Shale Hausler

Project Documentation

Circle

Description automatically generated with medium confidence

# Executive Summary

Developing software is a complex process with teams of people working together. Each person on that team needs to be able to track where they are in the project and document what bugs still need to be fixed, features added, or tasks that still need to be done. The issue is that each person organizes this information in a different way, and not everyone has access to this information to be able to help other team members complete the project on time and one budget. A solution is needed to keep all of this information together in an organized way so that any team member can jump in any part of the project and know exactly what needs to be done

# Project Scope

Develop a software suite to track and organize the development of a software project. This suite will keep track of known bugs, and intended features, in a file format that is easy to manipulate outside of the software suite.

One unique aspect of this project is that files associated with the development project will be stored with the development project. This feature will always keep the documentation with the project and is not tied to this software suite

A second feature is the saved file format will not be proprietary. Sometimes a user wants to manipulate data in a way that the program does not currently allow. By keeping file formats universal, this will allow a user to import the data into other programs to manipulate the data in a way they need if those features are not yet allowed in the software suite.

External Files & Data

Files are stored wherever a person would like to. There is no security, or keeping people from editing the files directly.

Programming Language | C#.NET

Written mostly in C#

Project Classes

Classes within the project are used to abstract re-usable pieces of code. Classes are also used to group related values, known as properties. The project utilizes these classes:

### formColors

This part names each object in the form, and its color address. This is used to change from a Dark to a Light theme.

Summary

This is a windows-based application, that is a poorly implemented version of a text editor. Nothing like what I had envisioned in the beginning. The first attempts to create a cross platform program in QT with QML and C++ were a failure, and I abandoned that method to late in the course to make this one what it really should have been. There were some nice graphics in the other version, and I started with those as I thought they would be the hardest and left basic function implementation until part of the GUI was done so that I could tie it together and then add the extra things. I was very wrong and was never able to get the basic core implementation to work. I abandoned that and had to scrap all that project and start from scratch with only a few weeks left in the course. This has some of the basic features and requirements for the assignment, but is not a complete project and no where near where it should be.

# APPENDIX C (CLIENT INSTALLATION INSTRUCTIONS)

Client can double click the “setup” application, and the software will install and launch itself. It will also make its self-available in the Windows start menu.

# APPENDIX D (DEVELOPER SETUP INSTRUCTIONS)

Open the .sln file in Visual Studios.