## Functional requirements

**Critical priority:**

1. As a project creator, I want to be able to create new projects in the system, so that our projects will become easier to manage.
2. As a project creator, I want to be able to assign team members to a project, such that they can work on projects.
3. As a product owner, I want to be able to add and remove requirements, to make sure the project will meet the customers specifications.
4. As a scrum master, I want to be able to add tasks to the project, such that the rest of the team members know what tasks they have to work on.
5. As a project creator, I want all information from the system stored in a single file, so that information will not be lost when the system is closed.

**High priority:**

1. As a project creator, I want to be able to assign roles to team members, so it will be clear what kind of responsibility they have.
2. As a product owner, I want to be able to prioritize requirements based on their importance(Low, Medium, High, Critical), such that features which are critical for the system to function will be developed first.
3. As a product owner, I want to change who is responsible for a requirement in case another team member is more suited for handling that requirement.
4. As a product owner I want requirements to automatically get marked with “Ended” when all tasks for a requirement are done, so that I can see what requirements I should be testing.
5. As a scrum master, I want change who is responsible for a task in case another team member is more suited for handling that task.
6. As a product owner, I want requirements to contain an id, user stories in who, what, why template, estimated time, a deadline, who is responsible, status, total hours spent, such that I can easily get an overview of all relevant information for a requirement.
7. As a product owner, I want to be able to approve or reject requirements, such that the project can reach a finished state, and make sure it meets the customers needs.
8. As a scrum master, I want each task to contain all information (Requirement ID, task ID, title, time estimation, deadline, responsible team member, status, hours spent) such that I can easily get an overview of all relevant information for a task.
9. As a team member, I want to be able to register a total amount of hours to the system whenever a task has been finished, so that I can keep track of its progress and see our productivity in regards to how well we can estimate time for tasks.
10. As a product owner, I want requirements to display total time spent, which should automatically be a total of time spent from all related tasks, so it will be possible to track the progress of a requirement*.*
11. As a product owner, I want the total estimated time of all tasks related to a specific requirement, be exactly the same amount as the estimated time for that requirement, such that the project will stay on schedule.
12. As a scrum master, I want to mark the status of each task in the form of: started, not started, finished; such that team members know which tasks should be worked on.

**Low priority:**

1. As a team member, I want to be able to search information regarding the projects by ID, responsible team members, deadlines, such that I can easily access specific information of a given topic, without having to go through the entire system.
2. As a project creator, I want to have the ability to change the roles of the team members, so they can work on more suitable tasks.
3. As a customer, I want the project’s description, requirements and their status, displayed on a website, such that I can track its progress.
4. As a project creator, I want to be able to remove team members from a project, in case the team size needs to be adjusted.

## Non-functional requirements

1. The GUI should be implemented using Java/JavaFx.
2. The website has to function using Google Chrome (version 86.0.4240.193, release date 2020-11-10) Mozilla Firefox (version Firefox 82, release date 2020-10-20) Safari (version Safari 14.0, release date 2020-9-16).

## 

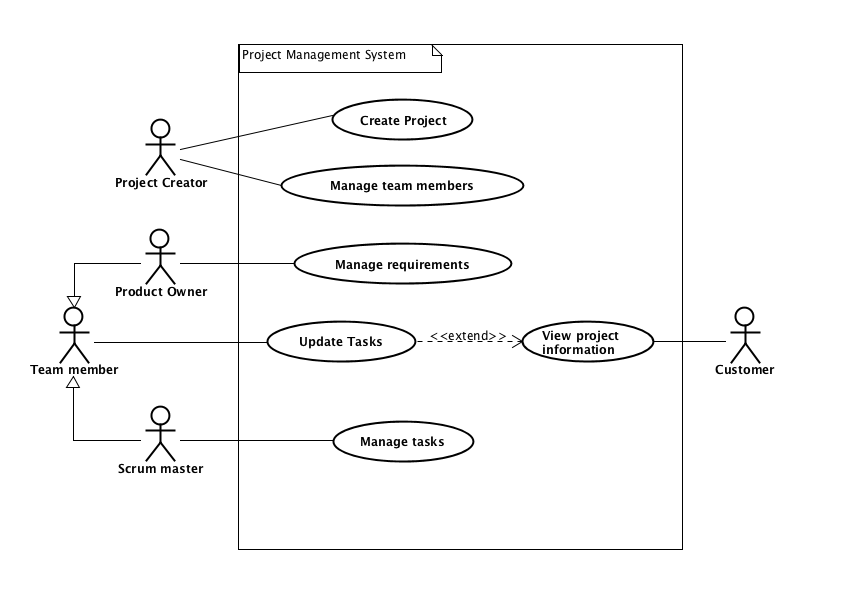


Fig. 1. Project management system - Use case diagram.

| **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. |

## 

| **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) |

## 

Fig. 2. Create project - Activity diagram.

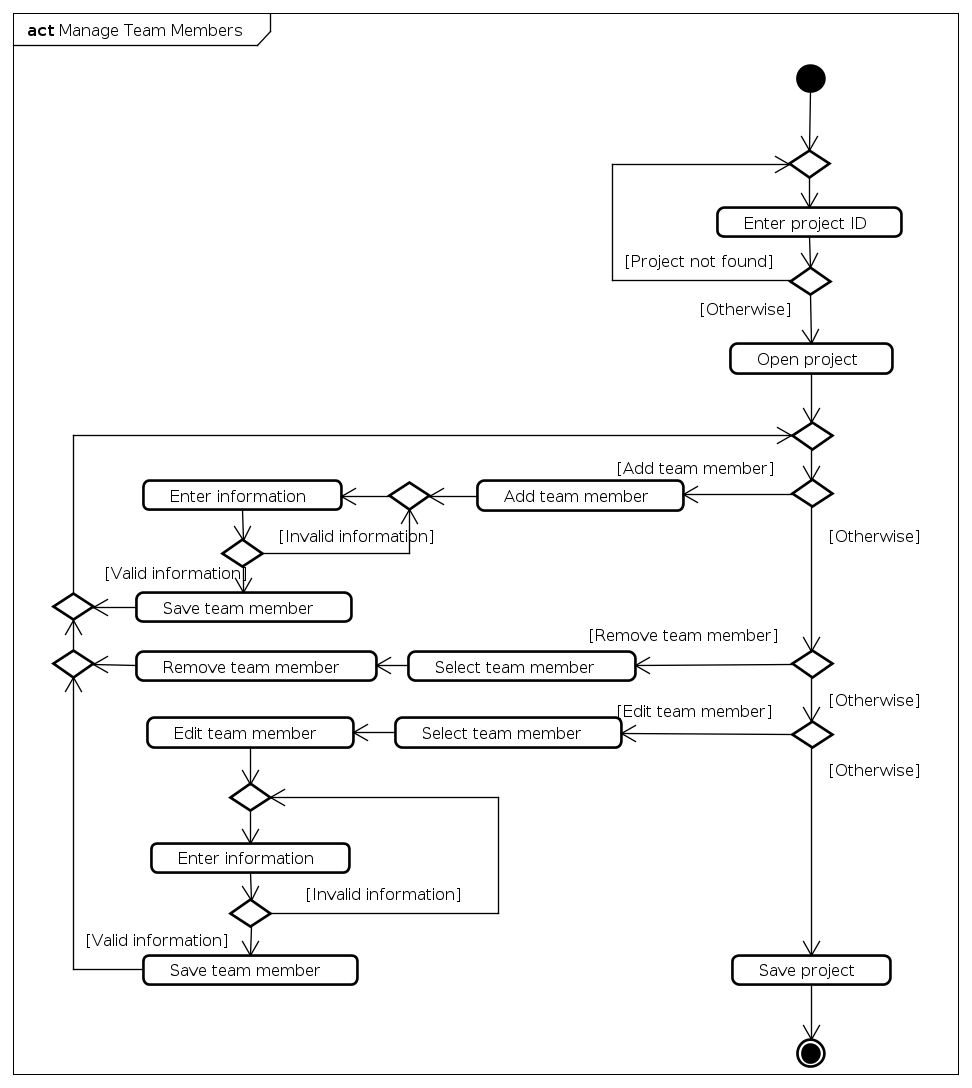


Fig. 3. Manage team members - Activity diagram.

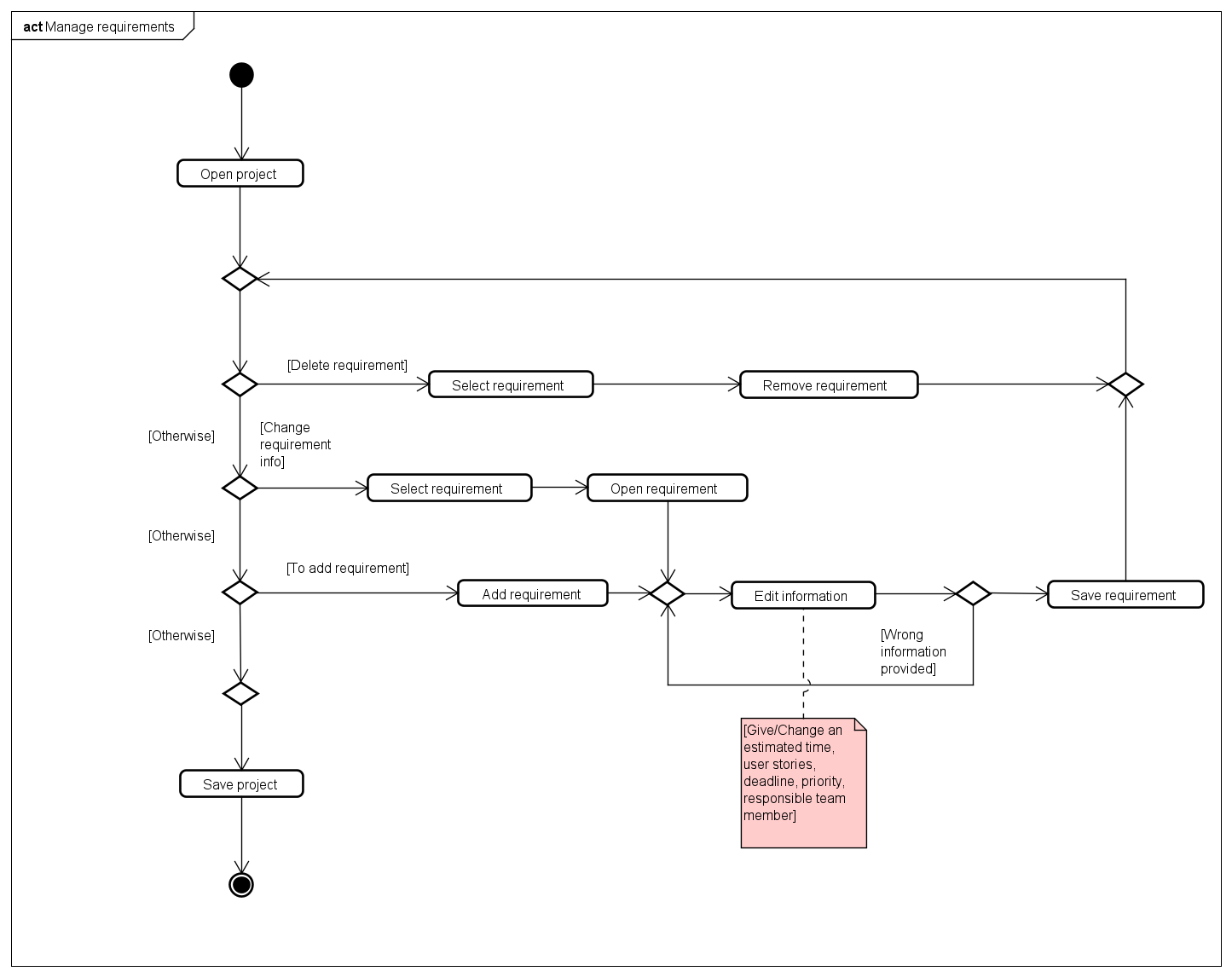


Fig. 4. Manage requirements 2nd version - Activity diagram.

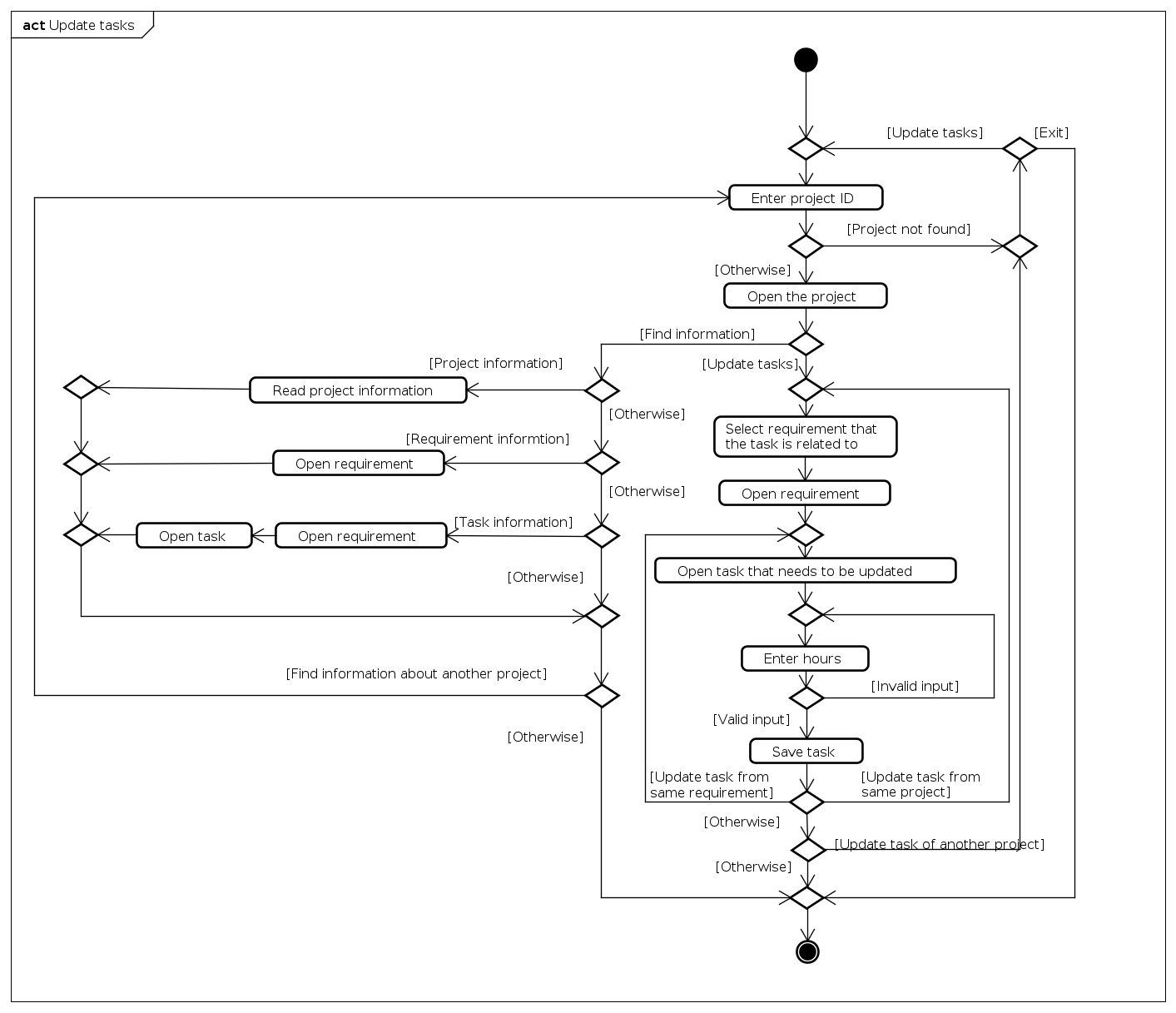


Fig. 5. Update tasks - Activity diagram.

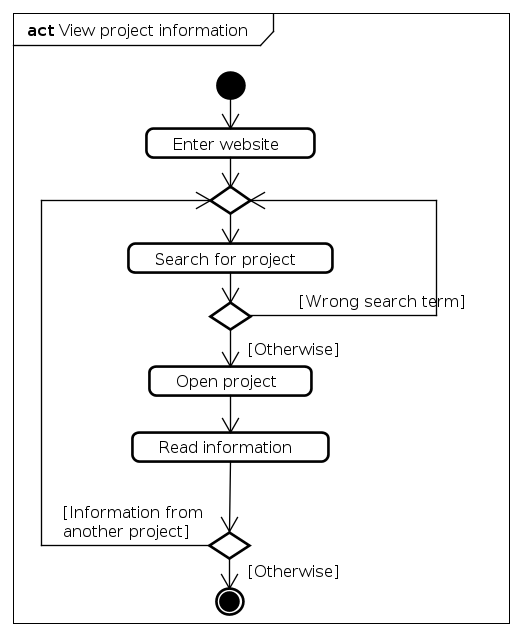


Fig. 6. View project information - Activity diagram.

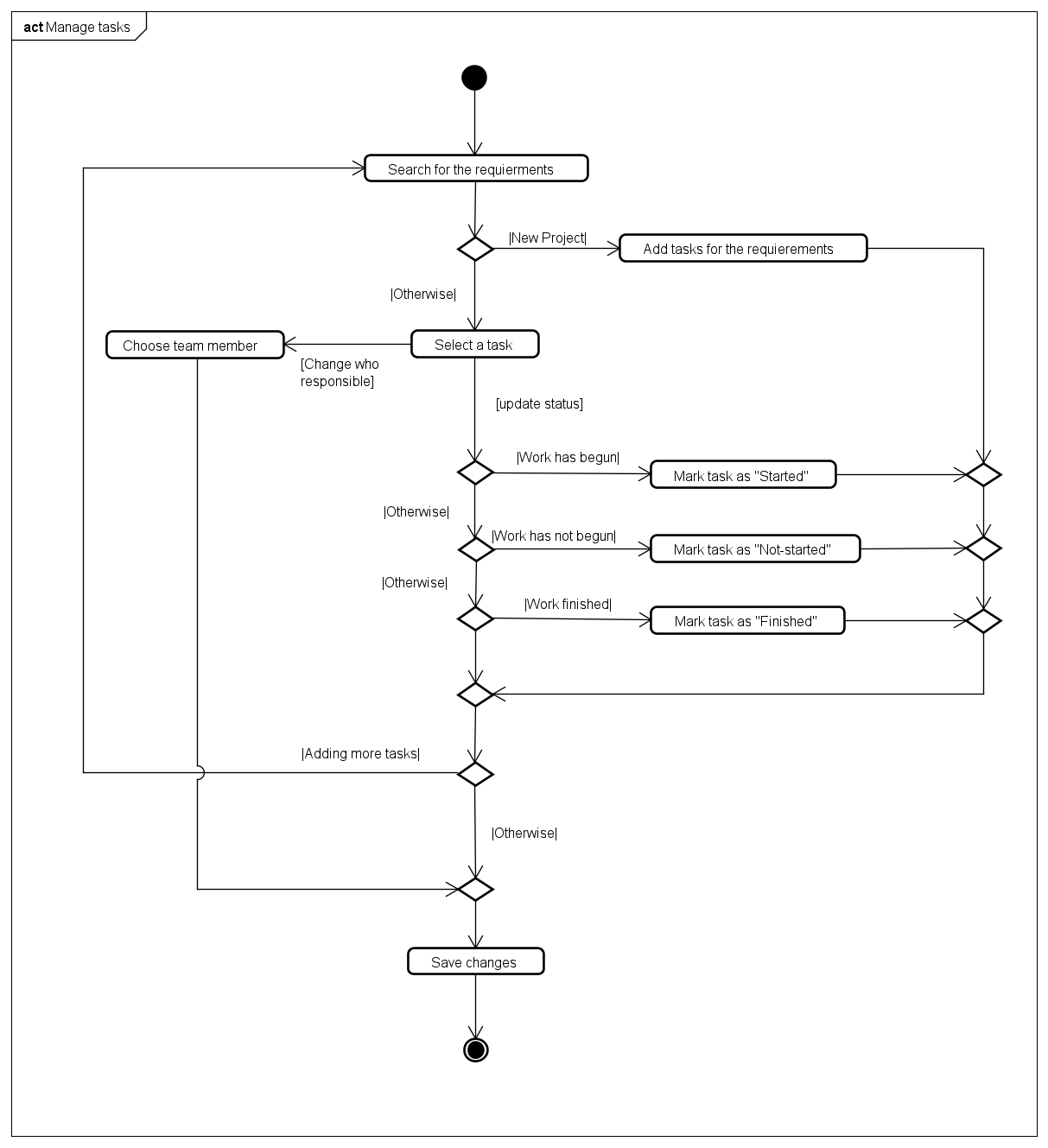


Fig. 7. Manage tasks - Activity diagram.

## 

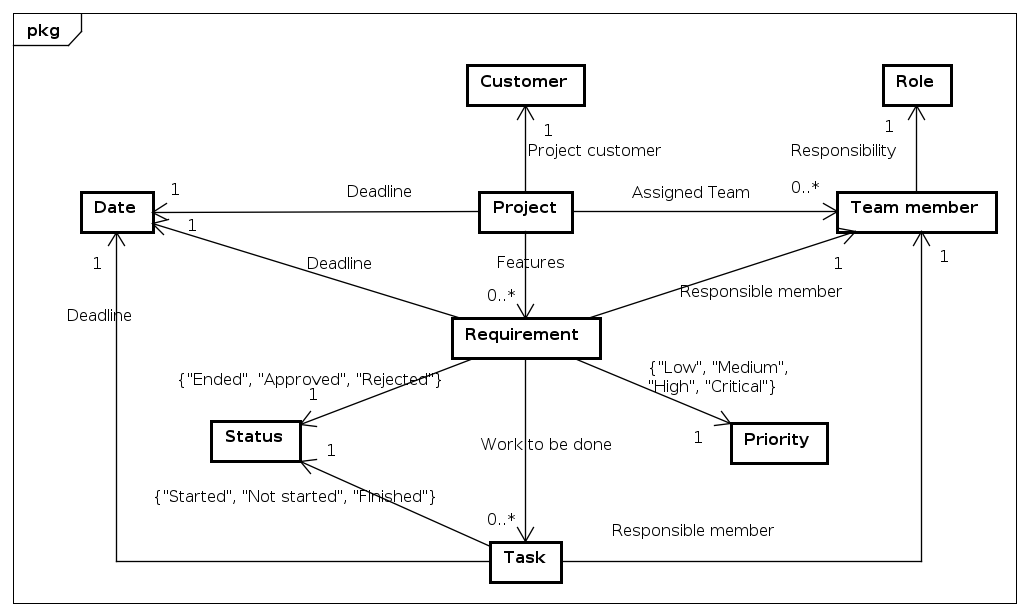


Fig. 8. Project management system - Domain model.