

**School Of Computer Science and Engineering**

**VIT-AP UNIVERSITY, AMARAVATI**

SWE3002: Software Project Management

Review-2

**Ember Text Editor**

**Submitted To:**

Dr. Rohit Kumar Das

**Submitted By:**

Tathagata Guha Ray (21BCE7778)

Software Requirement Specification (SRS)

For Ember - Text Editor Application

1. **Introduction:**

**1.1 Purpose:**

This document is intended to outline the characteristics of Ember Application so that it can be used as a reference for developers and as a software validation document for potential clients.

This software concept is valuable in certain situations that will help provide a range of services to people that require it.

* 1. **Intended Audience:**
* **Stakeholders:** Clients, Developers, Project Designers
* **Users:** Developers, Students, Casual Users

**1.3 Intended Use:**

The software requirements specification serves as the foundation for this whole project. It also establishes the template for all development teams to follow. It’s utilized to deliver essential information to a variety of teams including development, quality assurance, operations and maintenance.

**1.4 Scope:**

* The aim for this software is to provide a lightweight and feature-full text editor application.
* The application has a modern interface.
* It must have very high performance, so that it runs with minimal overhead on any hardware specification.
* App & Editor preferences must be customizable by the user.

**1.5 Definitions & Acronyms:**

* **SRS :** Software Requirements Specifications
* **GUI :** Graphical User Interface
* **Stakeholders :** A person, group or organization who participates in the project and has direct influence over the outcome(s) of that project.

**2. Overall Description:**

Ember – Text Editor is an application developed to be lightweight and quick. It is aimed at individual users and developers who want to view or edit a single document as quickly as possible, while also providing a host of features that improve the productivity of the user.

Ember aims to save time and resources of the host system and of the user.

**2.1 Product Perspective:**

Ember – Text Editor is designed as a standalone application that focuses on providing a streamlined and efficient text editing experience. Unlike comprehensive text editors with extensive plugin ecosystems, Ember emphasizes speed and simplicity, catering to users who need to quickly open, edit, and save documents without unnecessary distractions.

**2.2 Product Functions:**

* Quick document loading and editing to enhance user productivity.
* Lightweight application design to minimize system resource usage.
* Basic syntax highlighting for popular programming languages to aid developers.
* Support for multiple file formats to accommodate diverse editing needs.
* Auto-save feature to prevent data loss during editing sessions.
* Customizable user interface to adapt to user preferences.
* Undo and redo actions to streamline the editing process.

**2.3 User Needs:**

* Users need a fast and responsive text editor that allows for quick loading and editing of documents without unnecessary delays.
* The application should be resource-efficient, ensuring it can run smoothly on a wide range of hardware without compromising system performance.
* Developers require syntax highlighting for better readability and easier coding in popular programming languages.
* Users need the ability to customize the user interface and editor preferences to tailor the application to their specific workflow and aesthetic preferences.
* An auto-save feature is essential to prevent data loss during unexpected disruptions, such as power outages or accidental closures.
* Support for multiple file formats is necessary to handle diverse document types that users may encounter.
* Ease of use with intuitive features that cater to both casual users and more technical users like developers is important for broad accessibility.

**2.4 Principal Actors:**

Principal Actors in this software are the Developers, Students and casual users.

**2.5 General Constraints:**

The application is currently supported on only Windows Operating System, 10 or higher.

**3. System Features and Requirements:**

**3.1 Functional Requirements:**

The functional requirements define the core operations and features the Ember text editor will provide to its users. These include:

* **Document Handling**: Ember must allow users to open, view, edit, and save various file types, such as .txt and programming files like .cpp, .py, .html, etc.
* **Syntax Highlighting**: The editor should automatically highlight syntax for popular programming languages like Python, C++, JavaScript, and more.
* **Undo/Redo Functionality**: Users must be able to undo and redo changes within the document, albeit with some limited depth, for efficient editing.
* **Auto-Save**: The application will automatically save documents at regular intervals to avoid data loss.
* **Customizable UI**: Users will have the ability to change the editor's theme, font, and layout settings to suit their preferences.

**3.2 Non-Functional Requirements:**

Non-functional requirements define the quality attributes and constraints that Ember must adhere to. These include:

* **Usability**: Ember must have an intuitive, clean interface that allows for quick navigation of functions without requiring users to follow complex procedures.
* **Portability**: The text editor should be lightweight and retain the same performance characteristics across multiple system specifications, maintaining the same user experience.
* **Scalability**: Ember should efficiently handle both small text files and larger codebases, providing a seamless experience regardless of file size.
* **Reliability**: The application must ensure data integrity by minimizing crashes and ensuring robust auto-save functionality.
* **Maintainability**: Ember should have a modular design, allowing easy updates, bug fixes, and feature additions without affecting existing functionality.

**3.3 Performance Requirements:**

Performance requirements focus on ensuring optimal performance in various scenarios. For Ember:

* **Low Memory Usage**: The application should consume minimal system resources, ensuring it remains responsive and fast even on machines with limited hardware capabilities.
* **Fast Startup Time**: Ember must launch in under two seconds, enabling users to start editing quickly without delays.
* **Efficient File Handling**: Files, regardless of their size, should load within a few milliseconds, ensuring that users can view and edit documents without any lag.
* **Smooth UI Interaction**: Actions such as scrolling, typing, and menu navigation should happen instantaneously, providing a responsive user experience.