|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-01: Login | | |
| **Primary actor** | Employee | **Secondary actors** | Employee Database System |
| **Description** | An employee accesses the Bug Spotting application and he’s prompted by a login tab which asks for his credentials. | | |
| **Trigger** | When an employee clocks in at work. | | |
| **Preconditions** | PRE-1. Employee is registered in the database. | | |
| **Postconditions** | POST-1. A new tab opens depending on employee’s role. | | |
| **Normal flow** | 1. **Successful login** 2. Employee is asked for credentials by BSS (username and password). 3. BSS opens a new tab if credentials are correct. | | |
| **Alternative flows** | **None** | | |
| **Exceptions** | **1.0.E1 Incorrect password**   * 1. If employee’s username is found in the database, but the given password doesn’t match he is prompted by an the BSS with an error and a try again message.   **1.1.E1 Employee not found**  1. If employee’s username is not found in the database he is prompted by the BSS with an employee not found error. | | |

|  |  |
| --- | --- |
| Method-level traces | bugspotting.controllers.LoginController#login  bugspotting.controllers.LoginController#lforgottenPassword |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-02: Register a bug | | |
| **Primary actor** | Tester | **Secondary actors** | Employee Database System |
| **Description** | A tester who uses BSS can insert a new bug into the system by providing an appropriate name and description. | | |
| **Trigger** | Tester spots a bug in production and wants to have it fixed by a programmer. | | |
| **Preconditions** | PRE-1: Tester is logged into BSS | | |
| **Postconditions** | POST-1: Bug is introduced into the system and can be viewed by all programmers currently connected in the BSS. | | |
| **Normal flow** | * 1. **Register a bug**      + 1. BSS provides the tester with a tab which allows him to input the bug’s title and description and a button to submit the bug.        2. The programmers are alerted that a new bug was found that needs fixing. | | |
| **Alternative flows** | **None** | | |
| **Exceptions** | 2.0.E1 Bug description not long enough (min. 20 characters). | | |

|  |  |
| --- | --- |
| Method-level traces | bugspotting.controllers.TesterController#registerBug |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-03: View bugs | | |
| **Primary actor** | Programmer | **Secondary actors** | None |
| **Description** | A programmer can see a list with all bugs that are unresolved. | | |
| **Trigger** | Programmer logs in the BSS. | | |
| **Preconditions** | PRE-1: Programmer is logged into the BSS. | | |
| **Postconditions** | None | | |
| **Normal flow** | **3.0 View list of bugs**  1. Programmer is prompted by a list of unresolved bugs after successfully logging in. | | |
| **Alternative flows** | None | | |
| **Exceptions** | 3.0.E1 If there are no bugs in the system the list is replaced by a check icon. | | |

|  |  |
| --- | --- |
| Method-level traces | bugspotting.controllers.ProgrammerController#initializeBugs |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-04: Fix bug | | |
| **Primary actor** | Programmer | **Secondary actors** | Employee Database System |
| **Description** | A programmer marks a bug as fixed. | | |
| **Trigger** | Programmer indicates that he fixed a bug. | | |
| **Preconditions** | PRE-1: Programmer is logged into BSS.  PRE-2: A bug is selected from the list. | | |
| **Postconditions** | POST-1: Other programmers are informed that the bug was fixed.  POST-2: Bug disappears from every programmer’s list. | | |
| **Normal flow** | **1.0 Fixing a bug**   * 1. Programmer selects a bug from the list.   2. A fix bug button appears on screen.   3. Programmer presses the button.   4. Bug disappears from list for every programmer.   5. Programmer is prompted by a confirmation of fixing the bug. | | |
| **Alternative flows** | None | | |
| **Exceptions** | 4.0.E1 There are no bugs in the system, so the programmer can’t fix any bug.  4.0.E2 Two programmers fix the same bug at the same time. | | |

|  |  |
| --- | --- |
| Method-level traces | bugspotting.controllers.ProgrammerController#fixBug  bugspotting.controllers.ProgrammerController#initializeBugs |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID and name** | UC-05: Manage app | | |
| **Primary actor** | Admin | **Secondary actors** | Employee Database System |
| **Description** | Admin can manage employees of BSS. | | |
| **Trigger** | Admin requests to make changes to the system. | | |
| **Preconditions** | PRE-1: Admin is logged into the BSS | | |
| **Postconditions** | POST-1.0: Application state can change if the admin makes any changes in the employee section.  POST-1.1: Application state stays the same if admin didn’t make any changes. | | |
| **Normal flow** | **5.0 Manage application**   1. Admin logs in the BSS. 2. Admin can view/modify/add/delete employees. | | |
| **Alternative flows** | None | | |
| **Exceptions** | None | | |

|  |  |
| --- | --- |
| Method-level traces | bugspotting.controllers.AdminController#initializeEmployees  bugspotting.controllers.AdminController#addEmployee  bugspotting.controllers.AdminController#modifyEmployee  bugspotting.controllers.AdminController#deleteEmployee |

**Iterations**

1. **Iteration 1**
   1. UC-01: Login
   2. UC-05: Manage App

**2. Iteration 2**

1. UC-02: Register a bug
2. UC-03: View bugs

**3. Iteration 3**

**1.** UC-04: Fix bug