SOFTWARE REQUIREMENTS SPECIFICATION

for

B-Track

Version 1.0

Prepared by Albert Negura

June 20, 2020

Contents

Section 1

Document Overview

This document outlines the SRS including the complete project specifications, requirements, complete database specifications and development plan for B-Track, a Bug Tracker or Issue Tracker or Ticket system web app developed in $C\#$, employing the $.NET$ framework, MVC pattern and SQL Server Database technology. This document was written and compiled in LaTeX.

Section 2 provides an overview of the project, the indended stakeholders and the assumptions and constraints used in the development of this project in a way meant to mirror the method applied by various corporations for similar software development projects.

Section 3 provides a list, description and purpose for all the functional and non-functional requirements relating the end user requirements, user interface, performance, security and software quality.

Section 4 provides a complete classification and ranking of all the functional requirements while also providing the dependencies for each of the functional requirements.

Section 5 provides the corresponding Database design, basic UI design and UML design, together with other explanatory diagrams.

The necessary planning and other helpful material not directly corresponding to the software requirements specifications will be featured in the Appendix.

Section 2

Project Overview

2.1 Project Introduction

B-Track is a Bug Tracker, Issue Tracker or Ticket system meant to mirror the functionality of similar modern day system used by various corporations throughout the world. Such a system is used to maintain accurate records of development processes on a per-project basis or to help with customer or client tech support depending on the desired use case.

2.2 Stakeholders

This project does not have any direct clients and was created with the goal to act as a suitable programming exercise and example of good software development practices.

2.3 Assumptions and Constraints

This project is meant to act as a Web-based service application developed using the JetBrains Rider 2020 IDE Professional edition and the JetBrains DataGrip 2020 database IDE, available through the JetBrains student package. It also employs the Model-View Controller design pattern at its core. The web pages are to be built with HTML5 and Bootstrap. The page should be hosted online using AWS, with the link being provided on GitHub.

Section 3

Requirements

This section describes all the functional and non-functional requirements and / or features for the Bug Tracking software. The project will only be considered complete upon having all the requirements outlined in this section met.

3.1 Functional Requirements

3.1.1 End User Requirements

End users must be able to perform the following actions:

3.1.2 User Interface Requirements

3.2 Non-functional Requirements

3.2.1 Performance Requirements

3.2.2 Security Requirements

3.2.3 Software Quality Attributes

	Description	Purpose
1	Register as a user / Login	Secure system access
2	Assign/unassign users to/from roles	Role-based security
3	Create projects	Organization of resources / project managers or admin only
4	Assisng/unassign users to/from projects	Organization of resources / project managers or admin only
5	Create tickets	Log software issue instance
6	Assign tickets to user	Assign responsible developer to ticket and enforce accountability
7	Edit submitted tickets	Make modifications to existing tickets
8	Create ticket comments	Add progress comments and other important information to ticket history
9	Create ticket attachments	Add helpful visuals and other documentation to the ticket history
10	List comments, attachments per ticket	Organization of resources
11	List history of changes to ticket	Very important view for the developer on the project
12	List tickets by owner	Provide easy tracking of tickets logged by a particular user
13	List tickets by assignment	Provide easy tracking of tickets logged by a particular developer
14	List tickets by project	Organization of resources
15	Filter ticket lists by ticket type, priority and status	Ease of use and access
16	Filter ticket lists by creation date/time (i.e. all tickets after indicated time/date)	Ease of use and access
17	Sort ticket list by title, owner, assignment, creation of recent update date/time, ticket type, priority, status and project	Ease of use and access
18	Full text search of all relevant fields	Ease of use and access

Section 4

Classification of Functional Requirements

4.1 End User Requirements

4.2 User Interface Requirements

Section 5

Developer Add-ons

5.1 Database Design

5.2 User Interface Design

5.3 UML Design