BUG TRACKER SRS

**Introduction**

This document details the project plan for the development of “BugTracker”.

It is intended for programmers and testers working in software development. This plan will include a summary of:

* how the system will function
* the scope of the project from the development viewpoint
* the technology used to develop the project, and
* the metrics used to determine the project’s progress
* Overall Description

For software development companies using a bug tracker is an everyday task. Bug tracking is the process of tracking bugs or errors in issues or tickets. Large systems can have hundreds or thousands of defects. Each one needs to be evaluated, monitored, and prioritized for debugging. In some cases, bugs may need to be tracked over a long period of time.

**Customers**

The customers will be enterprise companies and any developers team that need to create a new piece of software.

**Platform**

The bug tracker will be a web application developed in C# using the framework ASP.NET MVC.

To host the website, we’ll be using Amazon Web Services S3, EC2 and RSD via the service Elastic Beanstalk.

Authentication will be done with Asp.Net Core IdentityFramework.

**Account Types**

Admin: will be able to access all functionality of the app, including:

* Create, Edit and delete any projects.
* Managing responsibility and access to any projects.
* View and monitor any project
* Update user type

Standard user will have access to some functionality of the app, including:

* Create projects
* Edit and delete projects in which he is a project manager
* Managing responsibility and access projects in which he is a project manager
* View and monitor projects in which he is a member

**Project Roles**

A user can be assigned to a project with any of these roles:

Project Manager:

* Are allowed to create new projects.
* Are allowed to edit and delete projects created by them.
* Are allowed to manage responsibility and access to projects created by them.

Reporter:

* Are allowed to create, edit and delete tickets
* Are allowed to add comments on tickets
* Are allowed to attach files on tickets
* Are not allowed to change the status of the ticket

Developer:

* Are allowed to update the ticket status from To Do to In Progress and from In Progress to In Review
* Are allowed to attach and delete files to tickets
* Are allowed to add comments on tickets
* Are not allowed to create or delete tickets
* Are not allowed to update any ticket property other than the status
* Are not allowed to close bug

Reviewer:

* Are allowed to update the ticket status from In Review to Done and from In Review to In Progress
* Are allowed to add comments on tickets
* Are allowed to attach files to tickets
* Are not allowed to create and delete tickets
* Are not allowed to update any ticket property other than the status

**System Features**

**Functional Requirements**

* Ability to authenticate a user
* Ability to create, delete and edit projects
* Ability to create, delete and edit tickets
* Ability to add and revoke access to a project
* Ability to see a list of users assigned to a project
* Ticket status: To Do, In Progress, In Review, Done
* Ticket Priority: Low, Medium and High
* Ability to browse the tickets for a project: ability to do sorting and search by keyword
* Ability to attach files to tickets
* Ability to assign a user to a ticket
* Ability to add comments to a ticket
* Ability to track any change/update in the project and in a ticket (timeline)

**User Interfaces**

* Front-end & Back-end software: ASP.Net MVC
* Database software: Azure database for MySQL
* Authentication service: Auth0

**Performance Requirements**

* The application should load and be usable within 3 seconds
* The database should be normalized to prevent redundant data and improve performance

**Software Quality Attributes**

* Availability: Because this application is critical to business communication, we will have a goal of 99.99% availability.
* Correctness: The application should never allow anyone to read messages or discussions not intended for that person.
* Maintainability: The application should use continuous integration so that features and bug fixes can be deployed quickly without downtime.
* Usability: The interface should be easy to learn without a tutorial and allow users to accomplish their goals without errors.