SOFTWARE REQUIREMENTS SPECIFICATION

for

Issue Tracker

Version 1.0

Prepared by : Riley Wium

March 9, 2020

Contents

1 Introduction

1.1 Purpose

This is a portfolio project for the creation of a custom issue tracker. If the system is built robustly enough it can even be reworked into a bug tracker, project manager, or scrum board. The systems purpose is to allow for the management of tasks and sub-tasks making scheduling easier and gaining data on the time and effort it takes to complete certain tasks. The system should allow for the creation, manipulation, and completion of task objects. The task objects are to be stored on a database and accessed via website. The project will include some functionality which are superfluous to my personal use of the tool but will serve it as a portfolio piece. Information about the design and progress of the issue tracker will be stored in this document.

1.2 Project Scope

The project is created for personal use as a tool for organising my tasks during projects and is meant to be completed over the course of a couple weeks, with the deadline being the end of February 2020.
The project will mainly use C#, .NET, SQL, and the MVC design pattern. It will also likely incorporate Bootstrap and a third party encryption service.
The application will keep track of the task name, task feedback, difficulty rating of task, task descriptions, completion status, and the people assigned to the task. Users will be able to login with accounts assigned ID's and host projects, invite others, and be invited themselves. The user's can either be project owners or contributors, with the former allowing for more control over the project. The project will be focused on the Windows platform, but will be tested to see how it functions on iPad and Linux environments. All created data will persist in the database and only be accessible by the projects team via a browser.

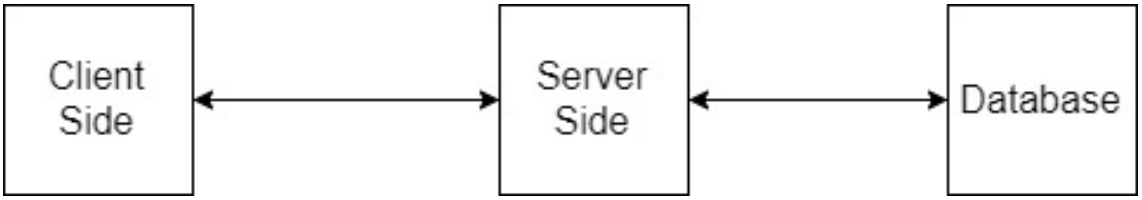

Figure 1.1: Issue Tracker System

2 Overall Description

2.1 Product Perspective

The project must be accessed via browser of which Chrome, Firefox, and Edge will be tested. Windows will be the main platform the project is developed for but Linux and Mac systems will be tested as well.

2.2 User Classes and Characteristics

The issue tracker has two types of users.

- Project Owners
- Project Contributors

Project contributors have the permission to add issues, assign users, complete tasks, edit descriptions, rate difficulty, and assign feedback.
Project owners have the permissions of contributors but can also add new contributors, delete the project, and rename the project.

2.3 Product Functions

This Issue Tracker should include the following functionality.

2.3.1 User Interfaces

Client Entry Screen: Covers main issue tracking functionality. The user can post a issue for permitted projects to a list of total list of issues. The list of issues can be filtered by completion status. A screen for adding issues will prompt the user for a name and additional info supplied.

Login Screen: Allows for users to access their account information and projects. Accompanied by a registration screen. The layout will be based on how the third party software functions.

Project Screen: Covers main project and team creation functionality. A project can be created or modified from this screen.

2.3.2 Synchronization Server

The server software will open a port on the network and will accept incoming connections with clients. Upon connection, the server will serve requested data to the client program. If the client program attempts to update a project, the server software will check the user's credentials to make sure that they have the rights to edit the projects. If they do, the server will update the database accordingly. If the data is modified on both server and client, the user will be prompted if they wish to push their changes.

2.3.3 Potential Functions

- email accessibility
- ticket dependency (ticket 2 can only be completed after ticket 1)
- email ticket creation
- easy to rework to bug tracker, project manager, or scrum board
- time tracker
- file attachments
- more user permission customization

3 System Features

The system will be a three-tiered design with a Server Application, a Database Engine, and Client Applications. The user will only interact with the system through client applications running on a PC. The general user interface will be forms with buttons and other familiar interactive tools. The user will use this interface to check-out a project, which will interact with the server application (via http) to retrieve the necessary data. The user will view and manipulate that data locally and then check-in the project, again through the server. The server will interact with the database to centrally store all the data for the system.

3.1 Description and Priority

The project has features that are main and also some are sub. Here is listed all the features necessary for this software.

Issue Board	
Purpose	Track, create, and manipulate issues for the project your working on
Priority	High
Input	Entered by the user. Includes: name, description, assignee, reporter, date reported, and comment
Processing	Validate input is of right type and format
Output	Recall previously stored information from database

Project Board	
Purpose	Track, create, and manipulate projects
Priority	High
Input	Input name to make a project. Input other username to invite to project
Processing	Validates data entered
Output	list of permitted projects

Login Page	
Purpose	Ensure account security in creation and login
Priority	Medium
Input	username and password to both create and login
Processing	checks if username is taken and if password is valid
Output	access to account

Synchronize Server	
Purpose	A server will open a port on the network and will accept connections with clients. Wait for incoming connections from a client computer
Priority	Low
Input	Inputs will come as predefined commands from the client computer
Processing	Server will follow the commands and output its result. If an invalid code is recieved the server will send an error message
Output	The server will send predefined codes and results to the client

3.2 Functional Requirements

Back-End - .NET framework, C# language.
Font-End - BootStrap. possibly ASP.NET?, ASP.NET MVC, React, React Redux, JavaScript.
Database - SQL possibly MongoDB.

4 Other Nonfunctional Requirements

4.1 Performance Requirements

When not accessing the server, a typical modern PC (like those found in the University of Utah EMCB 130 lab) running the client software should have no noticeable lag (i.e. less than 1/10 second) in retrieving data that is stored locally. When pulling data from the server, average response time to bring up information regarding a particular project should be no more than two seconds. The PC's stored data files should consume no more than 100 MB, regardless of the overall size of the database or the number of projects assigned to any given user. The PocketPC's stored data files should consume no more than 10 MB. The database is expected to grow proportionally to the number of Project and project items that have been created, and the amount of data entered for each.

4.2 Security Requirements

Only registered users can enter the website. The login functionality will be handled using third party software

4.3 Software Quality Attributes

Database, logical and also UI tests will be made for main functionality.

5 Bibliography

- Latex Template by Md. Yasmi Tohabar and Akaash
-