## **Use case template**

5. Urmarire Bug-uri

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | T1: Log in | | |
| Primary actor | User | Secondary actors | User Database |
| Description | A User (Tester or Programmer) accesses the login window to view their workspace. | | |
| Trigger | A User wants to register/resolve a bug. | | |
| Preconditions | PRE1: The login information exists in the database | | |
| Postconditions | POST1: The User can access their respective windows. | | |
| Normal flow | 1. Login successfully 2. The User is prompted with two text-fields to fill out their account information: user and password and a Login button to access the platform. 3. The User enters their information, with the password remaining hidden. 4. The User presses the Login button 5. The User can now access the platform. | | |
| Alternative flows | 1.1. The User has entered a non-existent username  1. The user is informed that the username that they have entered does not exist in the database, and an OK button  2. The user presses OK  3. Return to step 1 of the normal flow  1.2. The User has entered their password incorrectely  1. The user is informed that the password they enetered does not match with the username, and an OK button  2. The user presses OK  3. Return to step 1 of the normal flow, with the username filled out | | |
| Exceptions | 1.0E One of the text-fields has been left blank  1. The user is informed that they should fill out all the information before trying to log in, and an OK button  2. The user presses OK  3. Return to step 1 of the normal flow. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | T2: Report a bug | | |
| Primary actor | Tester | Secondary actors | Programers |
| Description | A Tester accesses the Bug-Report Form, entering a name ,a description and confirms his registration in the system. | | |
| Trigger | A Tester wants to register a bug. | | |
| Preconditions | PRE1: The Tester is logged into the system | | |
| Postconditions | POST1: The bug is stored in the database  POST2: The bug is visible to all Programers that are currently logged in. | | |
| Normal flow | 1. Register a bug   1.The Tester can see the current list of Bugs to check if it hasn’t been reported yet  2.The Tester is prompted with two text-fields to fill out the bug’s name and description, and a Confirm button  3.The Tester fills out the form and presses Confirm  4.The Tester recieves a confirmation message that asks if they would like to make any changes to the bug’s information before it is stored in the system  5.The Tester presses Send  6.The bug is storred in the database  7.All other logged in Programers can see the changes made in the bug list | | |
| Alternative flows | 2.1 Change the bug’s information  1. The Tester wants to change the bug’s information before sending  2. Return to step 2 of the normal flow, keeping the Testers original input | | |
| Exceptions | 2.0E The form has been left blank  1. The system informs the Tester with a warning message that they’ve left the form blank  2. The bug with incomplete information is not stored in the database | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ID and name | T3: Mark a bug as resolved | | |
| Primary actor | Programmer | Secondary actors | Bug Database |
| Description | A Programmer has resolved a bug from the list and wants to delete it. | | |
| Trigger | A Programmer has resolved a bug. | | |
| Preconditions | PRE1: The Programmer is logged into the system | | |
| Postconditions | POST1: The bug is deleted from the database  POST2: The change is visible to all Programers that are currently logged in. | | |
| Normal flow | 1. Delete a bug   1.The Programmer can search the bug in the list by name.  2.The Programmer presses the delete button after selecting a bug.  3.The Programmer is prompted with a confirmation button.  4.The Programmer presses Confirm   1. The Programmer recieves a confirmation message that the bug has been deleted 2. The bug is deleted from the database 3. All other logged in Programers can see the changes made in the bug list | | |
| Alternative flows | 2.1 Reject the confirmation to delete the bug  1. The Programmer doesn’t want to delete the bug that thay’ve chosen, pressing the No button from the confirmation prompt  2. Return to step 1 of the normal flow  2.2 The Programmer wants to delete multiple bugs  1. The Programmer selects multiple bugs from the list  2. Continue with step 3 of the normal flow | | |
| Exceptions | 3.0E No bug has been selected  1. The Programmer recieves a warning message.  2. Continue with step 1 of the normal flow | | |