Contents

[Use case model 2](#_Toc20988468)

[Use case diagram 2](#_Toc20988469)

[Use case description 3](#_Toc20988470)

[Projet CRUD 3](#_Toc20988471)

[Start sprint 4](#_Toc20988472)

[Task CRUD 5](#_Toc20988473)

[Change task state 8](#_Toc20988474)

[Close sprint 9](#_Toc20988475)

[Close project 9](#_Toc20988476)

[Glossary 10](#_Toc20988477)

# Use case model

## Use case diagram



## Use case description

### Project CRUD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case Number: 1** | | | | |
| **Name** | | Project CRUD | | |
| **Brief description** | | Users creates the projects | | |
| **Actors** | | Scrum Masters or team members | | |
|  | | | | |
| **Preconditions** | | | | |
| System users must be logged in to the system | | | | |
| **Flows of events:** | | | | |
| **1.0** | **Create project** | | | |
| **Step** | **User Actions** | | | **System Actions** |
| 1 | Choose create project command | | | Shows the form that must filled to create project |
| 2 | The user fills the necessary fields   * Title * Description * Sprint duration * Start date * Due date * //Team members | | | Validate input format while filling out the form |
| 3 | Add project members | | |  |
| 3.1 | Choose select team members command | | | Shows list of registered system users |
| 3.2 | Filter and choose system user | | | Validates chosen system users |
| 4 | The user sends save project request to the system | | | The system saves the project and returns a success message or a fail message in case of an exception |
| **Post Condition** | | | | |
| The new project persisted in the system. | | | | |
| **Business Rule** | | | | |
| Any team members can create projects | | | | |
|  | | | | |
| **1.1** | **Update project** | | | |
| 1 | The user selects project update command | | The system displays back an editable form with the project details pre-filled. | |
| 2 | The user updates parts that they want to update and request system to save the project | | The system updates the record and returns a success message or a fail message on exception. | |
| **Postconditions** | | | | |
| The project will be updated | | | | |
| **Business rule** | | | | |
| Updated field must be notified | | | | |
| **1.2** | **Archive project** | | | |
| 1 | The user selects project archive command | | The system displays back confirmation form | |
| 2 | The user selects confirm to delete project | | The system validates to confirm the task is not assigned to any user and archives the task and returns a result message. | |
| **Postconditions** | | | | |
| The project will be deleted | | | | |
| **Business rule** | | | | |
|  | | | | |

### Start sprint

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 2** | | | |
| **Name** | | Start sprint | |
| This use case allows the Scrum Master to start a Sprint for project | | | |
| **Actors** | | Scrum Master | |
| **Pre conditions** | | | |
| The Scrum Master must be logged into the system. There must be no active Sprint. Tasks must be added to Backlog | | | |
| **Flows of Events:** | | | |
| **2. Basic Flows** | | | |
| **2.1 Start a new sprint** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | Scrum Master to select the tasks which should be done by upcoming sprint | | Once Scrum Master selects the tasks which should be done by upcoming Spring, from backlog, where all tasks are selective, system shows “Start them as a new sprint” functionality will be available. |
| 2 | Scrum Master to call “Start them as a new sprint” functionality | | After Scrum Master called the Sprint start function, The system updates the record and returns the success message or a fail message on exception. |
| **Post conditions** | | | |
| The new sprint to board and locked for changing Sprint total effort. | | | |
|  | | | |
| **Business Rules** | | | |
| No duplicate active sprint. A unique sprint ID is generated by static variable. | | | |
| **1.2 Read/view Sprint** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | Scrum Master or other users to view Sprint | | System to show the tasks in the Sprint |
| 2 | Scrum Master or other users to see the Sprint they want to watch | | System to show previously done and active sprint list |
| 3 | Scrum Master or other users to select the Sprint they want to watch | | System to show the tasks in the selected sprint |

### Task CRUD

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case number: 3** | | | |
| **Name** | | Task CRUD | |
| **Brief description** | | Scrum master and all team members can add task. | |
| **Actors** | | Team member |  |
| **Preconditions** | | | |
| Scrum master or member must be logged in to the system. | | | |
| **Flows of events:** | |  |  |
| **Basic Flows** | |  |  |
| **3.0** | **Create Block** | | |
| **Steps** | **User actions** | **System actions** | |
| 1 | Choose project board. | Show project detail information. -Project backlog -Project sprint list | |
| 2 | The user chooses create task command | System displays the task form with the fields for -task name -description -priority - development effort point -testing effort point -developer -tester -due date -start date *-duration -created user -created date* | |
| 3 | The user fills out the form and requests the system to save the details | The system saves the task and returns a success message or a fail message in case of an exception. | |
| **Postconditions** | | | |
| Task is saved to the database. | | | |
| **Business Rule** | | | |
| Assigned developer and tester cannot be same user. | | | |
| **3.1** | **Read/View task** | | |
| **Steps** | **User actions** | **System actions** | |
| 1 | The user select sprint. | System show list of tasks assigned to the chosen sprint by kanban view. Kanban view is organized according to task states, which are: -New -Developing -To be tested -Testing -Approved -Deployed | |
| 2 | The user selects to view a task they want | The system return the task object as a string with task detail and related comments in task detail screen. | |
| **3.2** | **Update task** | | |
| **Steps** | **User actions** | **System actions** | |
| 1 | The user select sprint. | System returns list of tasks in kanban view. | |
| 2 | The user selects the task they want to update. | The system displays back an editable form with the task details pre-filled. | |
| 3 | The user updates the parts they want to update and request system to save the new details. | The system updates the record and returns a success message or a fail message on exception. | |
| **Postconditions** | | | |
| The task will be updated. | | | |
| **Business Rule** | | | |
| ?? | | | |
| **3.3** | **Archive task** | | |
| **Steps** | **User actions** | **System actions** | |
| 1 | The user select sprint. | System show list of tasks assigned to the chosen sprint by kanban view. Kanban view is organized according to task states, which are: -New -Developing -To be tested -Testing -Approved -Deployed | |
| 2 | The user selects to delete a task they want. | They system displays a confirmation dialogue window. | |
| 3 | The user selects OK on the confirmation dialog window to confirm deleting the task. | The system checks to confirm the task is not assigned to any user and archives the task and returns a result message. | |
| **Postconditions** | | | |
| The task will be archived. Which won’t show in sprint or backlog. | | | |
| **Business Rule** | | | |
| The task must have not been assigned to any other user. | | | |
| Scrum master can archive task which is assigned to user. | | | |
| User can archive task only he/she created. | | | |
| **1.3** | **Delete task** | | |
| **Steps** | **User actions** | **System actions** | |
| 1 | The scrum master goes to archived task list. | System show list of archived tasks with task name and sprint number. | |
| 2 | The scrum selects to delete a task they want | They system displays a confirmation dialogue window. | |
| 3 | The user selects OK on the confirmation dialog window to confirm deleting the task. | The system deletes the task and returns a result message. | |
| **Postconditions** | | | |
| The task will be deleted. | | | |
| **Business Rule** | | | |
| The task must have not been assigned to any other user. | | | |
| Scrum master can archive task which is assigned to user. | | | |
| User can archive task only he/she created. | | | |

### Change task state

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use case number: 4** | | | | |
| **Name** | | Change task state | | |
| **Brief description** | | Scrum master can change status of any task. Developer and tester can change their assigned task status. | | |
| **Actors** | | Developer |  | |
| **Preconditions** | | | | |
| Scrum master or member must logged in to the system.  Task is assigned to the current user. | | | | |
| **Flows of events:** | | | | |
| **Basic Flows** | | | | |
| **4.0** | **Change development task state** | | | |
| **Steps** | **User actions** | | | **System actions** |
| 1 | Choose task to be changed state | | | Select the task |
| 2 | Change the status of the task to DEVELOPING | | | System  ask finish date  update task information and  save state change log. |
| 3 | The user call save command | | | The system saves the task and returns a success message or a fail message in case of an exception. |
| **Preconditions** | | | | |
| The task is assigned to the developer | | | | |
| **Postconditions** | | | | |
| Task status is updated. Change log is created for the task | | | | |
| **Business Rule** | | | | |
| 1. Developer can change status from NEW to DEVELOPING, or DEVELOPING to TO BE TESTED, DEVELOPING to NEW TO BE TESTED to DEVELOPING/NEW 2. Tester can change task status from TO BE TESTER to APPROVED, APPROVED to TO BE TESTED | | | | |

### Close sprint

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 5** | | | |
| **Name** | | Close sprint | |
| This use case allows the Scrum Master to close a active sprint for project | | | |
| **Pre conditions** | | | |
| The Scrum Master must be logged into the system. There must be an active Sprint. | | | |
| **Flows of Events:** | | | |
| **5. Basic Flows** | | | |
| **5.1 Close sprint** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | Scrum Master to call the functionality “Close current sprint” | | System to display confirmation message to Scrum Master |
| 2 | Scrum Master to confirm sprint closure | | The system updates the record and returns the success message or a fail message on exception. |
| **Post conditions** | | | |
| Not closed tasks will be moved to backlog | | | |
|  | | | |
| **Business Rules** | | | |
|  | | | |
|  | | | |
| **2.2 Add retrospective meeting note to recently closed Sprint** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | Scrum Master to choose recently closed sprint | | System to display the tasks in the selected sprint |
| 2 | Scrum Master to choose “Add retrospective meeting note” | | System to display a field to be populated by text |
| 3 | Scrum Master to add and call the command to add note to the sprint | | The system updates the record and returns the success message or a fail message on exception. |

### Close project

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 6** | | | |
| **Name** | | Close Project | |
| **Brief description** | | Users closes the projects | |
| **Actors** | | Scrum Master | |
|  | | | |
| **Preconditions** | | | |
| Scrum master must be logged in to the system | | | |
| **Flows of events:** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | Choose close project command | | Validates the project and if the project is able to close then shows form that will get closing comment or return fail message in case of an exception |
| 2 | IF(Project can be closed)   * Scrum master writes closing comment and send close request to the system   ELSE   * Shows fail message and ends UC | | Validating entered text and saves text and returns a success message or a fail message in case of an exception |
| **Post Condition** | | | |
| The project status changes to closed | | | |
| **Business Rule** | | | |
| 1. All sprints must be closed  2. Backlog must be empty | | | |

# Glossary

1. **Project** - an individual or collaborative enterprise that is carefully planned and designed to achieve a particular aim
2. **Scrum Master** - responsible for promoting and supporting Scrum. Scrum Masters do this by helping everyone understand Scrum theory, practices, rules, and values.
3. **Tester** - a person who tests something, especially a new product.
4. **State –** Tasks can exist in one of described states
5. **Effective planning report –** The report that shows how good is our plan
6. **Sprint duration –** one sprint’s timeline
7. **Archive task –** it will soft deletes task from project
8. **Scrum** - Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.
9. **Developer -** A developer is an individual that builds and create software and applications.
10. **Task state** - Every task must be in one of the states: “New”, “Developing”, “To be tested”, “Testing”, “Approved” and “Deployed”
11. **Burn Down chart -** The burn down is a chart that shows how quickly you and your team are burning through the tasks. It shows the total effort against the amount of work we deliver each iteration.
12. **Task -** A piece of work to be done or undertaken. Tasks are used to be extracted from user stories
13. **User Story -** A user story is a very high-level definition of a requirement, containing just enough information so that the developers and tester can produce a reasonable estimate of the effort to implement it
14. **Agile SWD method -** It is a type of project management process, mainly used for software development, where demands and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customers.