**Use Case Document: Bug Tracking System**

# **Use Case 1: Create Bug**

**Description:** The user creates a new bug for a specific project.

**Actors:** User/Admin

**Precondition:** The user is authenticated and authorized to create bugs.

**Basic Flow:**

1. The actor selects the "Create Bug" option from the UI.
2. The system displays a form for the user to input the bug details such as name, description, project, and status.
3. The actor enters the bug details and submits the form.
4. The system creates a new bug with the provided details and stores it in the database.
5. The system sends a success message to the user.

**Alternate Flows:**

* A1: The actor provides invalid input. The system displays an error message and prompts the user to correct the input.

# **Use Case 2: Read Bug**

**Description:** The actor views the details of a specific bug.

**Actors:** User/Admin

**Precondition:** The user is authenticated and authorized to read bugs.

**Basic Flow:**

1. The actor selects the "View Bug" option from the UI.
2. The system displays a list of bugs with their names and projects.
3. The actor selects a bug from the list.
4. The system retrieves the details of the selected bug from the database.
5. The system displays the bug details to the user.

**Alternate Flows:**

* A1: The actor selects an invalid bug. The system displays an error message and prompts the user to select a valid bug.

# **Use Case 3: Update Bug**

**Description:** The actor updates the details of a specific bug.

**Actors:** User/Admin

**Precondition:** The user is authenticated and authorized to update bugs.

**Basic Flow:**

1. The actor selects the "Update Bug" option from the UI.
2. The system displays a list of bugs with their names and projects.
3. The actor selects a bug from the list.
4. The system retrieves the details of the selected bug from the database and displays them in a form.
5. The actor modifies the bug details in the form and submits it.
6. The system updates the bug details in the database.
7. The system sends a success message to the user.

**Alternate Flows:**

* A1: The actor provides invalid input. The system displays an error message and prompts the user to correct the input.
* A2: The actor selects an invalid bug. The system displays an error message and prompts the user to select a valid bug.

# **Use Case 4: Delete Bug**

**Description:** The actor deletes a specific bug.

**Actors:** Admin

**Precondition:** The actor is authenticated and authorized to delete bugs.

**Basic Flow:**

1. The Admin selects the "Delete Bug" option from the UI.
2. The system displays a list of bugs with their names and projects.
3. The Admin selects a bug from the list.
4. The system retrieves the details of the selected bug from the database and prompts the user to confirm the deletion.
5. The Admin confirms the deletion.
6. The system deletes the bug from the database.
7. The system sends a success message to the user.

**Alternate Flows:**

* A1: The Admin selects an invalid bug. The system displays an error message and prompts the user to select a valid bug.

# **Use Case 5: Search Bug**

**Description:** The actor searches for bugs based on name, project, and status.

**Actors:** User, Admin

**Precondition:** The user is authenticated and authorized to search bugs.

**Basic Flow:**

1. The actor selects the "Search Bug" option from the UI.
2. The system displays a search form with fields for bug name, project, and status.
3. The actor enters the search criteria and submits the form.
4. The system retrieves the bugs that match the search criteria from the database.
5. The system displays the list of matching bugs to the user.

**Alternate Flows:**

* A1: The actor provides invalid input. The system displays an error message and prompts the user to correct the input.
* A2: No bugs are found that match the search criteria. The system displays a message indicating that no bugs were found.

# **Use Case 6: View Bugs by User**

**Description:** The user views the list of bugs raised by him/her.

**Actors:** User

**Precondition:** The user is authenticated and authorized to view bugs.

**Basic Flow:**

1. The user selects the "My Bugs" option from the UI.
2. The system retrieves the list of bugs that were raised by the user from the database.
3. The system displays the list of bugs to the user.

**Alternate Flows:**

* A1: No bugs were raised by the user. The system displays a message indicating that no bugs were found.

# **Use Case 7: View Bugs by Project**

**Description:** The user views the list of bugs raised in a specific project.

**Actors:** User

**Precondition:** The user is authenticated and authorized to view bugs.

**Basic Flow:**

1. The user selects the "Project Bugs" option from the UI.
2. The system displays a list of projects.
3. The user selects a project from the list.
4. The system retrieves the list of bugs that were raised in the selected project from the database.
5. The system displays the list of bugs to the user.

**Alternate Flows:**

* A1: The user selects an invalid project. The system displays an error message and prompts the user to select a valid project.
* A2: No bugs were raised in the selected project. The system displays a message indicating that no bugs were found.