Use Cases for Bug Application

|  |  |  |  |
| --- | --- | --- | --- |
| Id and name | UC -1: Add a Bug | | |
| Primary actor | Tester | Secondary Actors | Programmer |
| Description | The tester discovers a new bug and decides to add it to the bugs list. He sees the bugs list and tries to add a new bug by giving it a short description and a level of risk | | |
| Trigger | A tester indicates that he wants to to add a new bug | | |
| Preconditions | PRE 1 – Tester is logged in into BA  PRE 2 – The user must be in a tester account | | |
| Postconditions | POST 1 – The new bug is added to the list with status of “Unresolved”  POST 2 – List is updated so all others tester and programmers are able to see the new bug | | |
| Normal flow | 1. Add valid bug 2. Tester asks to add new bug 3. BA displays a window where the attributes of the task must be introduced (description, level of risk) 4. Tester writes the description and choose the level of risk 5. Tester selects that the bug can be added 6. BA displays a message of confirmation 7. BA updates the list of bugs with the new one | | |
| Alternative flow | 1.1Cancel a bug  4. Tester selects to cancel the procces  5. BA displays a message of cancelation confirmation | | |
| Exceptions | 1. E1 The attributes of the bug are empty   1. BA informs tester that the attributes are empty  2 a. If the tester cancel the bug addition proccess, BA terminates the UC  2 b. If the tester fills in the empty attributes then return to step 4 of normal flow | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Id and name | UC-2 : Update bug status | | |
| Primary actor | Programmer | Secondary Actors | Bug Application |
| Description | A Programmer accesses the Bug Application, views the list of bugs and select one to change its status to the current one. | | |
| Trigger | A Programmer indicates that he wants to change the status of a bug | | |
| Preconditions | PRE 1 – Programmer must be logged in, in a programmer account  PRE 2 – The bug must be in the list og bugs | | |
| Postconditions | POST 1 – The bug’s status is changed to the new one  POST 2 – This change must be visible in the bugs list | | |
| Normal flow | 1. Change status   1. Programmer views the list of bugs  2. Programmers selects a bug from the list and asks to change its status  3. BA opens new window where current status is showed and new status is a select box with all possible statuses „ Unresolved” , „In Progress”, „Finished”  4. Programmer selects desired field and confirms change of status  5. BA confirms change of status  6. BA changes the status of bug in the bugs list | | |
| Alternative flow | 1. Cancel change of status   4. Programmer cancel the proccess of changing the status  5. BA closes the window and cancel the proccess | | |
| Exceptions | - | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Id and name | UC-3: Filtering bugs by status | | |
| Primary actor | Programmer | Secondary Actors | Bug Application |
| Description | A Programmer accesses the Bug Application, views the list of bugs and selects to filter them by status. Then he will select the status by which the bugs will be filtered and then he will get a list with the coresponding bugs. | | |
| Trigger | A Programmer indicates that he wants to filter the bugs by status. | | |
| Preconditions | PRE 1 – Programmer must be logged in, in a programmer account | | |
| Postconditions | POST 1 – All bugs with the given status must be present in the filtered list | | |
| Normal flow | 1. Filter list of bugs   1. Programmer views the list of bugs  2. Programmer selects that he wants to filter them  3. BA display a window from where the user can select the wanted status to filter the bugs  4. Programmer selects the desired status from a select box and confirms the filtering proccess  5. BA filter the bugs by the given status and display them to the programmer | | |
| Alternative flow | 1.1  4.Programmer selects the desired status from a select box and cancel the filtering proccess  5. BA cancel the proccess | | |
| Exceptions | 1. The list of filtered bugs is empty   BA will display a message informing the programmer that there are no bugs with the given status | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Id and name | UC-4: Filtering bugs by risk level | | |
| Primary actor | Programmer | Secondary Actors | Programmer |
| Description | A Programmer accesses the Bug Application, views the list of bugs and selects to filter them by risk level. Then he will select the risk level by which the bugs will be filtered and then he will get a list with the coresponding bugs. | | |
| Trigger | A Programmer indicates that he wants to filter the bugs by risk level. | | |
| Preconditions | PRE 1 – Programmer must be logged in, in a programmer account | | |
| Postconditions | POST 1 – All bugs with the given risk level must be present in the filtered list | | |
| Normal flow | 1. Filter list of bugs by risk level   1. Programmer views the list of bugs  2. Programmer selects that he wants to filter them by risk level  3. BA display a window from where the user can select the wanted risk level to filter the bugs  4. Programmer selects the desired risk level from a select box and confirms the filtering proccess  5. BA filter the bugs by the given risk level and display them to the programmer | | |
| Alternative flow | 1.1  4. Programmer selects the desired risk level from a select box and cancel the filtering proccess  5. BA cancel the proccess | | |
| Exceptions | 1. The list of filtered bugs is empty 2. BA will display a message informing the programmer that there are no bugs with the given risk level | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Id and name | UC – 5: Bugs list visualization | | |
| Primary actor | Tester | Secondary Actors | BA |
| Description | The tester wants to visualize the list of all bugs | | |
| Trigger | The tester logs in the BA | | |
| Preconditions | PRE - 1: Tester is logged in | | |
| Postconditions | POST - 1: The full list of bugs must be displayed | | |
| Normal flow | * 1. See list of bugs   1. Tester opens BA and log in  2. BA opens up and display main window with the list of bugs | | |
| Alternative flow | - | | |
| Exceptions | 1.0 E1 The log in data is not correct  1. BA will display a message informing the user that, the given log in data is not correct. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Id and name | UC – 6: Bugs list visualization | | |
| Primary actor | Programmer | Secondary Actors | BA |
| Description | The Programmer wants to visualize the list of all bugs | | |
| Trigger | The t Programmer logs in the BA | | |
| Preconditions | PRE - 1: Programmer is logged in | | |
| Postconditions | POST - 1: The full list of bugs must be displayed | | |
| Normal flow | * 1. See list of bugs   1. Programmer opens BA and log in  2. BA opens up and display main window with the list of bugs | | |
| Alternative flow | - | | |
| Exceptions | 1. E1 The log in data is not correct 2. BA will display a message informing the user that, the given log in data is not correct. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Id and name | UC-7: Authentification | | |
| Primary actor | Programmer | Secondary Actors | BA |
| Description | The programmer wants to open the Bug Apllication so it has to login first | | |
| Trigger | Open BA | | |
| Preconditions | - | | |
| Postconditions | POST 1: If log in data is valid then the app must open in the coresponding account  POST2: If log in data is NOT valid a message informing the user about this must pop up. | | |
| Normal flow | * 1. Logging in   1. BA displays the log in window  2. Programmer fills in the fields with his log in data and proceds the log in  3. BA verify the input data and opens the app in the coresponding account. | | |
| Alternative flow | Saved logging in   1. BA displays the log in window but the data is already filled in   2. Programmer fills in the fields with his log in data and proceds the log in  3. BA verify the input data and opens the app in the  coresponding account. | | |
| Exceptions | * 1. E1 Invalid log in data   A message pop up informing the user that the data in invalid. | | |