Software Engineering Software Requirements Specification (SRS) Document

Bug Tracker

September 20th 2020

1.0.0

By Vishon Singh

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1.0	Vishon Singh	Basic start of the application, added authentication first to the application in the backend and authentication checks in the front end. All basic authentication has been setup and is now functioning.	09/25/2020
0.2.0	Vishon Singh	Basic UI Layout has started. Dashboard, My Projects, My Tickets, Navigation, and Side Bar have been built and everything is responsive. Also added the logout functionality.	09/26/2020
0.3.0	Vishon Singh	Basic UI Layout is ready, will add some extra minor UI with some changes later on, but the base Layout is done. Now Functionality setup will begin.	09/27/2020
0.4.0	Vishon Singh	Project Functionality and Components have been built and are working on both front and backend. Now the app is able to create and edit projects. Also, now able to assign users to the projects by choosing through all users registered.	09/30/2020
0.4.0	Vishon Singh	Started building Ticket System functionality. So far, the app is able to add tickets, assign them to specific users on the project team, it all shows up for the specific user the ticket is assigned to. Added authorization checks to make sure only Admin/Owners can add/remove users and roles.	10/1/2020
0.5.0	Vishon Singh	Ticket Functionality completed. Users can now add tickets, modify tickets, assign to other users, add comments to tickets, and add attachments. Also, there is a ticket history system to keep track of changes. Starting notification system development now.	10/3/2020
0.6.0	Vishon Singh	Notification functionality is complete, UI is built on the Front End, Backend functionality is complete, tested on multiple user accounts to see if notifications are sent and received, all is working. Adding delete functionality now, then doing bug fixes and minor feature additions.	10/5/2020
0.6.1	Vishon Singh	Fixed server-side issues that could possibly crash the server. Sanitized my inputs on the frontend, added delete functionality, added description to tickets, and fixed some minor bugs. Added some error handling. Setting up charts now and then a support page to receive issues on this app.	10/5/2020

Version	Primary Author(s)	Description of Version	Date Completed
0.7.2	Vishon Singh	Charts have been set up and hooked up to the backend database info. Sanitized more inputs, fixed issue with body scrolling when notification modal was open. Body will no longer scroll when notification modal is open.	10/5/2020
0.7.3	Vishon Singh	Added proxies to fix frontend to backend connection issues during development. Now instead of having to call 'http://localhost/APIROUTEHERE I can do /APIROUTEHERE	10/6/2020
1.0.0	Vishon Singh	Production version is live and had many routing issues. Spent 5 hours fixing routing issues. Also, upon logout 2 JSON error messages would appear due to a useEffect dependency, fixed it by adding a conditional to not run code if that dependency is undefined.	10/6/2020
1.0.1	Vishon Singh	Fixed issue where it wouldn’t let me add users to projects, this was due to a change I made in one of the select forms. Fixed another issue where it wouldn’t delete all personnel related to a project upon project deletion, has been fixed and database has been cleaned up.	10/6/2020
1.1.2	Vishon Singh	Fixed issue where users could bypass front end max length checks and save long spam inputs to database, now the backend has length checks. Also fixed notifications modal glitching on mobile screens. Added support page and setup the layout. Added hover effects to dashboard items	10/6/2020

Table of Contents

1. Introduction

1.1 Purpose: Keep track of all current bugs and issues for current projects being worked on. Upon logging in, users will be presented with a basic dashboard showing active issue tickets. They can also view a projects page where they can view all their current projects being worked on or create new projects. Upon clicking a project, the user will see Details for Project, Users Assigned to this Project, and Tickets for this Project. Each will be grouped in their own container group and will have ‘Add’ buttons to add new entries/users. Each entry for any container group for tickets will have a “More Details” link to view full details of the ticket. Inside each ticket will have more information, like comments, priority, open/closed status, ticket history, and more. This project will be responsive on all devices.

1.2 Document conventions: None, this will be kept basic and simple to follow to not overcomplicate things

1.3 Intended audience: This whole project is being developed and managed by a single developer, Vishon Singh. This document will be made publicly available for anyone to see and the project itself will be open sourced.

1.4 Scope: The goal of this software is to build a tracking system for all future projects, where all bugs and issues will be added into this system for each individual project.

1.5 References: None.

2. General Description

2.1 Product perspective: A very basic Issue Tracker was created in the past by me. I am building this new one to be used with all my future projects and I will use this Bug Tracker project to keep track of any bugs I come across while building out this application. This application is meant to utilize my skills in TypeScript, React, NodeJS, Express, MongoDB, and Mongoose, along with Sass (SCSS) and Material UI.

2.2 Product features: This application will include a login system using JWT Tokens and HTTP-Only Cookies and will use React in the Front End utilizing Hooks and Context API to manage basic logged in user data. Toward the later stage of the project before final release, setup demo functionality so people can demo this application without having to register. It will use Material UI to manage how everything is displayed on the page and help with responsiveness so this can be used on all devices. The issue tracker will start off basic and will be grown from there once core functionality is down. Core functionality will include, User Authentication, Application Dashboard showing Projects and Tickets, Current Projects Tab, Tickets Assigned to Current Logged in User, Simple User Profile.

2.3 User class and characteristics: Software Developers, Web Developers, Managers of Development Teams, Workers in IT fields, QA Testers, and more. This is a general bug/issue tracker that can be utilized by anyone who handles bugs/issues on a daily basis in any field.

2.4 Operating environment: This project is being developed on Windows 10 and using VSCode.

2.5 Constraints: Main constraint to watch out for is Database Schema Design, make sure it is a flexible design and be scaled to grow with the application and have no issues doing so. Make sure Front End is designed well, utilize the full power and benefits of React Hooks, if not, development will become much harder. Create error handling in back and front end.

2.6 Assumptions and dependencies: JWT + HTTP-Only tokens will be used for Authentication, TypeScript will be standard in this project, React hooks must make use of all hooks including some advanced hooks. Material UI must be used to keep a clean and flexible design on all devices. Sass (SCSS) will be used for styling along with Material UI. Will be using React-router, possibly Axios or we may stick with Fetch for the Front End, will decide more into development and stick with one. Basic stack will be MERN Stack + TypeScript. Sass, React, NodeJS, Express, MongoDB, Mongoose, TypeScript, and Material UI. Some dependencies will include BCrypt, Cookie Parsing, and more.

3. System Requirements

3.1 Functional requirements

- Follow MVC Software Design Pattern for this project.

- Setup Routes, Controllers, and Handling Requests (GET/POST/etc)

- User Authentication using JWT and HTTP-Only cookies. Register, Login, and Demo Guest Login. General user roles may be added later in development for the roles of the actual website. Start the Schema Model, begin with the User Schema to finish the User Authentication.

- Figure out a clean and flexible Schema Model design that will scale with the application. Make sure to avoid unbound arrays.

- Start figuring out and designing the small functionalities for this application. Creating Projects > Creating Issue Tickets > Adding Users to Project > Edit/Delete Functionality and more.

4.External Interface Requirements

4.1 User Interfaces

Interfaces will be built using Material UI and combining that with my CSS skills. The main goal is to make this responsive on all devices, I will be taking a Mobile First approach to keep track of how the app looks on mobile mainly, then on desktop/other devices. The interface will be kept very simple and basic to help with not confusing the users and to also make this application easier to get working on Mobile, Tablet, and Desktop screens. The main goal is to avoid tables in the front-end design because tables can be tricky to make responsive, I may use a card approach to separate entries into cards instead of table data.

4.2 Hardware Interfaces

This is a Web App and will be supported on all hardware that has a browser. No special hardware is needed to build this web application

4.3 Communications Interfaces

No communication standards will be utilized in building this application.

4.4 Software Interfaces

This will utilize the MERN Stack + TypeScript. The Front End will be React, with Material UI, and Sass (SCSS). The Back End will be NodeJS, Express, MongoDB, and Mongoose with JWT Tokens and HTTP-Only cookies being utilized for Authentication. This will be built in VSCode with ESLint and Prettier extensions.

5. Non-Functional Requirements

5.1 Performance requirements

The main performance benefits will come from utilizing React Hooks correctly in the Front End and designing the schema in a flexible but scalable way, while avoiding unbound arrays.

5.2 Safety requirements

Will be using BCrypt encryption to keep user passwords safer, I know the risk of JWT + HTTP-Only tokens, but I will run this on HTTPS, although that won't get rid of XSS risks the risk will still be low. I will also be adding some error handling and security checks inside the code during development.

5.3 Security requirements

Will be looking into more security during development.

UPDATE 09/26/2020: While building the authentication I have become more aware of possible risks of JWT + HTTP-Only methods. With HTTP-Only at least people won't be able to tamper with the cookie from client-side. Also added Same-Site: true to make sure outside sources can't easily tamper with the Cookie or JWT Data.

5.4 Software quality attributes

The main thing I am aiming for is flexibility and responsiveness on all devices for this application.

5.5 Other requirements

None as of yet, will continue during development. 9/20/2020 – Vishon Singh