

USE CASE DIAGRAM DESCRIPTION

BUGSEEKER

Introduction

The BugSeeker is a software tool designed to track and manage software defects or bugs throughout the software development lifecycle. This document outlines the use cases of the Bug Tracking Application to provide a clear understanding of its functionalities and features.

Use Case Descriptions

1. Create Bug

Purpose: This use case allows a Developer or Tester to create a new bug report in the bug tracker application. They provide necessary details such as bug title, description, severity, and associated project or module.

Actors: Developer, Tester

Preconditions: The user is logged into the bug tracker application.

Flow of Events:

1. The user selects the option to create a new bug report.
2. The system presents a form to the user, requesting bug details such as title, description, severity, and project/module association.
3. The user fills in the required information and submits the bug report.
4. The system validates the input and creates a new bug report in the bug tracking system.
5. The system notifies the user that the bug report has been successfully created.

Postconditions: A new bug report is created and added to the bug tracking system.

2. View Bugs

Purpose: This use case enables a Tester or Admin to view a list of existing bugs in the bug tracker application. They can filter and search for bugs based on various criteria such as project, status, severity, or assigned developer.

Actors: Tester, Admin , Developer

Preconditions: The user is logged into the bug tracker application.

Flow of Events

- 1.The user selects the option to view the list of bugs.
2. The system retrieves the bug data from the bug tracking system.
3. The user may apply filters or search criteria to narrow down the list of bugs.
4. The system presents the filtered or searched bug list to the user.
5. The user can view the details of a specific bug by selecting it from the list.

Postconditions: The user can see a list of bugs matching the specified criteria.

3.Assign Bug

Purpose: This use case allows an Admin to assign a bug report to a specific Developer for resolution. The Admin selects the bug and assigns it to the appropriate Developer, who becomes responsible for fixing the bug..

Actors: Admin

Preconditions: The user is logged into the bug tracker application as an Admin.

Flow of Events:

1. The Admin selects the option to assign a bug to a Developer.
2. The system presents a list of available bugs that are unassigned or require reassignment.
3. The Admin selects a bug from the list and chooses the Developer to assign the bug to.
4. The system updates the bug record, assigning it to the selected Developer.
5. The system notifies the Admin and the Developer about the bug assignment..

Postconditions: The selected bug is assigned to the designated Developer..

4.Update Bug

Purpose: This use case enables a Developer or Admin to update the details of a bug report. They can modify information such as bug status, priority, assigned developer, or additional comments to track the progress or provide further details.

Actors: Developer , Admin.

Preconditions: The user is logged into the bug tracker application

Flow of Events:

1. The user selects the option to update a bug report.
2. The system retrieves the bug details for the selected bug.
3. The user modifies the necessary fields, such as status, priority, assigned developer, or adds comments.
4. The user submits the updated bug report.
 5. The system validates the changes and updates the bug record in the bug tracking system.
 6. The system notifies the user that the bug report has been successfully updated.

Postconditions: The bug report is updated with the new information.

5.Resolve Bug

Purpose: This use case allows a Developer to mark a bug as resolved once they have fixed the underlying issue. They update the bug status and may provide resolution details or comments.

Actors: Developer

Preconditions: The user is logged into the bug tracker application as a Developer.

Flow of Events:

1. The Developer selects a bug that they have resolved.
2. The system retrieves the bug details.
3. The Developer updates the bug status to "Resolved" and provides resolution details

or comments if necessary.

4. The Developer submits the resolved bug report.
5. The system validates the changes and updates the bug record in the bug tracking system.
6. The system notifies the Developer and relevant stakeholders that the bug has been resolved.

Postconditions: The bug status is changed to "Resolved" and the bug is ready for testing and verification.

6. Close Bug

Purpose: This use case allows an Admin to close a bug report once it has been resolved and verified. They mark the bug as closed and provide any final comments or notes.

Actors: Admin

Preconditions: The user is logged into the bug tracker application as an Admin.

Flow of Events:

1. The Admin selects a resolved bug that has been verified.
2. The system retrieves the bug details.
3. The Admin updates the bug status to "Closed" and adds any final comments or notes.
4. The Admin submits the closed bug report.
5. The system validates the changes and updates the bug record in the bug tracking system.
6. The system notifies the Admin and relevant stakeholders that the bug has been closed.

Postconditions: The bug status is changed to "Closed" and the bug is no longer actively tracked.