Interim Report 2023

Text Editor

Interim Report

Table of Contents

[1. Introduction 3](#_Toc150863717)

[1.1 Plain text editors 3](#_Toc150863718)

[1.2 Rich Text Editors 4](#_Toc150863719)

[1.3 Objectives 5](#_Toc150863720)

[1.4 Language Considerations 6](#_Toc150863721)

[2. Requirements Analysis 6](#_Toc150863722)

[2.1 User Interface 6](#_Toc150863723)

[2.2 Requirements 6](#_Toc150863724)

[2.2.1 Mandatory 6](#_Toc150863725)

[2.2.2 Desired 6](#_Toc150863726)

[3. Project Plan 7](#_Toc150863727)

[3.1 Phases and Tasks 7](#_Toc150863728)

[4. Log 7](#_Toc150863729)

[Bibliography 8](#_Toc150863730)

# Introduction

A text editor is a software application which allows a user to create and edit text files. Plain text editors are used by developers to write programs, and include features such as syntax highlighting, code completion, support for debugging and embedded git. Rich text editors are used to create documents, and allow users to change font colour and size, as well as add images and pages. This report will give an overview of the goals and objectives of this project, including the problems that need solving and research required.

## Plain text editors

A screenshot of a computer

Description automatically generatedOne example of a text editor is vim [1]. Vim is a command line text editor, which is included on most Unix based systems

Figure : Vim

Vim is an editor that will take longer to get used to than other text editors, due to its lack of a mouse-driven interface, modal nature and the need to learn specific commands and shortcuts. However, many users find that the initial steep learning curve of vim pays off in increased productivity.

On the other end of the spectrum for plain text editors is Visual Studio Code (VSCode) [2]. VSCode is a very popular text editor, with a clean and intuitive user interface.

A screenshot of a computer

Description automatically generated

Figure : Visual Studio Code

It includes a sidebar for file navigation, an integrated terminal and a customizable layout. This provides a powerful and modern experience for a developer. VSCode also includes a rich set of features, including a vast library of extensions, git integration, a debugger, Live Share, and support for a wide range of languages.

## Rich Text Editors

A screenshot of a computer

Description automatically generatedRich text editors, also known as word processors, are a different type of text editor. These editors allow you to edit text with a variety of formatting options beyond plain text. They are used for many purposes, including documentation, blogging, presentations and note taking.

Figure : TextEdit

One example of a rich text editor is TextEdit [3], which is a built-in rich text editor for macOS. It is a basic word processor, however, includes many features to help you format text. It also includes a plain text mode, which is suitable for coding and basic note taking. The interface is minimalistic and user-friendly, including a toolbar with formatting options.

Another word processor is Microsoft Word [4], which is one of the most popular and feature-rich word processors. It provides a comprehensive set of tools for creating, editing and formatting documents.

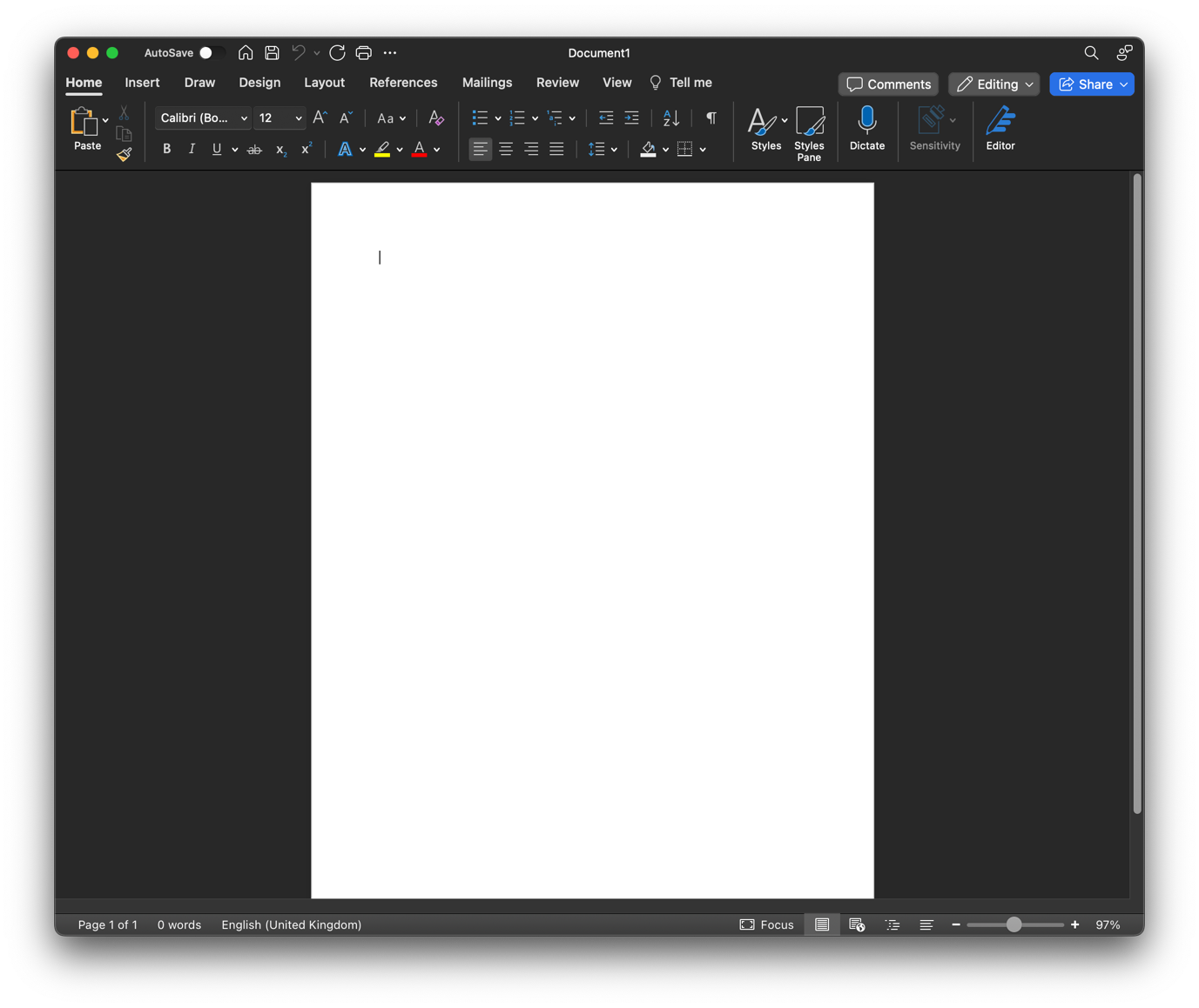
The user interface is user-friendly, and includes a toolbar at the top, providing a vast range of formatting and editing options.

Figure : Microsoft Word

## Objectives

The aim of this project is to create a plain text editor for developers to use. The ideal outcome of this project is to have a text editor which has a simple, user-friendly design, with as many features as possible.

## Language Considerations

The Electron [5] framework will be used to build this text editor, which will be used to create a desktop app through HTML, CSS and JavaScript. There are multiple reasons for using Electron to create a text editor. Firstly, Electron allows you to build applications that can run on different operating systems, including macOS, Windows and Linux. Secondly, it has a large ecosystem of libraries, plugins and tools available, which will help with including certain features in the text editor. As well as this, Visual Studio Code was built using Electron, which shows it can be used to create a feature-rich text editor with extensive features, and a highly customizable UI.

A potential downside to using Electron, is that the memory consumption can be high compared to native applications.

# Requirements Analysis

## User Interface

One of the first requirements for the text editor is the layout. The layout will consist of a ribbon bar at the top for tabs, a navigation panel on the left for file operations, and the text input area on the rest of the screen.

## Requirements

The intended users for this text editor are programmers who require a platform which will include many features and a clean and easy to use user interface.

### Mandatory

1. Text editor runs on Windows and MacOS
2. Editing capabilities:
   1. Inserting and deleting text
   2. Copying and pasting text
   3. Undo/redo functionality
3. File operations
   1. Creating a new file
   2. Saving a file
   3. Opening an existing file
4. Multiple tabs/windows
5. Syntax highlighting
6. Plugin support

### Desired

1. Search and replace
2. Keyboard shortcuts
3. Customisation, for example theme choices
4. Auto-save

# Project Plan

The plan is critical for a successful project. This section will outline the tasks required to complete the project and a timeline, providing a roadmap for the development of the project.

## Phases and Tasks

1. Proposal
2. Planning and designing.
   1. Designing the text editor’s architecture
   2. Plan implementation of key features
   3. Explore and research Electron capabilities
3. Coding
4. Interim Report
5. User testing
6. Refinement and optimization.
   1. Refine the user interface to improve user experience
   2. Optimize performance and address potential bottlenecks
   3. Expand functionality based on feedback
7. Final Report

Figure : Gantt Chart

Timetable to show when I will work on the project during the week:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Mon | Tue | Wed | Thu | Fri |
| 9:00 |  | Lecture |  |  |  |
| 10:00 |  | Lecture |  | Lecture |  |
| 11:00 | Lecture | Project |  |  |  |
| 12:00 | Lecture | Project | Project |  | Project |
| 13:00 | Project | Lecture | Project | Lab |  |
| 14:00 |  |  | Project | Lab |  |
| 15:00 | Seminar |  | Project |  |  |
| 16:00 |  |  |  |  |  |
| 17:00 |  | Lab |  |  |  |
| 18:00 |  |  |  |  |  |
| 19:00 |  |  |  |  |  |

This table shows the hours I have set aside to work specifically on the project. This will give me 8 hours per week.

# Log

**Term 1 Week 1**

Before starting this project, I already had experience with using HTML, CSS and JavaScript, however I hadn’t used them in Electron before. After choosing the project and deciding to use Electron to code it, I started researching to understand the framework and how I can use it.

After researching, I started writing my project proposal.

**Term 1 Week 2**

* Finished project proposal
* Created a plan for tasks, drawing out when to start certain tasks and which tasks need to be completed

**Term 1 Week 3**

* Started design sketches to understand what I wanted the text editor to look like.
* Set up file structure and GitHub [6] repository

**Term 1 Week 4**

* Began coding the basic structure of the text editor
* Continued to research Electron

**Term 1 Week 5**

* Created the layout of the text editor content
* Research on code editor libraries such as CodeMirror and Ace.

**Term 1 Week 5**

* Continued to add code to text editor
* Research into file management
* Started interim report

**Term 1 Week 6**

* Continued to add code to text editor
* Improved code

**Term 1 Week 7**

* Completed interim report

# Bibliography

|  |  |
| --- | --- |
| [1] | Vim, “Vim Documentation,” Vim, 2023. [Online]. Available: https://www.vim.org/docs.php. [Accessed 13 11 2023]. |
| [2] | Microsoft, “Visual Studio Code,” Microsoft, 2023. [Online]. Available: https://code.visualstudio.com/. [Accessed 13 11 2023]. |
| [3] | Apple, “TextEdit User Guide,” Apple, 2023. [Online]. Available: https://support.apple.com/en-gb/guide/textedit/welcome/mac#:~:text=With%20TextEdit%2C%20you%20can%20open,open%2C%20edit%20and%20convert%20documents. |
| [4] | Microsoft, “Microsoft Word,” Microsoft, 2023. [Online]. Available: https://www.microsoft.com/en-gb/microsoft-365/word. |
| [5] | OpenJS, “Electron Docs,” Electron, 2023. [Online]. Available: https://www.electronjs.org/docs/latest. |
| [6] | B. Pickle, “Text Editor,” TechTerms.com, 10 October 2022. [Online]. Available: https://techterms.com/definition/texteditor. [Accessed 13 11 2023]. |