Software Requirements Specification

for

Web-based project management system

**Prepared by**

**Brazil Analysis Team**

**Gabriel Oliveira** [gabriel.pa.oliveira@gmail.com](mailto:gabriel.pa.oliveira@gmail.com) **Analyst**

**Lauriane Moraes** [lauriane.cpm@gmail.com](mailto:lauriane.cpm@gmail.com) **Analyst**

**Adalto Sparremberguer** [tec.adalto@gmail.com](mailto:tec.adalto@gmail.com) **Designer**

**University:** PUC-RS

**Country:** Brazil

**Oct 20th, 2014.**

**Contents**

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Overall Description 5

2.1 Product perspective 5

2.2 Product functions 5

2.3 Constraints 5

2.4 Assumptions and dependencies 5

3. Specific Requirements 6

3.1 Login 6

3.2 Project 10

3.3 Iteration 16

3.4 Manage Team Members 20

3.5 Work item 25

3.6 Home Dashboard 35

3.7 Screen Diagram 40

3.8 Use Case Diagram 41

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author(s)** |
| October 20th | 1.0 | Fist version | Brazil Analysis Team |
| October 24th | 2.0 | Second version | Brazil Analysis Team |
|  |  |  |  |

# 

# 1. Introduction

This document is the Software Requirements Specification (SRS) for a web-based project management system to support distributed software development.

## 1.1 Purpose

This document should fully describe the external behaviour of the application or subsystem identified. It also describes non-functional requirements, design constraints and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

## 1.2 Scope

This document explains all the user interactions (user cases, UC) with the system and the database. Screen mockups and the relationship of them with UCs are outside the scope of this document.

## 1.3 Definitions, Acronyms and Abbreviations

WI Work item

UC Use Case

User Owner and Team member

## 1.4 References

This document was used as reference to build the screen mockups and the diagram showing the relationship of screens and UCs.

## 1.5 Overview

Chapter 2 describes the system in a high-level vision. Chapter 3 explains each Use Case

# 2. Overall Description

This section describes the general factors that affect the product and its requirements.

## 2.1 Product perspective

The web-based project management system need to supports:

* A group of developers working together, in one or more locations;
* Project development through successive iterations;
* Project progress tracking.

## 2.2 Product functions

High level description of all the functionalities are described below:

* **User management:** users are able to register within the system, delete its own account and change personal details. Those functions are described on Chapter 3.1;
* **Project management:** users are able to create and delete projects, as well as change the name of the Project he has ownership and navigate to other screens in order to better track the project progress. These functions are accessible through a “Project” tab, and are described in Chapter 3.2;
* **Iterations management:** users are able to create and delete Iterations inside a Project. Each project is created with a special “Backlog” iteration, that can’t be deleted. These functions are described on Chapter 3.3;
* **Members management:** user can invite other users to their projects, and project owners can promote users. Users can also see each other points on this screen. These functions are accessible through a “Project” tab, and are described in Chapter 3.4;
* **Work Items management:** inside each Iteration the user can create, delete and update work items. Each Work Item can have comments and links to other Work Items. The user can also list all the Work Items inside an Iteration, as well as filter and order them. These functions are described on Chapter 3.5;
* **Dashboard:** each user can see the Work Items he owns when accessing the “Home” tab, as well as links to other users Dashboard. These functions are described on Chapter 3.6. Global functions, like Global Work Item Search, are also described on Chapter 3.6.

## 2.3 Constraints

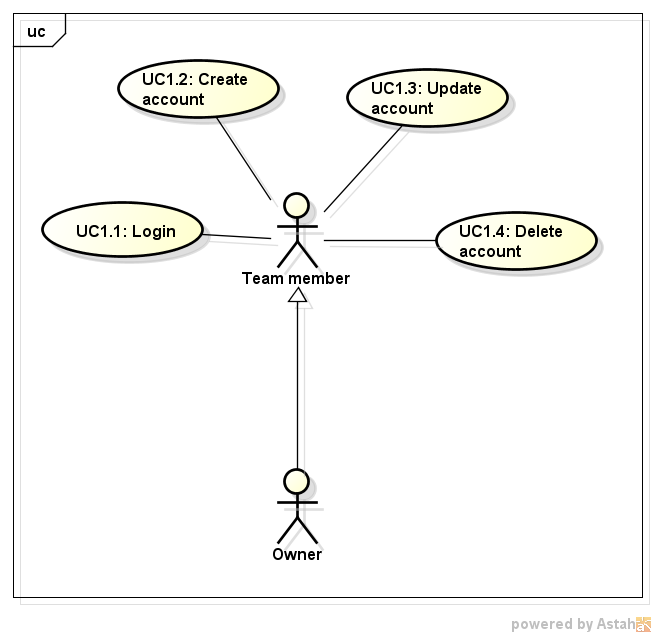
The same email can’t be used for multiple accounts.

## 2.4 Assumptions and dependencies

The system depends on a Database, where all information are accessible and persistent stored, and an Email service, to send system notifications.

# 3. Specific Requirements

## 3.1 Login



A “user” shall have a name (mandatory), a surname (mandatory), an email (mandatory), a gender (mandatory), a photo (optional) and a role (optional). The system shall thrown an error if the same Email is used to register more than 1 user. The password shall have 8 characters long and has a mixed of letters and numbers only. Optionally, it should have a Photo and the Role, if there is no photo uploaded the system should display an “avatar” matching with Gender inputted.

### UC1.1 - Login

|  |  |
| --- | --- |
| **UC description - Login** | |
| **Objective** | The user needs to access an application account. |
| **Priority** | High |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | N/A. |
| **Post Conditions** | The user accessed her/his account. |
| **Basic Flow** | 1. User goes to application page; 2. System loads a main screen; 3. System shows a popup, with the following fields: email and password, there is a “Login” button and “Sign Up” and “Forgot email/password” links; 4. User enters a email and password, so the user clicks on “Sign In” button; 5. System validates the data; 6. System loads the account. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 4, the User clicks on “Sign Up”**   1. Goes to UC1.2 on step 3.   **Alternative Flow 2 - At Step 4, the User clicks on “Forgot email/password”**   1. System shows a popup, with an email field to send a confirmation to stored email, the popup displays “Send email” and “Cancel” buttons; 2. User enters an email, so the user clicks on “Send email”; 3. A successful message is displayed to the user; 4. System sends an email to stored email in the application; 5. User goes to her/his personal email account and the user get her/his email and random password received; 6. User goes back to the application and the user enters a email and random password received; 7. System shows a popup, with a new password and confirmation password field; 8. User enter a new password and confirmation password, so the user clicks on “Done”; 9. System adds the new password to the database; 10. Return to Step 6.   **Alternative Flow 3 - At Step 1.1.5.5, if the informed email or password does not match the email and password stored**   1. An error message is shown to the user; 2. Return to Step 3. |
| **Exception Flow** | **Exception Flow 1 - At Step 4, the User doesn’t have permission**   1. An error message is shown to the user; 2. Return to step 2. |
| **Notes/Issues** |  |
| **Screen Flow** | |
| Screen_Login | | |

### UC1.2 - Create account

|  |  |
| --- | --- |
| **UC description - Create account** | |
| **Objective** | The user needs to get access from the application. |
| **Priority** | High |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | N/A. |
| **Post Conditions** | The account is created. |
| **Basic Flow** | 1. User could receive an invite via email; 2. User clicks on "Sign Up"; 3. System loads Users registration screen with the following fields: name (mandatory), surname (mandatory), email (mandatory), gender (mandatory), photo (optional), and role (optional) and a “Clear”, “Cancel”, and “Done” button; 4. User enters her/his personal data, so the user clicks on "Done". 5. System validates the fields; 6. System adds the account to the database. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 1, the new User receive an invite via email**   1. User clicks on a received link in her/his personal email; 2. Return to Step 3.   **Alternative Flow 2 - At Step 4, the User clicks on “Clear”**   1. System should clear all filled fields; 2. Return to Step 3.   **Alternative Flow 3 - At Step 4, the User clicks on “Cancel”**   1. Return to Step 2.   **Alternative Flow 4 - At Step 5, if the User enters invalid data**   1. An error message is shown to the user; 2. Return to Step 6. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

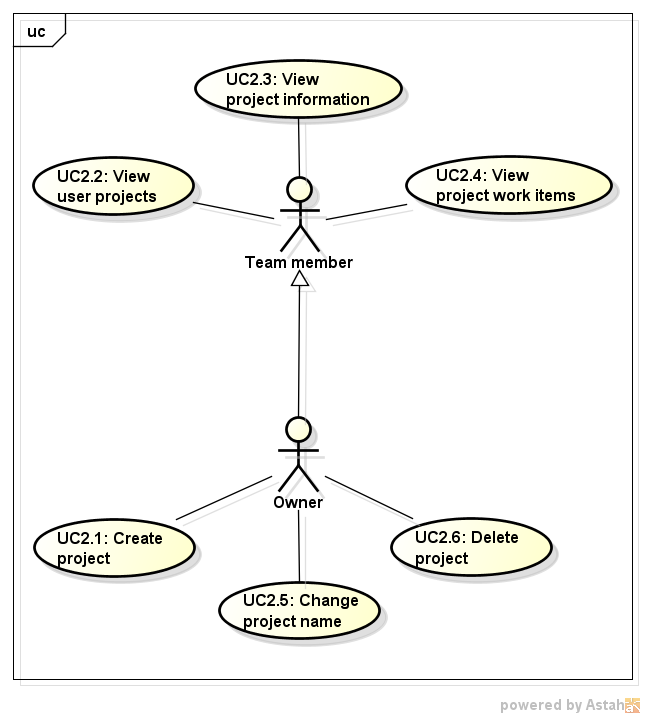
### UC1.3 - Update account

|  |  |
| --- | --- |
| **UC description - Update account** | |
| **Objective** | The user can edit all personal account data. |
| **Priority** | High |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and the user should be in her/his account configuration section. |
| **Post Conditions** | The data is updated and the “Home” tab Dashboard is shown. |
| **Basic Flow** | 1. User goes to Configuration icon and User edits the information he/she wants on the following fields: “name”, “surname”, “email”, “role”, “gender” or “photo”; 2. System shows a “Cancel” and “Done” button; 3. User clicks on “Done” button; 4. System validates the information; 5. System checks if some field has changed. If positive, it updates the new data on the Database. System shows a successful message to the user. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 3, the User clicks on “Cancel”**   1. Goes to UC6.1 on Step 3.   **Alternative Flow 2 - At Step 4, if the User enters invalid data**   1. An error message is shown to the user; 2. Return to Step 1. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

### UC1.4 - Delete account

|  |  |
| --- | --- |
| **UC description - Delete account** | |
| **Objective** | The user wants to close her/his account. |
| **Priority** | Medium |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged). |
| **Post Conditions** | The user account is removed, if there are projects and work items in this account these ones continue to exist in the database. Current user is unlogged and the Login screen is shown. |
| **Basic Flow** | 1. User goes to Configuration icon, so user clicks on “Delete my account”; 2. System shows a confirmation message to user; 3. User could “Cancel” the operation; 4. User clicks on “Confirm” button; 5. System deletes his/her account. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 3, the User clicks on “Cancel”**   1. Return to Step 1. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

## 3.2 Project



“Project” tab contains a list of Projects that the logged user is a member. Clicking on the “+” button, the user can create projects. The user only informs a title and hit “Done”. Clicking on the “-” button (initially disabled), the user can remove empty projects. Error message must be shown if the project is not empty.

When a user lose membership in a project, this project is not shown anymore on the user’s listing anymore.

Right clicking in a project the user can: View Project Information, Manage Members and Change Project Name. Left-Clicking in the Project selects it and enable “-” (removal) button. Left Double-Clicking in a project works as “View Project Information”

On View Project Information, the upper breadcrumb is updated to show the project’s name and the screen gets two listings: one, on the left, shows the iterations on that project (higher to lower), and the other, on the right, shows the members of that project. The user can click the project’s name in the upper breadcrumb to come back to “Project” tab. When a project is created, the iteration “Backlog” is also created.

### UC2.1 - Create project

|  |  |
| --- | --- |
| **UC description - Create project** | |
| **Objective** | The user is creating a new project and the system shall create and show it on user’s Project tab. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged). |
| **Post Conditions** | The project was added to the list of user’s project and is shown on Project tab. |
| **Basic Flow** | 1. User goes to Project tab; 2. System loads a list with all the projects a user owns or has membership; 3. User clicks on “+” button; 4. System shows a popup, with an input field and a “Done” and “Cancel” button, so the user can fill the project name. 5. User fills the project name; 6. User clicks on “Done” button; 7. System validates the project name; 8. System adds the project name to the Database; 9. System adds the project name to the listing on Project tab; 10. System creates the iteration “Backlog” inside that project. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 6, the User clicks on “Cancel”**   1. Return to Step 2.   **Alternative Flow 2 - At Step 7, the informed project name has more than 40 characters**   1. An error message is shown to the user; 2. Return to Step 4.   **Alternative Flow 3 - At Step 8, the informed project name already exists on the Database**   1. An error message is shown to the user; 2. Return to Step 4. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
| Screen_Projects_Create | |

### 

### UC2.2 - View User Projects

|  |  |
| --- | --- |
| **UC description - View User Projects** | |
| **Objective** | The user enters the Project tab. The system shall show all the projects that the user has membership or ownership, ordered creation date. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and there’s at least 1 (one) project she/he is part of in the Project tab listing. A message stating that “there`s no project “ must be shown otherwise. |
| **Post Conditions** | The project list was loaded and shown to the user. |
| **Basic Flow** | 1. User goes to Project tab; 2. System loads a list with all the projects a user owns or has membership; 3. System shows a number, after each project name, showing the sum of Work Items points neither on “done” nor “deleted” state on the “Backlog” iteration of the project; 4. System shows a number, after each project name, showing the sum of Work Items points on Done state on the “Backlog” iteration of the project. |
| **Alternative Flow** |  |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
| Screen_Projects | |

### UC2.3 - View Project Information

|  |  |
| --- | --- |
| **UC description - View Project Information** | |
| **Objective** | The user enters a project to see its iterations. The system shall show it ordered and update the upper breadcrumb properly. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and UC 2.2 also was followed. |
| **Post Conditions** | The project’s iteration list was loaded, ordered by the higher (most recent, with the higher iteration number) iteration to the lowest and shown to the user. |
| **Basic Flow** | 1. User left-clicks on a project’s name; 2. System shows some options; 3. User clicks on “View Project Information”; 4. System asks the Database for the project’s iterations; 5. System shows a number, after each iteration name, showing the sum of Work Items points neither on “done” nor “deleted” state on that iteration; 6. System shows a number, after each iteration name, showing the sum of Work Items points on Done state on that iteration;   ystem changes the page’s upper breadcrumb to have the selected  project name on it. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 1, the User double-clicks on the project’s name**   1. Goes to Step 4. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

### UC2.4 - View Project Work Items

|  |  |
| --- | --- |
| **UC description - View Project Work Items** | |
| **Objective** | The user enters a project to see all its work items. The system shall show them and update the upper breadcrumb properly. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and UC 2.2 also was followed. |
| **Post Conditions** | The project’s work item list was loaded and shown to the user in the same way as described on UC5.2. |
| **Basic Flow** | 1. User left-clicks on a project’s name; 2. System shows some options; 3. User clicks on “View Project Work Item”; 4. System asks the Database for the project’s work items; 5. System changes the page’s upper breadcrumb to have the selected project name on it. |
| **Alternative Flow** |  |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

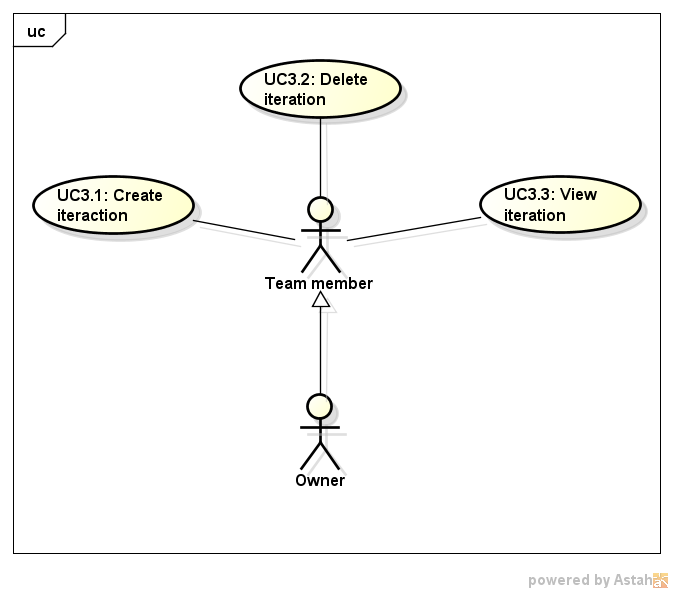
### UC2.5 - Change Project Name

|  |  |
| --- | --- |
| **UC description - Change Project Name** | |
| **Objective** | The owner of a project needs to rename a project. The system shall allow it. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and there’s at least 1 (one) project she/he owns of in the Project tab listing. |
| **Post Conditions** | The project’s iteration list was loaded with the proper project’s name alteration and shown to the user. |
| **Basic Flow** | 1. User goes to Project tab; 2. System loads a list with all the projects a user owns or has membership; 3. User left-clicks on a project’s name; 4. System shows some options; 5. User clicks on “Change Project Name”; 6. System shows a popup, with an input field populated with the project’s name and a “Done” and “Cancel” button, so the user can edit the project name; 7. User edits the project name; 8. User clicks on “Done” button; 9. System validates the project name; 10. System updates the project name entry into the Database; 11. System updates the project name entry on Project tab listing. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 8, the User clicks on “Cancel”**   1. System does not change the project name; 2. Return to Step 2.   **Alternative Flow 2 - At Step 9, the informed project name is empty**   1. An error message is shown to the user; 2. Return to Step 6.   **Alternative Flow 3 - At Step 9, the informed project name has more than 40 characters**   1. An error message is shown to the user; 2. Return to Step 6.   **Alternative Flow 4 - At Step 10, the informed project name already exists on the Database**   1. An error message is shown to the user; 2. Return to Step 6.   **Alternative Flow 5 - At Step 4, the User is not an owner of the selected project**   1. “Change Project Name” option is disabled (greyed out) and the user can’t click on it. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

### UC2.6 - Delete project

|  |  |
| --- | --- |
| **UC description - Delete Project** | |
| **Objective** | The user wants to close an empty project he owns and the system shall remove it from the user’s listing on Project tab. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and there’s at least 1 (one) project she/he owns of in the Project tab listing. |
| **Post Conditions** | The project was removed from the list of user’s project and is shown on Project tab. |
| **Basic Flow** | 1. User goes to Project tab; 2. System opens the user projects listing; 3. User right-clicks on a listed project; 4. System enables the “-” button; 5. User clicks on “-” button; 6. System shows a confirmation popup with “OK” and “Cancel” buttons; 7. User clicks on “OK” button; 8. System validates that the project does not have any work item by querying the Database; 9. System validates that the project does not have any members by asking the Database; 10. System removes the project name to the Database; 11. System removes the project name to the listing on Project tab. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 6, the User clicks on “Cancel”**   1. Return to Step 2.   **Alternative Flow 2 - At Step 3, the User clicked in a project he does not owns**   1. System will not enable the “-” button, so the user can't click on it; 2. Return to Step 3.   **Alternative Flow 3 - At Step 8, the Database informed that the project still have work items not on “deleted” status**   1. An error message is shown to the user; 2. Return to Step 3.   **Alternative Flow 3 - At Step 8, the Database informed that the project still have members**   1. An error message is shown to the user; 2. Return to Step 3. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

## 3.3 Iteration



View Iteration Information, which can be accessible by left Double-Clicking in a project’s name, shows the work items on that iteration, ordered by “points” on descendent order. The upper breadcrumb is updated to show the project’s name and the Iteration name. Any user can create Iterations.

When a project is created, the iteration number starts in 1 and have the special “Backlog” iteration already created. Clicking on the “+” button, the user can create Iterations. The System will create “Iteration N” and will decide what’s the value of N using that iteration number. Clicking on the “-” button, the user can remove empty iterations. The “Backlog” iteration can`t be deleted. Error message must be shown if the iteration is not empty.

The user can click the project’s name in the upper breadcrumb to come back to “View Project Information”.

A user receives notifications per email on creation and deletion of iterations on projects she/he owns.

### UC3.1 - Create iteration

|  |  |
| --- | --- |
| **UC description - Create Iteration** | |
| **Objective** | The user is creating a new iteration and the system shall create and show it when the user clicks on specific Project in Project tab. When an iteration is created an email should be sended to all project owners. |
| **Priority** | High |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and UC2.3 was loaded. All project members can add iterations. |
| **Post Conditions** | The iteration is added to the list of project’s iteration and is shown on Project name > Iteration number. |
| **Basic Flow** | 1. System shows all listed iterations for the selected project, the iterations are ordered by greater number; 2. User clicks on “+” button; 3. System creates a new iteration, incrementing the last iteration number created; 4. System adds the created iteration number to the Database; 5. System sends an email to all the project owners informing that a new Iteration was created; 6. System updates all listed iterations to Iterations listening. |
| **Alternative Flow** |  |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
| Screen_Create_Iterations | |

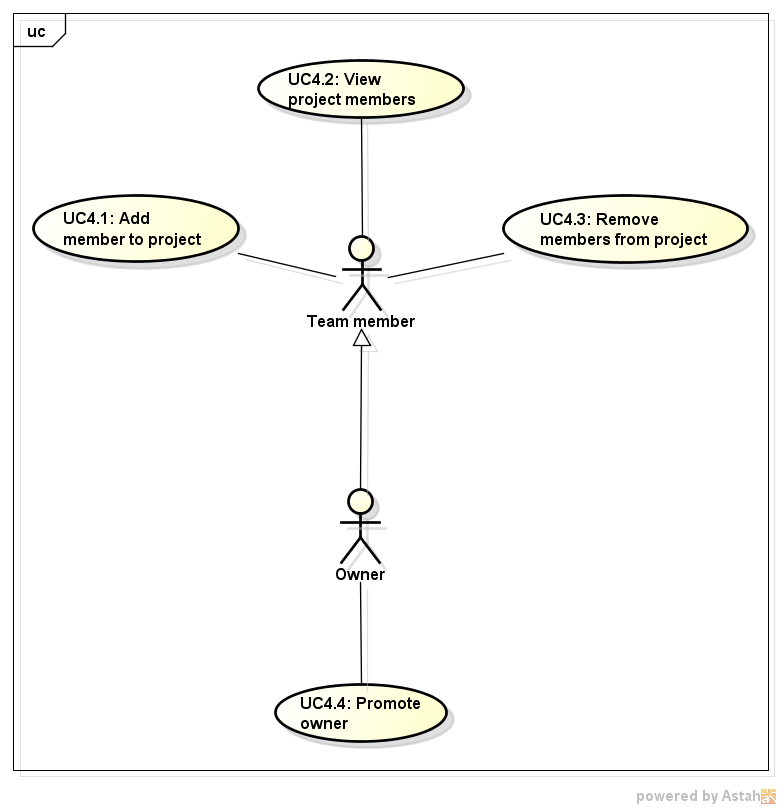
### UC3.2 - Delete iteration

|  |  |
| --- | --- |
| **UC description - Delete Iteration** | |
| **Objective** | The user wants to delete an empty iteration he owns and the system shall remove it from the Project’s iterations.. |
| **Priority** | High |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and UC2.3 was loaded. All project members can delete iterations. |
| **Post Conditions** | The iteration is deleted to the list of project’s iteration and is shown on Project. |
| **Basic Flow** | 1. System shows all listed iterations for the selected project, the iterations are ordered by greater number; 2. User clicks on a wished iteration; 3. User clicks on “-” button; 4. System validates that the iteration does not have any work item by querying the Database; 5. System deletes the iteration, decrementing the last iteration number created; 6. System sends an email to all the project owners informing that a Iteration was deleted; 7. System updates the listed iterations on Iterations listening. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 4, the User try to exclude an Iteration with work items**   1. An error message is shown to the user; 2. Return to Step 1.   **Alternative Flow 2 - At Step 4, the User try to exclude the “Backlog” Iteration**   1. An error message is shown to the user; 2. Return to Step 1. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

### UC3.3 - View iteration

|  |  |  |
| --- | --- | --- |
| **UC description - View Iteraction** | | |
| **Objective** | The user wish to view all iterations in a project, and the system shall provide a way to do that. When the user clicks on project link all iterations are listed. | |
| **Priority** | High | |
| **Source** | Lauriane Moraes (Business Analyst). | |
| **Actors** | * User | |
| **Preconditions** | UC1.1 was followed (User is logged) and UC2.3 was loaded. All project members can add iterations. | |
| **Post Conditions** | The iterations are listed. | |
| **Basic Flow** | 1. System shows the listed iterations for the selected project, the iterations are ordered by greater number. | |
| **Alternative Flow** |  | |
| **Exception Flow** |  | |
| **Notes/Issues** |  | |
| **Screen Flow** | | |
| Screen_List_Iterations | |

## 3.4 Manage Team Members



“Manage Members” option is reachable after right clicking in a project. Any user can Add or Remove members. The removal of members do not delete work items. All the project owners receives notifications per email on addition or removal of members on projects she/he owns.

The owner of a project can also give/remove ownership to other member by checking/unchecking a checkbox on the left of the member name. If checked, the member is also an owner and will receive the same notifications as the owner and can delete projects/iterations as such.

### UC4.1 - Add Member to Project

|  |  |
| --- | --- |
| **UC description - Add Member to Project** | |
| **Objective** | The User needs to add people (members) to work on her/his project and the system shall add it. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC4.2 has been followed. |
| **Post Conditions** | The user was added in the list of project’s users. The project was added in the member’s projects listing. |
| **Basic Flow** | 1. User clicks on “+” button; 2. System shows a popup, with an input field and a “Done” and “Cancel” button, so the user can search for a registered (or not) email to add to the project as a member; 3. User fills the email of the user he wants to search for; 4. User confirms the email and click on “Done”; 5. System validates the email informed; 6. System searches for the informed email in the Database; 7. System adds the user to the project’s members listing; 8. System adds the project to the user’s projects listing; 9. System sends an email to the user informing she/he gained access to a project; 10. System sends an email to all the project owners informing that a new member has gained access. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 3, the User searches for part of an email**   1. System open a list of already registered emails that matches the search string; 2. User clicks on the desirable email; 3. Go to Step 7.   **Alternative Flow 2 - At Step 6, the User searches for a valid email that does not belong to any registered user**   1. A warning message is shown to the User stating that an invite was sent to the informed email; 2. User clicks on “OK”; 3. System adds the user to the project’s members listing with a warning icon above the user name; 4. Go to UC4.2 on step 6. |
| **Exception Flow** | **Exception Flow 1 - At Step 11, the User searches for a non-valid email**  1. An error message is shown to the user;  2. Go to UC4.2 on step 6. |
| **Notes/Issues** | 1. A valid email has an “@” symbol, at least one character before it and at least one character after it. |
| **Screen Flow** | |
| Screen_Send_Invite | | |

### UC4.2 - View Project Members

|  |  |
| --- | --- |
| **UC description - View Project Members** | |
| **Objective** | The user needs to see people (members) that work on her/his project and the system shall show it. |
| **Priority** | High |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and UC2.2 was also followed. |
| **Post Conditions** | The project members listing is shown. |
| **Basic Flow** | 1. User right-clicks on a project’s name; 2. System shows some options; 3. User clicks on “Manage Members” button; 4. System opens the project members listing. Each member is a line on that listing. The list is ordered on alphabetical order; 5. System shows a number, after each member’s name, showing the sum of Work Items points neither on “done” nor “deleted” state where this user is the owner on the current project (remember, this screen is accessible by selecting a project); 6. System shows a number, after each member’s name, showing the sum of Work Items points on “done” state where this user is the owner on the current project (remember, this screen is accessible by selecting a project); 7. System changes the page’s upper breadcrumb to have the selected project name on it. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 4, the User has ownership on this project**   1. System shows a checkbox on the left of each user’s name. The checkbox is “checked” if the correspondent user is also an owner. The checkbox is “unchecked” otherwise. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
| Screen_List_Members | | |

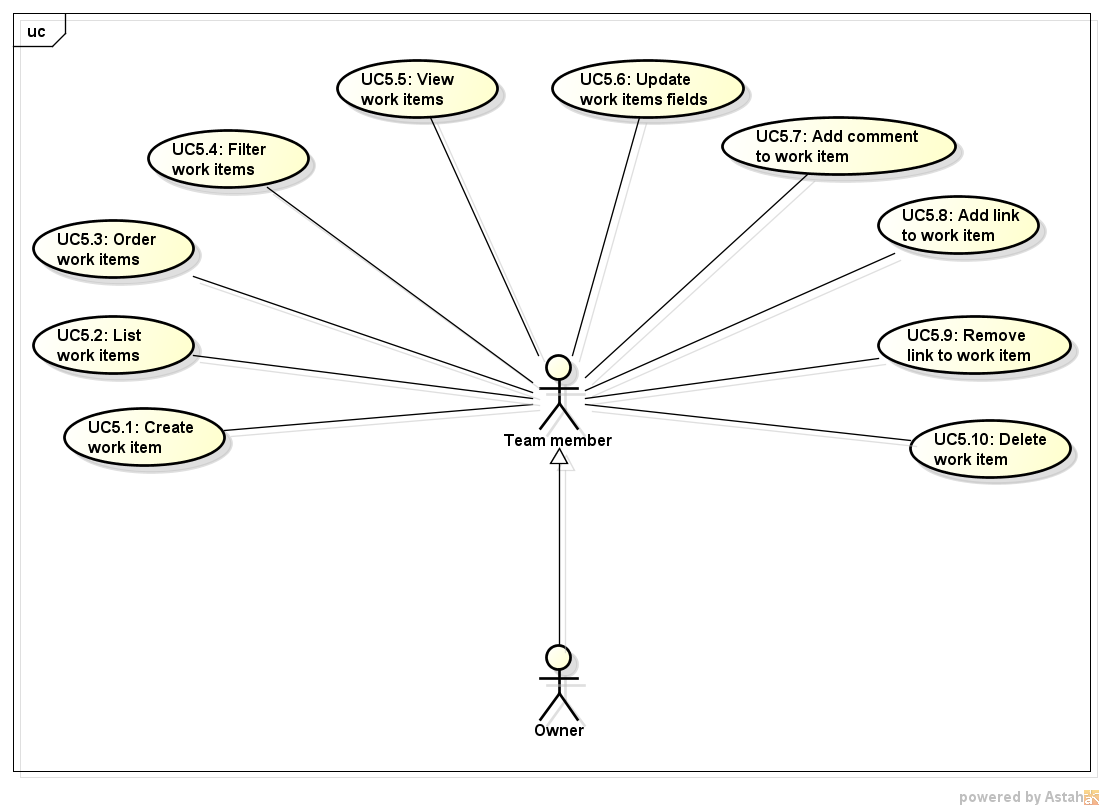
### UC4.3 - Remove Member from Project

|  |  |
| --- | --- |
| **UC description - Remove Member from Project** | |
| **Objective** | The user needs to remove people (members) that work on her/his project and the system shall remove it. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC4.2 has been followed. |
| **Post Conditions** | The user was removed of the list of project’s users. The project was removed from the user’s projects listing. |
| **Basic Flow** | 1. User left-clicks on a listed member; 2. System enables the “-” button; 3. User clicks on “-” button; 4. System shows a confirmation popup with “OK” and “Cancel” buttons; 5. User clicks on “OK” button; 6. System removes the user from the project’s members listing; 7. System removes the project from the user’s projects listing; 8. System sends an email to the member informing she/he lost her/his access; 9. System sends an email to all the project owners informing that a member has lost access. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 5, the User clicks on “Cancel”**   1. Go to UC4.2 on step 6.   **Alternative Flow 2 - At Step 1, the User clicked in a member name on a project he does not owns**   1. System will not enable the “-” button, so the user can't click on it; 2. Go to UC4.2 on step 6. |
| **Exception Flow** |  |
| **Notes/Issues** | 1. Remotion of members do not delete the work items she/he owns; 2. Only owners of a project can remove members. |
| **Screen Flow** | |
|  | |

### UC4.4 - Promote owner

|  |  |
| --- | --- |
| **UC description - Promote Owner** | |
| **Objective** | The user, who have a project, wishes to give Owner access to others on the project. All members allocated in the project have the same permissions when the owner option is checked. |
| **Priority** | High |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and there’s at least 1 (one) project she/he is owner. |
| **Post Conditions** | The owner access is released. The checkbox is “checked” if the correspondent user is also an owner. The checkbox is “unchecked” otherwise. |
| **Basic Flow** | 1. User goes to Project tab; 2. System loads a list with all the projects a user owns or has membership; 3. User right-clicks on a project’s name; 4. System shows some options; 5. User clicks on “Manage Members” button; 6. System opens the project members listing. Each member is a line on that listing. The system shows a checkbox on the left of each user’s name; 7. User marks the members who will have Owner access; 8. System releases the owner access to marked members. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 5, if the User don’t have ownership over the project**   1. System doesn’t show a checkbox on the left of each user’s name. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

## 3.5 Work item



Work Items are requirements or tasks. Or anything. Can be created inside an iteration, when clicking on “+” button. A popup (rounded-corners would be nice to have) will come up with all below information.

* The system shall create and attribute a sequential number as the work item identifier.  
  It must have “links” with other work items.
* It must have the following basic fields : name (input), description (multi-line input), points (input, only number), iteration (combobox, listing the iterations on current project, default to the current iteration), state (combobox with the options “not started”, “ongoing”, and “done”) and owner (combobox, listing the project’s members names) and the “created by” input, that can’t be edited, and shown the name of the user who created this Work Item.
* Any alteration to its fields is recorded as a comment. Any link addition/removal is recorded as a comment. Any user can add custom comments.
* Comments are read-only (can’t be deleted neither edited)

An user receives notifications per email on creation and deletion of work items on projects she/he owns.

### UC5.1 - Create work item

|  |  |
| --- | --- |
| **UC description - Create Work Item** | |
| **Objective** | The user wish to create a work item inside an iteration, that belongs to a project, and the system shall provide a way to do that. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC5.2 has been followed. |
| **Post Conditions** | The work item list is re-loaded, with the inclusion of the new work item, as on step 5 on UC5.2. |
| **Basic Flow** | 1. User clicks on “+” button; 2. System loads all the iterations in the selected project 3. System shows a popup with a “Done” and “Cancel” button and the basic fields described in the beginning of this chapter. “Add Comment” and “Add Link” buttons are present as well, and are better described on UC5.7 and UC5.8, respectively; 4. User fills in the information he wants; 5. User clicks on “Done” button; 6. System validates the information; 7. System generates an unique and sequential identifier to this new work item; 8. System creates on Database the new work item, along with its data; 9. System sends an email to all the project owners informing that a new work item has been created; 10. System save a new comment stating that the work item has been created by the given User, as per step 5.7.5.4 on UC5.7. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 6, the User clicks on “Cancel”**   1. Goes to step 12 on UC5.2.   **Alternative Flow 2 - At Step 6, the informed work item name is empty**   1. An error message is shown to the user as this is a mandatory field; 2. Return to Step 4.   **Alternative Flow 3 - At Step 6, the informed work item name has more than 40 characters**   1. An error message is shown to the user; 2. Return to Step 4.   **Alternative Flow 4 - At Step 6, the informed work item description is empty**   1. An error message is shown to the user as this is a mandatory field; 2. Return to Step 4.   **Alternative Flow 5 - At Step 6, the informed work item description has more than 165536 characters**   1. An error message is shown to the user; 2. Return to Step 4.   **Alternative Flow 6 - At Step 6, the informed points is not formed only by numbers**   1. An error message is shown to the user; 2. Return to Step 4.   **Alternative Flow 7 - At Step 6, the informed points is empty**   1. An error message is shown to the user as this is a mandatory field; 2. Return to Step 4.   **Alternative Flow 8 - At Step 5, the User adds a Comment to the Work item**   1. Goes to UC5.7; 2. Return to Step 4.   **Alternative Flow 9 - At Step 5, the User adds a Link to the Work item**   1. Goes to UC5.8; 2. Return to Step 4. |
| **Exception Flow** |  |
| **Notes/Issues** | 1. The system-generated unique identifier shall not be generated until the validation is made. |
| **Screen Flow** | |
|  | |

### UC5.2 - List Work Items

|  |  |
| --- | --- |
| **UC description - List Work Items** | |
| **Objective** | The user wish to list all work items inside an iteration, that belongs to a project, and the system shall provide a way to do that. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and UC2.3 was followed and there’s at least 1 (one) project he is part of in the Project tab listing. |
| **Post Conditions** | The work items from that iteration are shown, ordered by points (highest to lowest). |
| **Basic Flow** | 1. System shows the iterations listing for the selected project; 2. User right-double-clicks on an iteration’s name; 3. System asks the Database for the iteration’s work items; 4. System changes the page’s upper breadcrumb to have the selected project name and the selected iteration on it; 5. System shows the work items listing for the selected iteration in a table format; 6. System order the table by “points” column, on descendent order (higher to lower). All the other columns have no sorting criteria set; 7. System shows an empty first line in that table, with input fields below each columns’ name, so the User can filter the table properly. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 6, the User clicks on “Cancel”**   1. Return to Step 2.   **Alternative Flow 2 - At Step 3, the User clicked in a project he does not owns**   1. System will not enable the “-” button, so the user can't click on it; 2. Return to Step 3.   **Alternative Flow 3 - At Step 8, the Database informed that the project still have work items not on “deleted” status**   1. An error message is shown to the user; 2. Return to Step 3.   **Alternative Flow 3 - At Step 8, the Database informed that the project still have members**   1. An error message is shown to the user; 2. Return to Step 3. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
| Screen_Work_Item | | |

### UC5.3 - Order Work Items

|  |  |
| --- | --- |
| **UC description - Order Work Item** | |
| **Objective** | The user wish to order the work item list with several sorting criterias and the system shall provide a way to do that. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC5.2 has been followed and the iteration has work items on it or UC6.4 has been followed or UC2.4 has been followed. |
| **Post Conditions** | The work item list is re-loaded, with the proper sorting criteria set on the list, as on step 5 on UC5.2. |
| **Basic Flow** | 1. User left clicks on a column; 2. System set the sorting criteria for that column as ascendent (A->Z) if it was not set before, or as descendent (Z->A) if the previous sorting criteria was ascendent, or clear the sorting criteria entirely if the previous sorting criteria was descendent. |
| **Alternative Flow** |  |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

### UC5.4 - Filter Work Items

|  |  |
| --- | --- |
| **UC description - Filter Work Items** | |
| **Objective** | The user wish to filter the work item list and the system shall provide a way to do that. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC5.2 has been followed and the iteration has work items on it or UC6.4 has been followed or UC2.4 has been followed. |
| **Post Conditions** | The work item list is re-loaded, with the proper filter on the list, as on step 5 on UC5.2. |
| **Basic Flow** | 1. User left clicks on a column input field (created on step 5.2.5.14 of UC 5.2); 2. System allow the user to type on that field; 3. User hit “ENTER” keyboard key. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 2, the User clicks on “ESC” keyboard key or remove the focus on that input by mouse clicking on another place on the screen**   1. Goes to step 5 on UC5.2. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

### 

### UC5.5 - View Work Item

|  |  |
| --- | --- |
| **UC description - View Work Item** | |
| **Objective** | The user wish to view a work item inside an iteration, that belongs to a project, and the system shall provide a way to do that. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC5.2 has been followed and the iteration has work items on it or UC6.1 has been followed or UC6.4 has been followed or UC2.4 was followed. |
| **Post Conditions** | The work item information is shown on a popup. |
| **Basic Flow** | 1. User left-double clicks on a work item name; 2. System asks the Database for that work item information; 3. System shows a popup with a “Done” and “Cancel” button and the work item information. |
| **Alternative Flow** |  |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

### UC5.6 - Update Work item Fields

|  |  |
| --- | --- |
| **UC description - Update Work Item Fields** | |
| **Objective** | The user wish to edit a work item inside an iteration, that belongs to a project, and the system shall provide a way to do that. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC5.5 has been followed. |
| **Post Conditions** | The work item list is re-loaded, with the proper alteration of the work item, as on step 5 on UC5.2. |
| **Basic Flow** | 1. User changes the information he wants the basic fields described in the beginning of this chapter; 2. User clicks on “Done” button; 3. System validates the information; 4. System checks if the “name” field has changed. If positive, it saves a new comment informing the old and new value of it, as per step 4 on UC5.7; 5. System checks if the “points” field has changed. If positive, it saves a new comment informing the old and new value of it, as per step 4 on UC5.7; 6. System checks if the “state” field has changed. If positive, it saves a new comment informing the old and new value of it, as per step 4 on UC5.7; 7. System checks if the “owner” field has changed. If positive, it saves a new comment informing the old and new value of it, as per step 4 on UC5.7; 8. System checks if the “iteration” field has changed. If positive, it saves a new comment informing the old and new value of it, as per step 4 on UC5.7; 9. System checks if the “description” field has changed. If positive, it saves a new comment informing the old and new character count of it, as per step 4 on UC5.7; 10. System updates on Database the work item information. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 2, the User clicks on “Cancel”**   1. Goes to step 5 on UC5.2.   **Alternative Flow 2 - At Step 3, the informed work item name is empty**   1. An error message is shown to the user as this is a mandatory field; 2. Return to Step 1.   **Alternative Flow 3 - At Step 3, the informed work item name has more than 40 characters**   1. An error message is shown to the user; 2. Return to Step 1.   **Alternative Flow 4 - At Step 3, the informed work item description is empty**   1. An error message is shown to the user as this is a mandatory field; 2. Return to Step 1.   **Alternative Flow 5 - At Step 3, the informed work item description has more than 165536 characters**   1. An error message is shown to the user; 2. Return to Step 1.   **Alternative Flow 6 - At Step 3, the informed points is empty**   1. An error message is shown to the user as this is a mandatory field; 2. Return to Step 1.   **Alternative Flow 7 - At Step 3, the informed points is not formed only by numbers**   1. An error message is shown to the user; 2. Return to Step 1.   **Alternative Flow 8 - At Step 1, the work item state is set to “deleted” or “done”**   1. All the fields (except the “state” field) are on disabled state (on gray color) and can’t be modified; 2. Return to step 1. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
| Screen_Work_Item_Edit | | |

### UC5.7 - Add Comment to Work item

|  |  |
| --- | --- |
| **UC description - Add Comment Work Item** | |
| **Objective** | The user wish to add a comment to a work item inside an iteration, that belongs to a project, and the system shall provide a way to do that. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC5.1 or UC5.5 has been followed. |
| **Post Conditions** | The “comments” read-only field is updated with the new comment line. |
| **Basic Flow** | 1. User types the comment on “new comment” field on the work item popup; 2. User clicks on “Add Comment” button; 3. System validates the new comment; 4. System writes the current date, the User name and the new comment in a new line on the “comments” read-only field. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 3, the informed new comment has more than 1024 characters**   1. An error message is shown to the user; 2. Return to Step 1. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

### UC5.8 - Add Link to Work item

|  |  |
| --- | --- |
| **UC description - Add Link to Work Item** | |
| **Objective** | The user wish to add a link to another work item and the system shall provide a way to do that. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC5.1 or UC5.5 has been followed. |
| **Post Conditions** | The “link” read-only table is updated with the new link. |
| **Basic Flow** | 1. User fills the work item name she/he wants to search for; 2. User clicks on “Add Link” button; 3. System validates the work item name informed; 4. System searches for the informed work item name that belongs to the current project in the Database; 5. System adds the work item name to the “link” read-only table on this Work item; 6. System adds this Work Item Name to the “link” read-only table on the searched work item. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 3, the User searches for part of a work item name**   1. System open a list of work items in the current project that contains the searched string; 2. User clicks on the desirable work item; 3. Go to Step 4.   **Alternative Flow 2 - At Step 2, the User searches for a work item name that does not exist in the current project**   1. A warning message is shown to the User stating that no work item was found; 2. User clicks on “OK”; 3. Go to Step 1. |
| **Exception Flow** |  |
| **Notes/Issues** | 1. If the user double-clicks a Work Item name on the “link” read-only table, the UC5.5 shall be followed and the user will be able to update the linked work item. |
| **Screen Flow** | |
|  | |

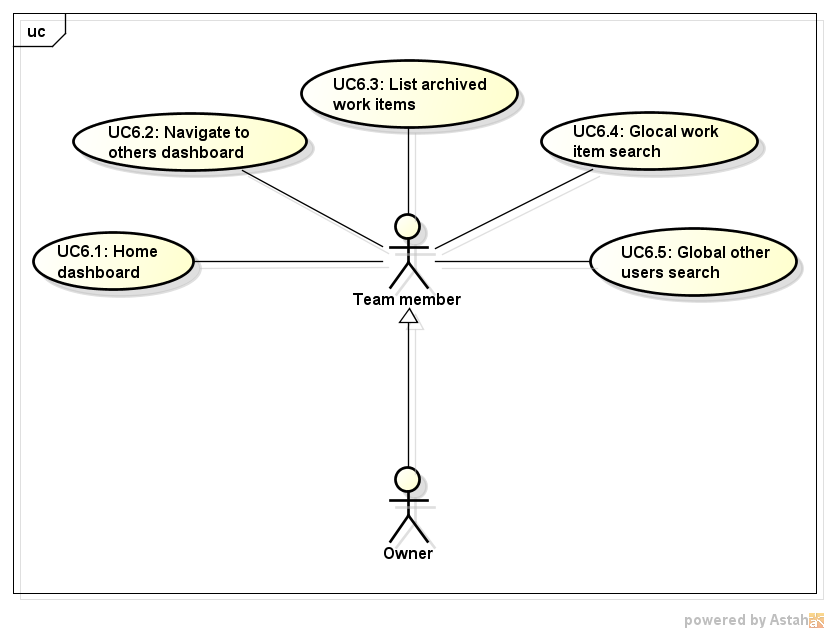
### UC5.9 - Remove Link to Work item

|  |  |
| --- | --- |
| **UC description - Remove Lonk Work Item** | |
| **Objective** | The user wish to remove a link to another work item and the system shall provide a way to do that. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC5.1 or UC5.5 has been followed. |
| **Post Conditions** | The “link” read-only table is updated with the removal of the link. |
| **Basic Flow** | 1. User clicks a work item name on the “link” read-only table; 2. User clicks on “Remove Link” button; 3. System shows a confirmation popup with “OK” and “Cancel” buttons; 4. User clicks on “OK” button; 5. System removes the work item name on the “link” read-only table on this Work item; 6. System removes this Work Item Name on the “link” read-only table on the searched work item. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 4, the User clicks on “Cancel”**   1. Goes to Step 1. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

### UC5.10 - Delete work item

|  |  |
| --- | --- |
| **UC description - Delete Work Item** | |
| **Objective** | The user wish to delete an work item in a way that its information are kept on the system. |
| **Priority** | High |
| **Source** | Gabriel Oliveira (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC5.2 has been followed or UC6.4 has been followed. |
| **Post Conditions** | The work item list is re-loaded, with the proper alteration of the work item, as on step 5 on UC5.2. |
| **Basic Flow** | 1. User left-clicks in a work item name; 2. User clicks on “-” button; 3. System shows a confirmation popup with “OK” and “Cancel” buttons; 4. User clicks on “OK” button; 5. System updates on Database the work item state field to “deleted”; 6. System sends an email to all the project owners informing that a work item has been set to “deleted” state; 7. System save a new comment stating that the work item has been set to “deleted” state by the given User, as per step 4 on UC5.7. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 6, the User clicks on “Cancel”**   1. Goes to Step 1. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

## 3.6 Home Dashboard



“Home” tab contains:

* The user’s information (name and photo);
* A listing, on the right, with all members of all the projects the user belongs to. Can be grouped by project and have a “+”/”-” to expand/collapse;
* A listing, on the left, with “Open Items” title, with all “not done” work items where the user is the owner ordered by points on descendent order and grouped by project (should have a “+”/”-” to expand/collapse it);
* Should have a “count of points” on the side of the project name;
* A link, under/next “Open Items” to the “Archived Items”, a list of “Done” work items where the user is the owner ordered by points on descendent order. When a “Done” Work Item is loaded should have read-only fields.

A “search work item” input is always on the screen. It searches under all the projects the user has access to.

When a user lose ownership of a work item, this item is not shown on the user’s dashboard anymore, not even on Archived Work Items.

### UC6.1 - Home Dashboard

|  |  |
| --- | --- |
| **UC description - Home Dashboard** | |
| **Objective** | When an user have her/his account loaded the system displays a Home dashboard, in Home tab. The screen have all strategic information to member have a nice control about the project progress. There are her/his personal data (These data are inputted when account is created): Photo, Name + Surname and Email. When her/his have any open work item should be displayed in a list. When there are done work items should be displayed in “Archived Work Items” link. All team members are displayed by project. |
| **Priority** | High |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged). |
| **Post Conditions** | The dashboard is displayed. |
| **Basic Flow** | 1. System loads the account; 2. User could click on Home tab; 3. System loads the personal data in a specific section; 4. System loads all open work items owned by this account (State = Not started/Ongoing), ordered by project and have a “+”/”-” to expand/collapse. These listed projects should have a status field where there are projects statuses (Done per Not done/Not deleted). Iterations are in an expanded Project, ordered by most recent iteration created and have a “+”/”-” to expand/collapse. The open work items should be in links to redirect to their work item page; 5. System loads a Team Member listing with all project members, ordered by project and have a “+”/”-” to expand/collapse. For all members should have links to redirect to their dashboard; 6. User could click on “Archived Work Items” link. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 4, the User don’t have any open work items owned by him/her**   1. System displays a project name she/he belongs to in Open Work Items listing, but the expand project option should be disabled; 2. Goes to Step 3.   **Alternative Flow 2 - At Step 4, the User don’t have any project**   1. System displays an empty Open Work Items listing; 2. Goes to Step 3.   **Alternative Flow 3 - At Step 5, if there is a team member who deleted the account**   1. System displays a Team Members listing, but there is no link to redirect the user to other; 2. Goes to Step 3.   **Alternative Flow 4 - At Step 6, the User don’t have any Done work items**   1. System don’t display an “Archived Work Items” link; 2. Goes to Step 3. |
| **Exception Flow** |  |
| **Notes/Issues** | 1. The sum of Work Items points Not done and Not deleted on the first iteration of the project (Backlog); 2. The sum of Work Items points Done on the first iteration of the project (Backlog). |
| **Screen Flow** | |
| Screen_Home | |

### UC6.2 - Navigate to others Dashboard

|  |  |
| --- | --- |
| **UC description - Navigate to other Dashboard** | |
| **Objective** | If the user have a team project/many team projects, this user can navigate to others. |
| **Priority** | High |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and UC6.1 was loaded. |
| **Post Conditions** | The navigation is available. |
| **Basic Flow** | 1. User clicks on “+” to expand the project members; 2. System lists all project members on alphabetical order; 3. System provides a link to all members; 4. User clicks on an specific member link; 5. System loads the member’s dashboard (same as UC6.1) to selected member. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 2, if a member deleted her/his account**   1. System don’t display a member in the Team Member listing (according to UC4.3); 2. Goes to Step 2. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

### UC6.3 - List Archived Work Items

|  |  |
| --- | --- |
| **UC description - List Archived Work Items** | |
| **Objective** | The user can see all its “Done” work items. The system shall show them.  User can access “Done” work items where the user is the owner ordered by points on descendent order. |
| **Priority** | High |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged) and UC6.1 was loaded. |
| **Post Conditions** | The “Done” Work Items should be displayed.The project’s work item list was loaded and shown to the user in the same way as described on UC5.2. |
| **Basic Flow** | 1. User clicks on “Archived Work Items” link; 2. System lists all “Done” Work Items owned by this account and created by this account. The table have the following fields: Project, Iteration, Work item id, Work item title, Created by; 3. System provides an item link to read the "Done" Work Item, because when a Work Items is in “Done” state all fields are read-only and this one can't be reopened, but the Work Item can be deleted. |
| **Alternative Flow** |  |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

### UC6.4 - Global Work Item Search

|  |  |
| --- | --- |
| **UC description - Global Work Item Search** | |
| **Objective** | The user wish to search for a work item (or a set of related work items with the same name) and the system shall provide a way to do that. |
| **Priority** | High |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged). |
| **Post Conditions** | The work items that fulfill the searched criteria are shown, ordered by points (highest to lowest). |
| **Basic Flow** | 1. User fills the work item name she/he wants to search for, preceeded by the “workitem:” keyword; 2. User clicks on “Search” button; 3. System validates the work item name informed; 4. System searches for the informed work item name in the Database (it can belong to any project the user has access to); 5. System open a list of work items (in the same way as UC5.2) that contains the searched string. |
| **Alternative Flow** | **Alternative Flow 1 - At Step 3, the User searches for part of a work item name**   1. System opens a list of work items that contains the searched string; 2. User clicks on the desirable work item; 3. Go to Step 4.   **Alternative Flow 2 - At Step 2, the User searches for a work item name that does not exist in the current project**   1. A warning message is shown to the User stating that no work item was found; 2. User clicks on “OK”; 3. Go to Step 1. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

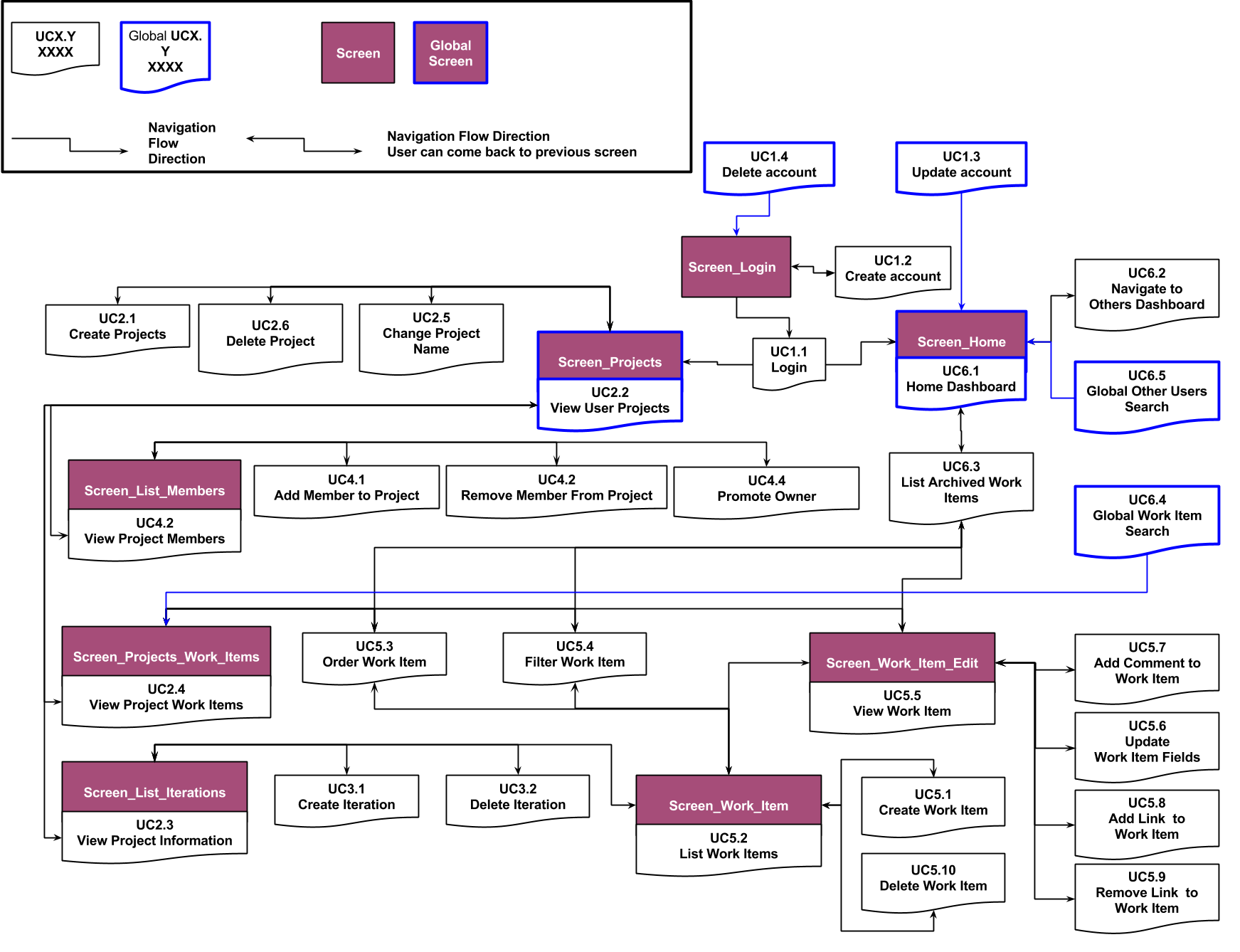
### 

### UC6.5 - Global Other Users Search

|  |  |
| --- | --- |
| **UC description - Global Other User Search** | |
| **Objective** | The user wish to search for other users dashboards and the system shall provide a way to do that. |
| **Priority** | High |
| **Source** | Lauriane Moraes (Business Analyst). |
| **Actors** | * User |
| **Preconditions** | UC1.1 was followed (User is logged). |
| **Post Conditions** | The searched user dashboard is shown, in the same way as UC6.1. |
| **Basic Flow** | 1. User fills the work item name she/he wants to search for, preceeded by the “user:” keyword;; 2. User clicks on “Search” button; 3. System validates the member name informed; 4. System searches for the member name in the Database; 5. System opens the searched user dashboard (in the same way as UC6.1). |
| **Alternative Flow** | **Alternative Flow 1 - At Step 3, the User searches for part of a user name**   1. System opens a list of users that contains the searched string; 2. User clicks on the desirable user; 3. Go to Step 4.   **Alternative Flow 2 - At Step 2, the User searches for a work item name that does not exist in the system**   1. A warning message is shown to the User stating that no user was found; 2. User clicks on “OK”; 3. Go to Step 1. |
| **Exception Flow** |  |
| **Notes/Issues** |  |
| **Screen Flow** | |
|  | |

## 

## 3.7 Screen Diagram



## 3.8 Use Case Diagram

