

Use case template

Descriptions of template fields:

- **ID and name:** Title should be descriptive and should usually begin with a verb, e.g. order, calculate, input, etc. ID can have any format but must be unique among all use cases.
- **Primary actor:** Person that wishes to accomplish a goal through the use of the system. Only a single primary actor per use case.
- **Secondary actors:** Actors that have an interest in the completion of the goal but that do not directly interact with the system.
- **Description:** Concise description of the purpose of the use case.
- **Trigger:** Condition internal or external to the system that prompts the use case to start.
- **Preconditions:** Conditions that must be true before the use case starts. Each should be labeled with an ID unique to the use case.
- **Postconditions:** Conditions that must be true after the use case ends normally. Each should be labeled with an ID unique to the use case.
- **Normal flow:** Detailed step-by-step description of the logical flow of the use case. It should describe an explicit two way interaction, with the system prompting for input and the actor responding accordingly. Each step should be numbered.
- **Alternative flows:** Flows that achieve the same goal as the normal flow but are expected to be less common or lower priority.
- **Exceptions:** Conditions that result in the normal flow ending prematurely due to an unrecoverable condition in the system. The condition that causes the flow should be clearly stated, as should be any other decisions that the actor must make in this situation.

ID and name	UC-01. Add bug		
Primary actor	Admin / Tester	Secondary actors	
Description	The user gives the information needed to add a new bug and the system saves it and displays it.		
Trigger	Selecting the 'Add bug' option.		
Preconditions	PRE-01. The user must be logged in beforehand as either an admin or a tester.		
Postconditions	POST-01. The list of bugs is updated to reflect the changes made.		
Normal flow	<ol style="list-style-type: none">1. System shows the user the list of options.2. User selects the add bug option.3. System opens a new view with input fields.4. User inputs the data requested (name, summary, functionality, description, steps to reproduce) and selects the add bug option.		

	5. System adds the newly created entity, lets the user know everything went well with a pop up ("Bug added!") and redirects them back to their home page.
Alternative flows	-
Exceptions	4b. The user inserted "bad" data and the System shows an error telling the user what inputs need to be changed.

ID and name	UC-02. Remove bug		
Primary actor	Admin	Secondary actors	-
Description	User opens the system, selects the bug they want to remove and then selects the 'Remove bug' option. The System removes it and shows the updated list.		
Trigger	The User selects the 'Remove bug' option.		
Preconditions	PRE-02. The user must be logged in beforehand asr an admin.		
Postconditions	POST-01. The list of bugs is updated to reflect the changes made.		
Normal flow	<ol style="list-style-type: none"> 1. System shows the user the list with bugs. 2. User selects what bug they wish to remove from the displayed list and then selects the 'Remove bug' option. 3. System removes the bug and lets the user know it went well with a pop up ("Bug removed!"). 		
Alternative flows	2b. User selects multiple bugs from the list. 3b. The System removes all selected bugs.		
Exceptions	3c. The user didn't select a bug before they selected the 'Remove but' option, the System shows an error to let the user know.		

ID and name	UC-03. Update bug		
Primary actor	Admin	Secondary actors	-
Description	User opens the system, selects the bug they want to update and then		

	selects the 'Update selected bug' option. The System asks the user for new information. User gives the information and System updates the said bug.
Trigger	User selects the 'Update selected bug' option.
Preconditions	PRE-02. The user must be logged in beforehand as an admin.
Postconditions	POST-01. The list of bugs is updated to reflect the changes made.
Normal flow	<ol style="list-style-type: none"> 1. System shows the user the list of bugs. 2. User selects a bug and then the 'Update selected bug' option. 3. System redirects the user to a new window with input fields which contain the information that is currently saved about the bug. 4. The user changes the fields they desire and selects the 'Update' option. 5. System lets the user know everything went well with a pop up ("Update done!"), and then redirects the user to their home page.
Alternative flows	-
Exceptions	<p>3b. The user did not select a bug, the System will show an error letting the user know.</p> <p>5b. The user gave 'bad' input. System will show an error letting the user know.</p>

ID and name	UC-04. See all bugs		
Primary actor	Admin / Tester / Programmer	Secondary actors	-
Description	User opens the system to see all the bugs that are currently listed in the application.		
Trigger	User logges in.		
Preconditions	PRE-03. The user must be logged in as either a tester, programmer or an admin.		
Postconditions	-		
Normal flow	<ol style="list-style-type: none"> 1. System shows user the 'Log in' option 		

	<ol style="list-style-type: none"> User logges in. System shows the user the list of bugs currently listed in the application.
Alternative flows	-
Exceptions	3b. Something goes wrong with the log in and the user is prompted with an error message.

ID and name	UC-05. Add functionality for testing		
Primary actor	Admin	Secondary actors	-
Description	The User adds to the system a new functionality for testing		
Trigger	User selects the 'Add new functionality' option.		
Preconditions	PRE-02. The user must be logged in as an admin.		
Postconditions	POST-02. The list of functionalities is updated to reflect the changes made.		
Normal flow	<ol style="list-style-type: none"> The System shows the user the option to add a new functionality. User selects the option. System redirects the user to a new window with multiple input fields (name, description, date). User inputs data and selects the 'Add' option. System adds the new option and lets the user know everything went well with a pop up ("Functionality added!"), then redirects them to their home page. 		
Alternative flows	-		
Exceptions	5b. User added 'bad' data, the system will show an error to let the user know.		

ID and name	UC-06. Remove functionality for testing		
Primary actor	Admin	Secondary actors	-

Description	The User removes an already existing functionality.
Trigger	User selects the 'Remove Functionality' option.
Preconditions	PRE-02. The user must be logged in as an admin.
Postconditions	POST-02. The list of functionalities is updated to reflect the changes made.
Normal flow	<ol style="list-style-type: none"> 1. System shows the user a list of functionalities. 2. User selects one of them and selects the 'Remove functionality' option. 3. The System removes the selected functionality and lets the user know it went well with a pop up ("Functionality removed!").
Alternative flows	<p>2b. User selects multiple functionalities</p> <p>3b. The System removes all selected functionalities and lets the user know it went well.</p>
Exceptions	3c. The user didn't select a functionality, the System shows an error to let the user know.

ID and name	UC-07. Update functionality for testing		
Primary actor	Admin	Secondary actors	-
Description	The User updates an already existing functionality.		
Trigger	User selects the 'Update Functionality' option.		
Preconditions	PRE-02. The user must be logged in as an admin.		
Postconditions	POST-02. The list of functionalities is updated to reflect the changes made.		
Normal flow	<ol style="list-style-type: none"> 1. System shows the user a list of functionalities. 2. User selects one and the selects the ' Update functionality' option. 3. System redirects user to a new window with multiple input fields which contain the current information about the functionality. 4. The user modifies whichever inputs they may want, then selects the 'Update' option. 		

	5. The System updates the said functionality and lets the user know it went well with a pop up ("Functionality updated").
Alternative flows	-
Exceptions	3b. User didn't select a functionality and the System shows an error to let the user know. 5b. The user inserted 'bad' data and the system will show an error to let the user know.

ID and name	UC-08. See all functionality for testing		
Primary actor	Admin	Secondary actors	-
Description	User opens the system to see all the functionalities that are currently listed in the application.		
Trigger	The user logges in.		
Preconditions	PRE-02. The user must be logged in as an admin.		
Postconditions	-		
Normal flow	<ol style="list-style-type: none"> 1. System shows user the 'Log in' option. 2. The user selects it. 3. The system shows the user all the functionalities currently known in the application. 		
Alternative flows	-		
Exceptions	3b. Something goes wrong with the login and the user is notified with an error message		

ID and name	UC-09. Sort bugs		
Primary actor	Admin, Tester	Secondary actors	-
Description	The User wishes to sort the displayed bugs in an order for easier use of the data.		

Trigger	User selects the 'Sort bugs' option.
Preconditions	PRE-01. The user must be logged in beforehand as either an admin or a tester.
Postconditions	POST-01. The list of bugs is updated to reflect the changes made.
Normal flow	<ol style="list-style-type: none"> 1. System shows the user the option to sort the displayed bugs. 2. User selects it. 3. The system will sort the bugs by a predefined condition and show the user the newly arranged data.
Alternative flows	-
Exceptions	3b. There are no items in the list therefore the System won't display anything.

ID and name	UC-10. Sort functionalities		
Primary actor	Admin / Programmer	Secondary actors	-
Description	The User wishes to sort the displayed functionalities in an order for easier use of the data.		
Trigger	The user selects the 'Sort functionalities' option		
Preconditions	PRE-04. The user must be logged in beforehand as either an admin or a programmer.		
Postconditions	POST-02. The list of functionalities is updated to reflect the changes made.		
Normal flow	<ol style="list-style-type: none"> 1. System shows the user the option to sort the displayed functionalities. 2. The user selects it. 3. The system will sort the bugs by a predefined condition and show the user the newly arranged data. 		
Alternative flows	-		
Exceptions	3b. There are no items in the list therefore the System won't display anything.		

ID and name	UC-11. Solve bug		
Primary actor	Admin / Programmer		
Description	The user solved a listed bug and wants to update the list to reflect it as such.		
Trigger	Selecting the 'Solve bug' option.		
Preconditions	PRE-04. The user must be logged in beforehand as either an admin or a programmer.		
Postconditions	POST-01. The list of bugs is updated to reflect the changes made.		
Normal flow	<ol style="list-style-type: none"> 1. The system shows the user the list of bugs and the option to 'solve bug'. 2. The user selects a bug and the option. 3. The system updates the bug and now saves it as solved. The system shows the user a message ("Bug resolved!") to let them know the operation went well. 		
Alternative flows	-		
Exceptions	3b. The user didn't select a bug so the system shows the user an error to let them know.		

ID and name	UC-11. Log out		
Primary actor	Admin / Tester / Programmer	Secondary actors	-
Description	The User exits the application.		
Trigger	User selects the 'Log out' option		
Preconditions	PRE-03. The user must be logged in as either a tester, programmer or an admin.		
Postconditions	POST-03. User is shown the login screen.		
Normal flow	<ol style="list-style-type: none"> 1. System shows the user the 'log out' option. 2. The user selects it. 3. System redirects the user to the login page. 		

Alternative flows	-
Exceptions	-

ID and name	UC-12. Log in		
Primary actor	Admin / Tester / Programmer		
Description	The User wants to enter the application.		
Trigger	User selects the 'Login' option		
Preconditions	PRE-04. The user must have an account as tester, admin or programmer.		
Postconditions	POST-03. The user is shown the main page screen correspondent to their role.		
Normal flow	<ol style="list-style-type: none"> 1. The system shows the user the input fields for the login and the login option. 2. The user completes the fields and selects the option. 3. The system redirects the user to their main page. 		
Alternative flows	-		
Exceptions	3b. The user inserted wrong data, the system will show them an error to let the user know.		