Team H

LLM Editor
Software Requirements Specification
For <Subsystem or Feature>

Version <1.0>

Revision History

Date	Version	Description	Author
<18/03/25>	<1.0>	First Phase Report	Khandaker, Fardin Ismam Saraf, Fardin Ferdous Uddin, Raida Z Yeung, Tak Kit

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the functional and non-functional requirements for the LLM-Based Text Editor System. This system integrates a Large Language Model (LLM) to assist users in editing and improving text, offering both manual and AI powered corrections. It supports different user roles, including free users, paid users and super users, each with specific privileges. The document serves as a guide for developers, testers and stakeholders to ensure the successful implementation of the system.

1.2 Scope

The LLM-Based Text Editor is a local GUI-based application designed for text editing with AI assistance. It enables users to submit, correct, and refine their text using an interactive, token-based model. Key features include:

- Text editing and correction: users can edit text manually or request LLM-Based corrections
- Token system: paid users purchase tokens to submit and correct text
- Blacklist management: a word filtering system allows flagged words to be censored
- Collaboration and sharing: paid users can share text files and work together
- Super user control: handles user complaints, approvals, and system moderation

This system ensures efficient AI powered text correction while maintaining user control, fairness and monetization.

1.3 Definitions, Acronyms, and Abbreviations

- LLM- Large Language Model, an AI model used for text analysis and correction
- GUI- Graphical User Interface, the visual system for user interaction
- Token- A virtual currency used within the system for user interaction
- Blacklist- A list of words that are censored

1.4 References

1.5 Overview

This document outlines the system’s functionalities, use cases and constraints. The document provides details of the user interactions, token-based interactions and AI integration.

2. Overall Description

2.1 Use-Case Model Survey

The use case diagram shown in Figure 1 illustrates how the users interact with the system. The functional requirements of the system defined by the use cases are covered in Section 3.1.

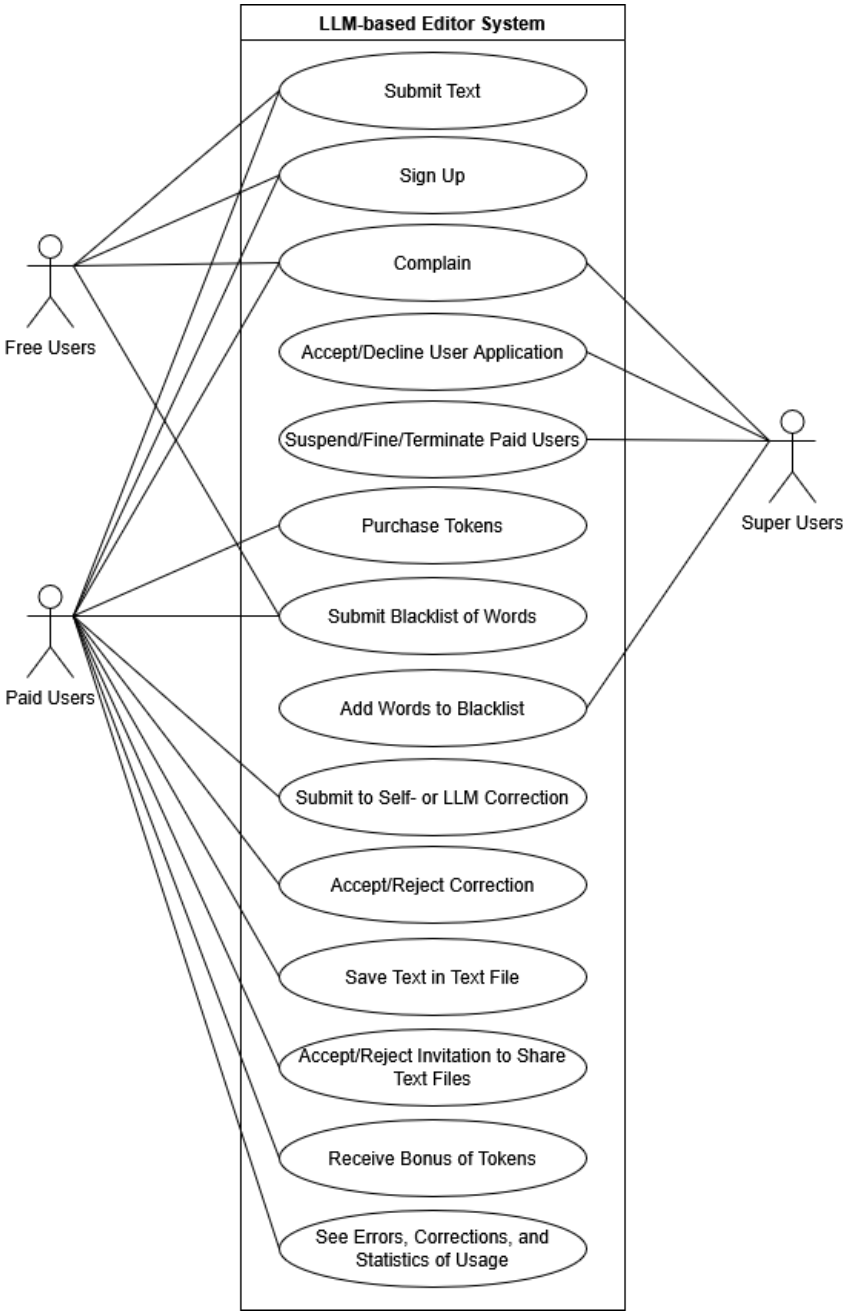

Figure 1. Use-Case Diagram

The system consists of 3 types of users:

Free users can freely submit texts with up to 20 words, but they will be thrown out of the system if they tried to submit more than 20 words, and they cannot log in as a free user within 3 minutes. Free users can sign up as paid users with a password.

Paid users can submit texts with more than 20 words, if their accounts have enough amount of money.

Super users can handle user sign-up and complaints, accept or deny user applications, and suspend, fine or terminate paid users.

2.2 Assumptions and Dependencies

The development of the LLM based text editor relies on assumptions and dependencies that impact the functionality of the system.

- Token-based system- the software depends on a functional token-based system for managing paid user access and transactions
- User management subsystem- the platform utilizes a user authentication and management system for handling free, paid and super users
- Blacklist enforcement- the application presumes an effective mechanism for maintaining and enforcing blacklisted words determined by the super users

3. Specific Requirements

3.1 Use-Case Reports

The use cases which define the functional requirements of the system are listed below:

Use case: Submit text

Description: Both free and paid users can freely submit texts with up to 20 words. If free users tried to submit more than 20 words, they will be thrown out of the system by super users. Paid users can submit texts with more than 20 words by typing in a text box or selecting a text file, and the system will charge the number of tokens based on the number of words for the submission.

Use case: Sign up

Description: Free users can sign up as paid users with a password, and super users handle user sign-up.

Use case: Complain

Description: Both free and users can complain to super users who handle complaints. Particularly, paid users can complain collaborators to super users, super users will notify the complained user, in which complained user must respond to when logging in, after seeing the disputes by the user being complained about.

Use case: Accept or decline user application

Description: Super users decide whether to accept or decline free users' applications to sign up as paid users.

Use case: Fine, suspend or terminate paid users

Description: If paid users submit text when the available tokens are less than the words, the submission will fail, and super users will penalize the paid users with half of the available tokens deducted. Super user can decide to punish the complainer or the complained by a certain number of tokens.

Use case: Purchase tokens

Description: Paid users can purchase a certain number of tokens.

Use case: Submit blacklist of words

Description: Both free and paid users can submit a blacklist of words that the users cannot use. If the users submit texts which contain any words in the blacklist, they will be replaced by the corresponding number of '*', and for paid users, the number of tokens is charged by the length of the blacklisted words.

Use case: Add words to blacklist

Description: Super users determine if the blacklist of words submitted by the users should be added to the blacklist of the system.

Use case: Submit to self-correction or LLM correction

Description: Paid users can choose to submit to self-correction or LLM correction. If the users choose self-correction, the system will charge the number of tokens equal to half of the words self-corrected.

Use case: Accept or reject correction

Description: If paid users submit to LLM correction, the LLM will return the corrected highlighted texts. The system will ask the users to accept the correction or not. If accepted, one token will be deducted. If not accepted, users can save the wrong word labeled by LLM as a correct, so that the word will not be highlighted anymore. The users can also reject the LLM's correction entirely by stating the reason. If super users decide the reason is acceptable, 1 token will be deducted from the paid users' account. Otherwise, 5 tokens will be deducted.

Use case: Save text in text file

Description: After the interactive correction process, the paid users can choose to save the text in a text file with a charge of 5 tokens.

Use case: Accept or reject invitation to share text files

Description: Paid users can share their text files by inviting other paid users, where the invited paid users can accept the invitation or reject it. If accepted, the invited users become collaborators on the text file, and they can choose self or LLM correction on the text file. If rejected, the inviter is charged 3 tokens for reckless inviting.

Use case: Receive bonus of tokens

Description: If paid users submit a text with more than 10 words to LLM with no error, the user will receive a bonus of 3 tokens deposited to the account.

Use case: See errors, corrections, and statistics of usage

Description: Free users and paid users should have different graphics user interface, the statistics of the paid users are shown in the top panel with edited texts, the number of used and available tokens, and the number of corrections being made. Paid users should be able to see all the errors and their corrections by the LLM or self-correction, and statistics of the usage of the system.

3.2 Supplementary Requirements

- User interface- The GUI displays correction history and blacklist features
- Security- Paid users must log in with a password
- Performance- Users can freely submit texts whether its paid or free users
- Usability- The user interface must be easy to navigate and quick text corrections
- Error handling- Users are notified of submission failures and penalties

4. Supporting Information