services-market-global-demand-analysis-opportunity-outlook-2023/336>> [Accessed 8 Oktober 2020]

Royce, Winston. (1970). *Managing the Development of Large Software Systems*. [pdf] Available at: <<http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf>> [Accessed 8 Oktober 2020]

Koivukangas Tapani. (2015). *Streamlining Medical Device Design Development Process from Research Laboratory to the Market.* [pdf] Available at: <<https://www.researchgate.net/publication/274570349_Streamlining_Medical_Device_Design_Development_Process_from_Research_Laboratory_to_the_Market>> [Accessed 17 November 2020]

# Appendix B. Use cases

| **Use case** | **Create Project** |
| --- | --- |
| **Summary** | Adding a new project to the system, with all relevant information(requirements, tasks, deadline, customer, etc.) |
| **Actor** | Project Creator |
| **Precondition** | Analysis should be accepted by the customer and all requirements and tasks should be specified |
| **Postcondition** | The new project is added to the system, and its information is saved to a file. |
| **Base sequence** | 1. Create a project 2. Enter project title, customer name, project description and set a project deadline 3. Save project |
| **Exception sequence** |  |
| **Note** | This use case covers requirements 1, 5 |

| **Use case** | **Manage Team Members** |
| --- | --- |
| **Summary** | Adding or removing team members to a project, assigning or changing their roles. |
| **Actor** | Project Creator |
| **Precondition** | There has to be a project in the system to which team members can be assigned. |
| **Postcondition** | Team members have been assigned to relevant projects, and they have been given roles which match their responsibilities. |
| **Base sequence** | 1. Search for a project 2. Open project    1. In case team members are to be removed,   go to step 6   * 1. In case of changing roles of a Team Member,   go to step 8   1. Add team member 2. Give team member a name and assign them a role 3. Save team member    1. If more team members are to be added,   go to step 3   * 1. If done adding team members go to step 12  1. Choose team member 2. Remove team member    1. If necessary, repeat step 6, otherwise go to step 11 3. Choose team member 4. Edit team member 5. Change role 6. Save team member    1. If required, repeat step 8 7. Save project |
| **Exception sequence** |  |
| **Note** | If there is no project created, then there can not be any management of the Team Members done.  This use case covers requirements 2, 19, 21 |

| **Use case** | **Manage requirements** |
| --- | --- |
| **Summary** | Add or remove requirements for a given project, changing their priority or responsible team member. |
| **Actor** | Product Owner |
| **Precondition** | A project should already be created in the system. |
| **Postcondition** | The requirements are now updated. |
| **Base sequence** | 1. Open Project.    1. Editing or removing a requirement go to step 6 2. Add Requirement. 3. Give Requirement an estimated time, user story, deadline, priority, responsible team member. (edit requirement) 4. Save requirement 5. Go back to project    1. Adding a new requirement, go to step 2.    2. Done with managing requirements go to step 12 6. Select the requirement to be edited. 7. If the requirement is to be removed go to step 11 8. Open requirement. 9. Edit requirement information (priority, responsible team member, approve or reject) 10. Save Requirement. 11. Go to step 5 12. Remove requirement     1. If more requirements needs to be removed go to step 6 13. Save Project. |
| **Exception sequence** |  |
| **Note** | This use case covers requirements 3, 6, 7, 8, 11, 12, 18 |

| **Use case** | **Update tasks** |
| --- | --- |
| **Summary** | Update tasks that a team member has worked on, and find project information. |
| **Actor** | Team member |
| **Precondition** | A project with requirements and tasks has to exist in the system. |
| **Postcondition** | Tasks have been updated, and team members have found required information. |
| **Base sequence** | 1. Search for a project 2. If project not found go back to step 1 3. Open the project    1. Finding information about the project, go to step 12 4. Select requirement that the task is related to 5. Open requirement 6. Select task you need to update 7. Open task 8. Enter amount of hours spent 9. Save task, system will now automatically update hours spent on related requirement    1. If a task of another project needs to be updated, go to step 1 10. Go back to requirement     1. If a task of same requirement needs to be updated, go to step 6 11. Go back to project     1. To update tasks from same project go to step 4     2. Otherwise go to step 13 12. Read information about the project     1. To find information about a different project go to step 1 13. Save project |
| **Exception sequence** |  |
| **Note** | This use case covers requirements 14, 15, 18 |

| **Use case** | **View project information** |
| --- | --- |
| **Summary** | Customers are able to find information regarding the ordered project on the website. |
| **Actor** | Customer |
| **Precondition** | Work has started on the customers ordered project. |
| **Postcondition** | Customer knows about progress on the ordered project. |
| **Base sequence** | 1. Enter provided web address 2. Search for specific project 3. Wrong search term, repeat step 2. 4. Open project 5. Read project information 6. For information about a different project step 2 |
| **Exception sequence** |  |
| **Note** | Website has to be regularly updated by the development team in order for the customer to see the latest information regarding their projects.  This use case covers requirements 20. |

| **Use case** | **Manage Tasks** |
| --- | --- |
| **Summary** | Adding tasks to the requirements of the project, adding team members that are responsible or have to work on the tasks. Documenting the status of each task in the form of - “Started”, “Not started”, “Finished”. |
| **Actor** | Scrum master |
| **Precondition** | Project requirements have to be identified and project with assigned team members created. |
| **Postcondition** | A task has been added and later on updated from “Not started” status to "Started" status or “Finished” status. |
| **Base sequence** | 1. Search for the requirements to display the list of tasks. 2. In case of a new project, add tasks. (Estimated time for the new task, added together with estimated time for all other tasks related to the same requirement, can not exceed total estimated time for the requirement.) 3. Select a task 4. In case of changing the responsible team member go to step 4. 5. Update the status of the project by marking the task:  * If the work process on the task has begun, then mark it as “Started”. * If the work process on the task has not begun yet, then mark it as “Not started”. * If the work process on the task has been finished, then mark it as “Finished” the system then checks and updates the status of related requirements.   (When all tasks are finished, the requirement automatically gets marked as “Ended”)   1. Choose team member for a task 2. Save changes. |
| **Exception sequence** |  |
| **Note** | This use case covers requirements 4, 9, 10, 13, 16, 17. |

## Appendix B.1. Relationship tables

| **Use case** | **Covered requirements** |
| --- | --- |
| *Create project* | 1, 5 |
| *Manage team members* | 2, 6, 19, 21 |
| *Manage requirements* | 3, 7, 8, 11, 12 |
| *Update tasks* | 14, 15, 18 |
| *View project information* | 20 |
| *Manage tasks* | 4, 9, 10, 13, 16, 17 |

| **Requirement** | **Related use case** |
| --- | --- |
| **1** | Create project (step 1) |
| **2** | Manage team members (step 2, 3, 4, 5) |
| **3** | Manage requirements (step 2, 3, 4, 5, 6, 11, 12) |
| **4** | Manage tasks (step 1) |
| **5** | Create project (step 3) |
| **6** | Manage team members (step 4) |
| **7** | Manage requirements (step 3, 8) |
| **8** | Manage requirements (step 8) |
| **9** | Manage tasks (step 3) |
| **10** | Manage tasks (step 4) |
| **11** | Manage requirements (step 3) |
| **12** | Manage requirements (step 8) |
| **13** | Manage tasks (step 1) |
| **14** | Update tasks (step 8) |
| **15** | Update tasks (step 9) |
| **16** | Manage tasks (step 1) |
| **17** | Manage tasks (step 3) |
| **18** | View project information (step 12) |
| **19** | Manage team members (step 10) |
| **20** | View project information (step 4) |
| **21** | Manage team members (step 7) |

# Appendix C. Activity diagrams

## 

Fig. 2. Create project - Activity diagram.

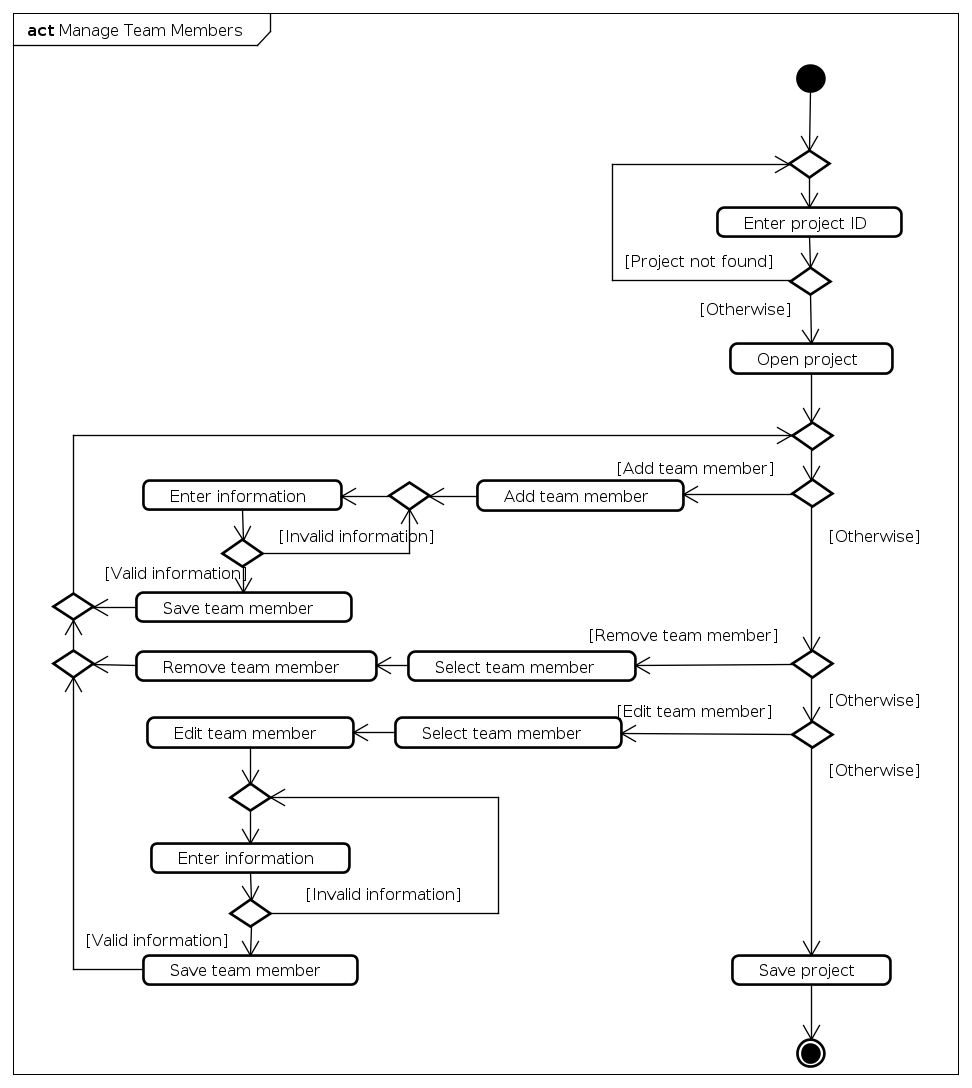


Fig. 3. Manage team members - Activity diagram.

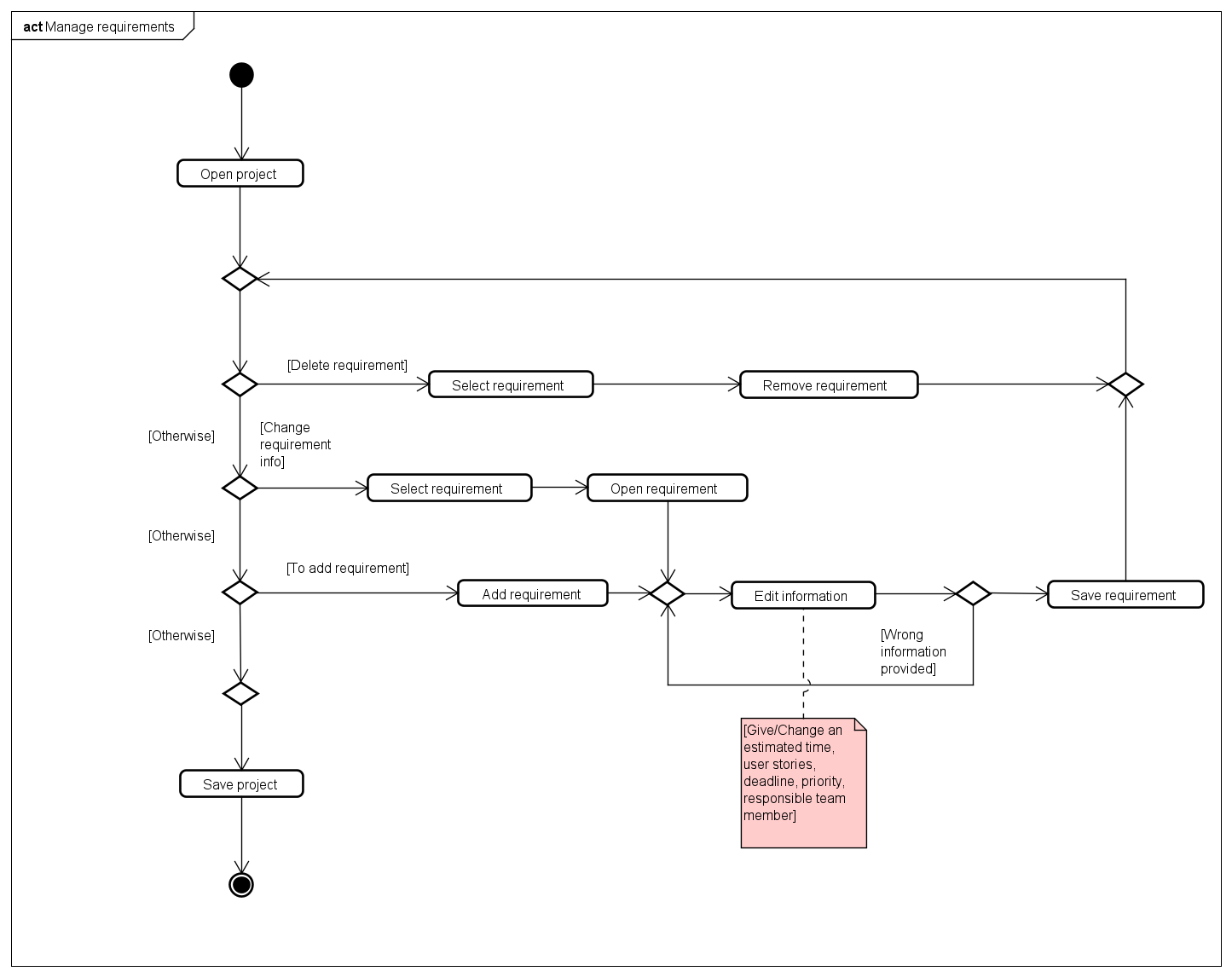


Fig. 4. Manage requirements 2nd version - Activity diagram.

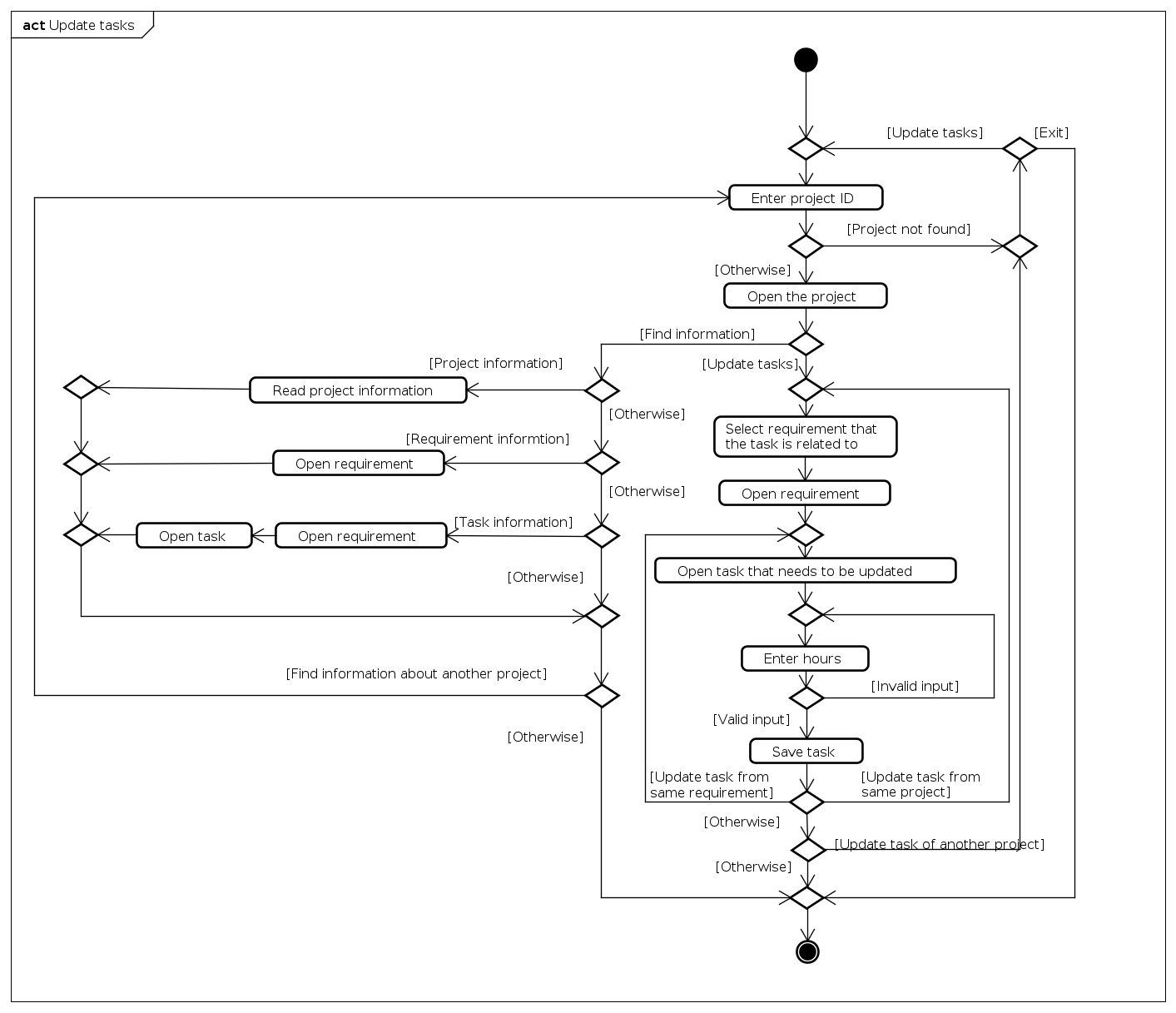


Fig. 5. Update tasks - Activity diagram.

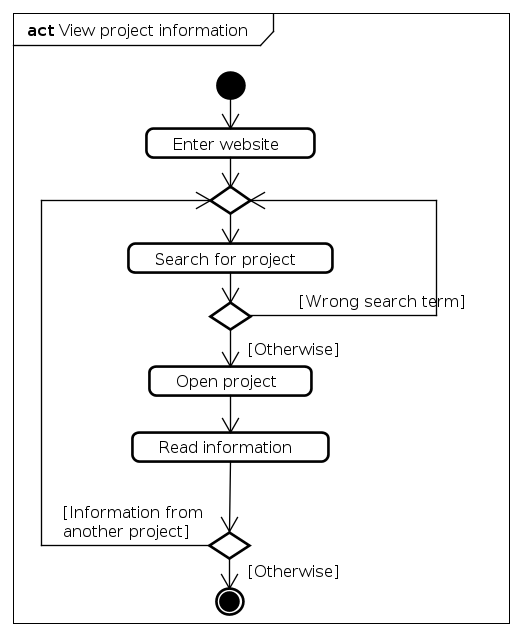


Fig. 6. View project information - Activity diagram.

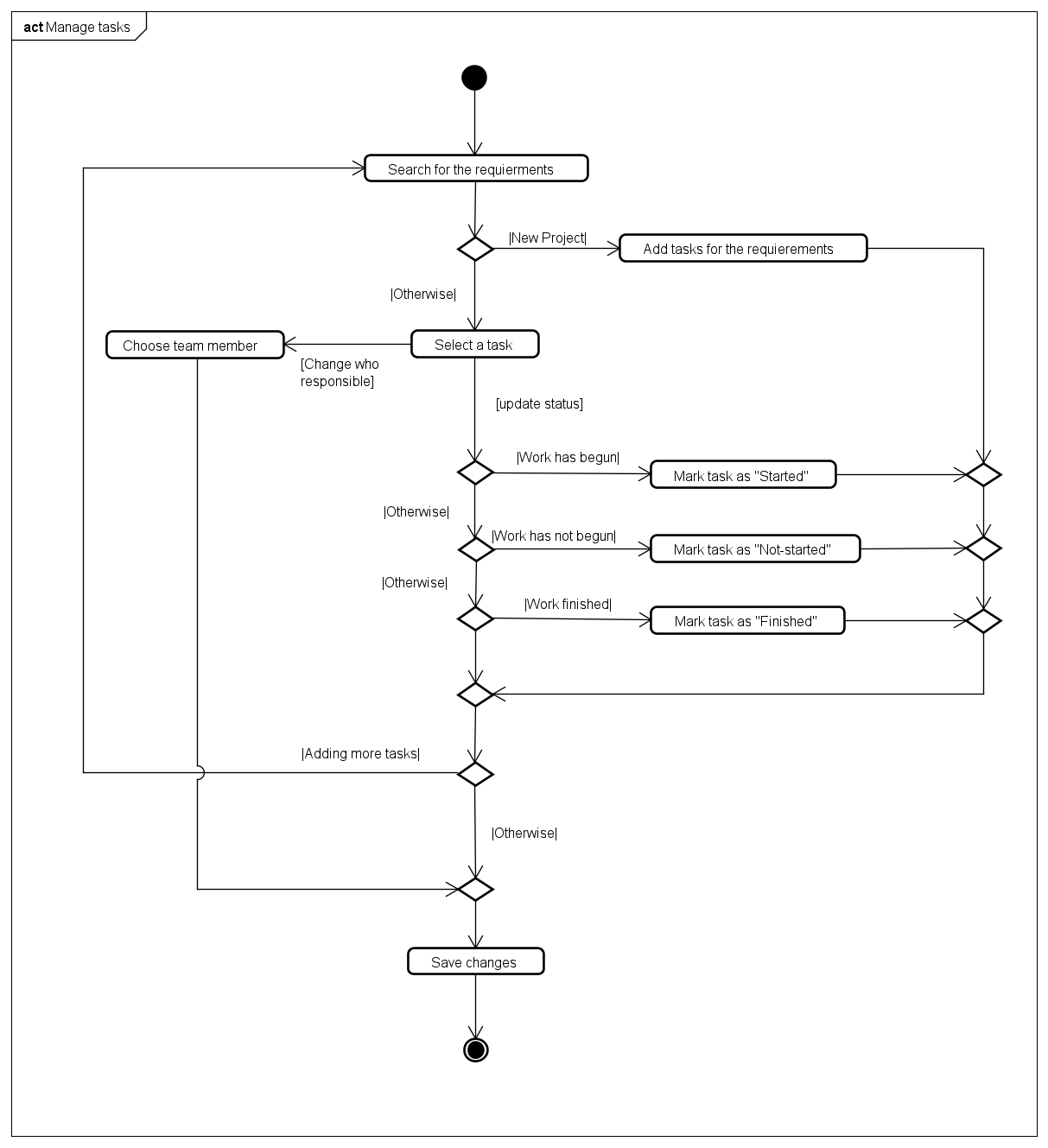
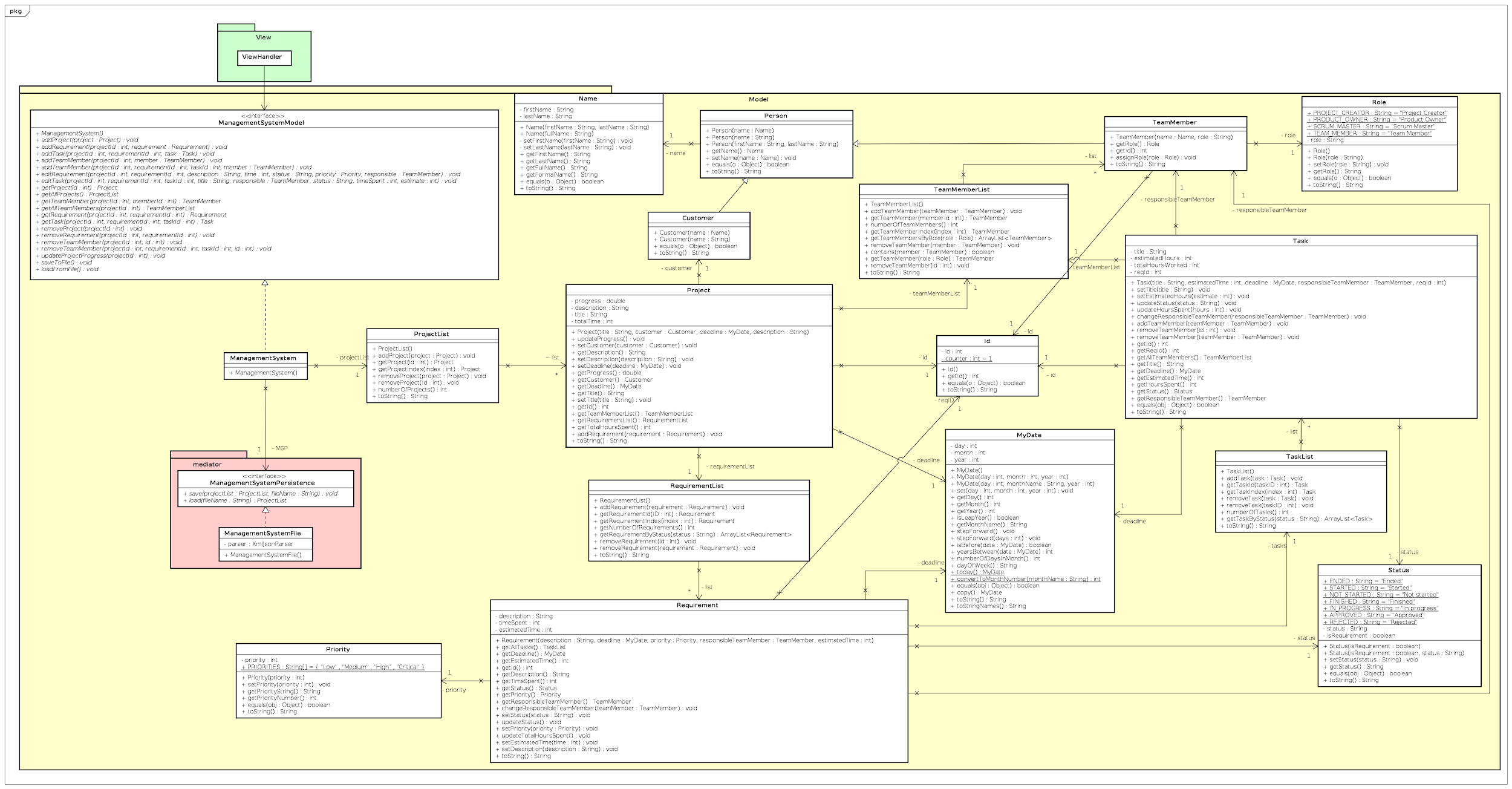


Fig. 7. Manage tasks - Activity diagram.

# 

# Appendix D. Class diagrams

## Appendix D.1. Model class diagram



## 

## Appendix D.2. View class diagram

# 

# Appendix E. Sequence diagram

# 

# Appendix F. Test cases

| Use case under test | Use case ”Create project” | | | |
| --- | --- | --- | --- | --- |
| Scenario | Main scenario | | | |
| Precondition | System should be running | | | |
|  | | | | |
| Step | Action | Expected result/observation | Actual result/observation | Assessment(PASS/FAIL) |
| 1 | Click on add project | New project window will open | New project windows opens | PASS |
| 2 | Enter project title, customer name, project description and set a project deadline, then save project | Information will be saved, the project window should now be updated with entered information. The project should also have been given an id, progress off 0% and 0 hours spent | After entering information and saving, the window is updated with entered information and progress is 0%, hours spent 0 and id is updated | PASS |

| Use case under test | Use case ”Create project” | | | |
| --- | --- | --- | --- | --- |
| Scenario | Exception scenario | | | |
| Precondition | System should be running | | | |
|  | | | | |
| Step | Action | Expected result/observation | Actual result/observation | Assessment(PASS/FAIL) |
| 1 | Click on add project | New project window will open | New project windows opens | PASS |
| 2 | Enter customer name and nothing else | Error will be displayed, explaining what information is missing | Nothing happens | FAIL |

| Use case under test | Use case ”Manage team members” | | | |
| --- | --- | --- | --- | --- |
| Scenario | Main scenario | | | |
| Precondition | System should be running, and there has to be a project in the system to which team members can be assigned. | | | |
|  | | | | |
| Step | Action | Expected result/observation | Actual result/observation | Assessment(PASS/FAIL) |
| 1 | Search for specific project | Specific project found | No search function implemente. Manual search works | PASS? |
| 2 | Select the project and open it | Project information will be displayed | Project window opens and information is displayed | PASS |
| 3 | Click on add team member | New team member window will be displayed | New team member window opens | PASS |
| 4 | Enter name, choose role and click save | Team member window will close, project window will open and the team member is now displayed in the team member list | Project window opens. Team member is list in team member list with name and role | PASS |
| 5 | Select team member to edit and click edit team member | Team member window opens and displays information of selected team member | Team member window opens, with selected team member name, role and the members unique id | PASS |
| 6 | Choose a new role and click save | Team member window closes, project window opens and the team members role is changed to the newly selected role | Project window opens and the newly saved role is displayed in the team member list | PASS |
| 7 | Select team member to be removed and click remove | Confirmation box will appear and ask if you want to remove selected team member | Confirmation box appears | PASS |
| 8 | Choose Ok | Selected team member will now be removed from the projects team member list | Team member disappears from the team member list | PASS |

| Use case under test | Use case ”Manage team members” | | | |
| --- | --- | --- | --- | --- |
| Scenario | Exception scenario | | | |
| Precondition | System should be running, and there has to be a project in the system to which team members can be assigned. | | | |
|  | | | | |
| Step | Action | Expected result/observation | Actual result/observation | Assessment(PASS/FAIL) |
| 1 | Search for specific project | Specific project found | No search function implemente. Manual search works | PASS? |
| 2 | Select the project and open it | Project information will be displayed | Project window opens and information is displayed | PASS |
| 3 | Click on add team member | New team member window will be displayed | New team member window opens | PASS |
| 4 | Only enter name, do not select a role | An error will be listed with information about what went wrong | Nothing happens | FAIL |

| Use case under test | Use case ”Manage requirements” | | | |
| --- | --- | --- | --- | --- |
| Scenario | Main scenario | | | |
| Precondition | System should be running, and a project should already be created in the system. The project in question should have at least one team member assigned | | | |
|  | | | | |
| Step | Action | Expected result/observation | Actual result/observation | Assessment(PASS/FAIL) |
| 1 | Select the project and open it | Project information will be displayed | Project window opens and information is displayed | PASS |
| 2 | Click on add requirement | New requirement window will be displayed | New requirement window is opened | PASS |
| 3 | Enter estimated time, user story, deadline, priority, responsible team member and save the requirement | Requirement window will be update with entered information, and will be given a unique id | Requirement window is updated, but window title is still “New Requirement” | FAIL |
| 4 | Click on back and choose ok in confirmation box | Project window will open, the newly added requirement will be displayed in the requirement list. | Project window opens, and the requirement is listed | PASS |
| 5 | Select the requirement and click on requirement details | Requirement window will open and show information from selected requirement | Requirement window opens and displays information of selected requirement | PASS |
| 6 | Edit estimated time, user story, priority, responsible team member and save the requirement | Requirement window will be update with entered information | Requirement window is updated with edited information | PASS |
| 7 | Click on back and choose ok in confirmation box | Project window will open, the requirement will be displayed in the requirement list, with the edited information | Project window opens, and the requirement is listed with edited changes displayed | PASS |
| 8 | Select requirement and remove it | Confirmation box will appear and ask if you want to remove selected requirement | Confirmation box appears | PASS |
| 9 | Choose Ok | Selected requirement will now be removed from the projects requirement list | Requirement is removed from the list | PASS |

| Use case under test | Use case ”Manage requirements” | | | |
| --- | --- | --- | --- | --- |
| Scenario | Exception scenario | | | |
| Precondition | System should be running, and a project should already be created in the system, with at least one requirement. The project in question should have at least one team member assigned | | | |
|  | | | | |
| Step | Action | Expected result/observation | Actual result/observation | Assessment(PASS/FAIL) |
| 1 | Select the project and open it | Project information will be displayed | Project window opens and information is displayed | PASS |
| 2 | Select a requirement and click on details | Requirement window will be displayed with information from selected requirement | Requirement window is opened and displays information about selected requirement | PASS |
| 3 | Enter characters into the estimate time box | Information will not be saved and an error text will be displayed with information about what is wrong | No information is save, error text is displayed, but it does not explain in an understandable way what went wrong | FAIL |

| Use case under test | Use case ”Update tasks” | | | |
| --- | --- | --- | --- | --- |
| Scenario | Main scenario | | | |
| Precondition | System should be running, and a project with one requirement which contains one task, should already be created in the system. | | | |
|  | | | | |
| Step | Action | Expected result/observation | Actual result/observation | Assessment(PASS/FAIL) |
| 1 | Search | Not implemented | Not implemented | FAIL |
| 2 | Select the project and open it. | Project information will be displayed | Project window opens and information is displayed | PASS |
| 3 | Note down total time spent | Note taken | Took note | -- |
| 4 | Select a requirement and click on details | Requirement window will be displayed with information from selected requirement | Requirement window is opened and displays information about selected requirement | PASS |
| 5 | Select a task and click on open | Task window will open and display information about selected task | Task window opens and displays information about selected task | PASS |
| 6 | Enter hours worked and save | Hours worked is now updated to include entered hours | Hours worked updated to include entered hours | PASS |
| 7 | Click on home, choose ok in confirmation box | Home window will be displayed | Home window is displayed | PASS |
| 8 | Select the project and open it. | Project information will be displayed | Project window opens and information is displayed | PASS |
| 9 | Read project information | Project information about time spent is updated with time entered earlier | Time spent is now increased by amount of hours entered earlier | PASS |

| Use case under test | Use case ”Update tasks” | | | |
| --- | --- | --- | --- | --- |
| Scenario | Exception scenario | | | |
| Precondition | System should be running, and a project with one requirement which contains one task, should already be created in the system. | | | |
|  | | | | |
| Step | Action | Expected result/observation | Actual result/observation | Assessment(PASS/FAIL) |
| 1 | Search | Not implemented | Not implemented | FAIL |
| 2 | Select the project and open it. | Project information will be displayed | Project window opens and information is displayed | PASS |
| 3 | Select a requirement and click on details | Requirement window will be displayed with information from selected requirement | Requirement window is opened and displays information about selected requirement | PASS |
| 4 | Select a task and click on open | Task window will open and display information about selected task | Task window opens and displays information about selected task | PASS |
| 5 | Enter 100 hours in the estimate field | Error will be displayed telling you that it is not possible to enter such a large amount of hours as an estimate | Error is displayed telling me that the current entered estimated time exceed estimated time for related requirement | PASS |

| Use case under test | Use case ”View project information” | | | |
| --- | --- | --- | --- | --- |
| Scenario | Main scenario | | | |
| Precondition | There should already exist a project with information in the system | | | |
|  | | | | |
| Step | Action | Expected result/observation | Actual result/observation | Assessment(PASS/FAIL) |
| 1 | Enter website | Website opens | No website | FAIL |
| 2 | Search for project | Project is found | - | | - | - | | - |
| 3 | Open project | Project information is displayed | - | | - | - | | - |
| 4 | Read information | Reader will now be informed about selected project | - | | - | - | | - |

| Use case under test | Use case ”Manage tasks” | | | |
| --- | --- | --- | --- | --- |
| Scenario | Main scenario | | | |
| Precondition | System should be running, and a project with at least one requirement should already be created in the system. | | | |
|  | | | | |
| Step | Action | Expected result/observation | Actual result/observation | Assessment(PASS/FAIL) |
| 1 | Search for the specific requirement and open it  \*\*if new project - add tasks\*\* | The list of tasks is displayed  \*\*Tasks have been added to the list\*\* | Task list is displayed when selecting and opening the requirement  \*\*No search function is implemented, need to search manually | PASS |
| 2 | Select and open task | The task information is displayed | The task information is displayed | PASS |
| 3 | Update the task | The possibility to update the status of the task to: “Started”, “Not-Started” or  “Finished”  \*\*when all tasks are finished, the requirement automatically gets marked as “Ended” | The dropdown menu with a possibility to change the task status to the mentioned states is there  \*\*requirement gets updated to “Ended” when tasks are finished | PASS |
| 4 | Choose team member for a task | The team member gets assigned to a task | The dropdown menu to choose from is implemented and working | PASS |
| 5 | Save changes | The previously made changes get saved and updated | The previously made changes get saved and updated | PASS |
| 6 |  |  |  |  |

| Use case under test | Use case ”Manage tasks” | | | |
| --- | --- | --- | --- | --- |
| Scenario | Exception scenario | | | |
| Precondition | System should be running, and a project with at least one requirement should already be created in the system. | | | |
|  | | | | |
| Step | Action | Expected result/observation | Actual result/observation | Assessment(PASS/FAIL) |
| 1 | Enter integers instead of String into the “Title” field | Should save the title with integers with it in case if tasks are numbered | Saves the title with integers | PASS |
| 2 | Entering the date past the deadline of the requirement | Gives an error saying that the date is way past to that of requirement | Does not give any errors and saves the date which is chosen | FAIL |
| 3 | Entering String into the field of “Hours spent” | Gives an error saying that the entered information should be in integer form | Does not do anything | FAIL |
| 4 | Entering the String into the field of “Estimate” | Gives an error saying that the input is not of correct type | Gives an error if the String is entered | PASS |

# Appendix G. Old GUI sketch

## 

# Appendix H. New GUI sketch

# 

# 

# Appendix I. Github repository

<https://github.com/chooseanother/G4_SEP1Z_A20>

# Appendix J. Group contract

Group Contract

***Group 4***

**Members:**

Aleksandrs Bistrovs **304542**

Henrik Koster **305916**

Kim Dahl Tranberg **172394**

Laurentiu Mihai **304456**

The following document has been developed and includes the code of conduct and cooperation between the team members of the group, where each member has to agree for the following terms and conditions as stated below:

**Participation**:

Strive for the best results, squeeze out maximum potential!

Invest the time, 2-3 hours each day - including the weekends.

Attend all the lectures, so everyone is on board.

Respect the team and their work, show interest for the project.

Try to divide the work equally.

**Communication**:

Talk about all the problems that arise during the project, try to find a solution between the members before calling in for help from the outside environment - supervisors.

Ask questions in case of doubt, make suggestions if there are any worthy ideas.

Give feedback to the team and be able to take constructive criticism from the peers.

**Meetings**:

Be serious and respectful towards the group members by showing up on the previously agreed meetings. Always come prepared and on time. In case of not being able to show up, always notify the team. Do not skip the meetings without a valid reason.

**Conflicts**:

When disagreements arise the team has agreed to solve them by a voting system. It means that the ideas which are backed up by the majority of the group are the ones which are taking place.

In case of any major conflicts or try to solve the problems by communicating within the group first. If that does not work - try to find a solution with the help of the supervisors.

**Deadlines**:

Respect the previously agreed time schedule and follow the deadlines. Try not to postpone the work until the last moment, thus making it sloppy and imprecise.

***Signatures: Date: 11.10.2020***

# Appendix K. Project report section authors

| Abstract | Shared |
| --- | --- |
| Introduction | Aleksandrs |
| Analysis | Henrik |
| Requirements | Shared |
| Use case diagram | Henrik |
| Use case description | Henrik |
| Activity diagram | Henrik |
| Domain model | Aleksandrs |
| Design | Aleksandrs |
| Class diagram | Aleksandrs |
| Sequence diagram | Kim |
| Gui design | Aleksandrs |
| Website | Aleksandrs |
| Implementation | Kim |
| Test | Aleksandrs |
| Junit tests | Aleksandrs |
| Use case test | Henrik |
| Results and discussion | Shared |
| Conclusion | Shared |
| Project future | Shared |