

Software Design Specification

for

Bug Squasher

Overview

Bug Squasher is an internal service that tracks software bugs that are initiated by users and how they are dispositioned by the software maintenance group. This specification will focus on the functionality and layout of the system. This specification is intended for the development team.

Specifications

Technology

This project will implement the technology full-stack commonly referred to as the MERN stack.

- **M**ongoDB - this will be the web accessible database
- **E**xpress.js - this will be used to setup the RESTful API
- **R**ead.js - this will be used to develop the client/frontend Single Page Application (SPA) to which the user will interface
- **N**ode.js - this will be the JavaScript runtime backend of the SPA

Other libraries may be used as necessary to complete this project.

Version control will be handled by creating a remote Git Repository to which developers can contribute and pull updated versions.

Testing will primarily be use cases. Postman will be implemented to test out the RESTful API calls.

User Environment

The primary users will be using a laptop or desktop machine. The project will not be optimized for mobile use.

Planned Deployment Strategy

The deployment strategy for this project is to deploy the entire project to heroku.com.

Functional Description

The application will handle two types of accounts: general user and maintenance (development) user. Each user will be required to log in from the Home Page.

General User

The general user is the primary user of this application. This is the user who will report bugs to be fixed by the maintenance user. Their view will only display the bugs that they have reported. The general user will have the following options available when interacting with bug reports:

- Create new bug report
- Update existing bug report (owned by the user)
- Delete existing bug report (owned by the user)

The user will be able to view their previously reported bugs and their status.

Create New Bug Report

When the general user wants to create a new bug report, they will click the button labelled "Report New Bug". A modal will appear with a form available for the user to complete. Once the user has completed the required fields they will click the "Submit Bug Report" button to submit the bug report or will click the "Cancel" button to cancel the process.

Update Existing Bug Report

The user can update an existing bug report by clicking the "Edit" button next to the target bug report in their list of bug reports. A modal similar to the one from the Create New Bug Report section will appear with similar fields. These fields will be populated with the current data from the bug report. The user may edit any of the fields not greyed out and then click the "Resubmit Bug Report" button to complete the editing process or the "Cancel" button to cancel the entire process. This process is not allowed for bug reports that have been closed out by a maintenance user.

Delete Existing Bug Report

The user can delete an existing bug report by clicking the "Delete" button next to the target bug report in their list of bug reports. A modal will appear asking them if they are sure they want to proceed with the deletion. The user can complete the delete function by clicking the "Delete" button or may cancel the process by clicking the "Cancel" button. This process is not allowed for bug reports that have been closed out by a maintenance user.

Maintenance (Development) User

The maintenance user will handle receiving incoming bug reports from the general user and will status the disposition of those identified bug reports. The maintenance user will have the following options available when interacting with bug reports:

- Assign themselves to the bug report

- Update the status of the bug report

Assign Bug Report

A maintenance user may assign themselves to a currently unassigned bug report. The user will click the “Assign” button next to the target bug report. A modal will appear asking the user if they wish to continue. The user may complete the process by clicking on the “Assign Me” button or may cancel the process by clicking the “Cancel” button.

Update Assigned Bug Report Status

A maintenance user may update the status of a bug report that is assigned to them. The user will click the “Update Status” button next to the target bug report. Several options will be available:

- Review - the bug is being reviewed
- In-process - the bug is in process of being fixed
- Closed - the bug has been closed

The user may complete this process by clicking the “Update Status” button or cancel the process by clicking the “Cancel” button.

Milestones

I. Develop Backend - (1 week)

- Create a MongoDB Cluster
- Integrate Node.js and Express.js with the MongoDB Cluster
- Develop data models using Mongoose.DB
- Develop RESTful API with Express.js

II. Develop Frontend - (1 week)

- Develop basic user interface with React.js.
 - Component development
 - Develop state management with Redux.js
 - Develop http protocols to the backend using axios.js

III. Testing - (1 week)

Perform basic use case and connectivity testing on local host and then after heroku.com deployment.