Software Requirements Specification

Echo

Version 1.0

Prepared by

Ashwin Binu Abraham
Deepak P
Salihu Ahamed
Fathima Nooha Kottangodan

TKM College of Engineering

5th March 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Salihu Ahamed	25/02/2023	Initial draft prepared	1.0
Ashwin Binu	26/02/2023	Added additional functional requirements	1.0
Fathima Nooha	28/02/2023	Added overall description section	1.0
Deepak P	28/02/2023	Edited external interface requirements	1.0
Fathima Nooha	02/03/2023	Updated performance requirements	1.0
Ashwin Binu	03/03/2023	Edited System features part	1.0
Deepak P	04/03/2023	Added security requirements	1.0
Salihu Ahamed	05/03/2023	Finalized document for submission	1.0

1. Introduction

1.1 Purpose

The purpose of this project is to develop a mobile application that enables visually impaired users to easily read and send emails using voice commands. The application aims to provide visually impaired users with a solution that is tailored to their unique needs and challenges, allowing them to use email in a more intuitive and efficient way. The project also aims to improve the quality of life for visually impaired users by providing them with a tool that increases their independence and participation in society.

1.2 Document Conventions

This document is based upon:

- IEEE 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
- IEEE 830-1984 - IEEE Guide for Software Requirements Specifications

1.3 Intended Audience and Reading Suggestions

Intended audience includes college professors, developers, industry experts, and other stakeholders who may be involved in the development or evaluation of the system

Review the overall design and system description, then focus on specific requirements and any diagrams or visual aids that can help clarify system behavior. Note any areas of interest or concern that could impact involvement or use of the system.

1.4 Product Scope

The mobile application aims to provide an accessible and convenient email client for visually impaired individuals by allowing them to manage their Gmail and Outlook accounts through voice commands. The purpose of the application is to provide visually impaired individuals with a user-friendly and efficient means of managing their email, which is essential in today's digital world. The key objectives of the application include:

- To provide a hands-free and intuitive email management experience through voice commands.
- To enable speech-to-text and text-to-speech functionality for ease of use.
- To offer compatibility with assistive technology, making it fully accessible for visually impaired individuals

To ensure the success and sustainability of the application, the following business strategies will be implemented:

- Partnership with assistive technology companies: Collaboration with assistive technology companies can enable bundled product and service offerings, expanding the app's reach to the target market.
- Targeted marketing: Marketing the application through channels that cater to the visually impaired community, such as organizations and support groups, will help reach the intended audience more effectively.

1.5 References

- Gmail API: https://developers.google.com/gmail/api/guides
- Outlook API: https://learn.microsoft.com/en-us/outlook/rest/
- Flutter Text To Speech Plugin: https://pub.dev/packages/flutter_tts
- Flutter Speech To Text Plugin: https://pub.dev/packages/speech_to_text
- Wake Word Detection Plugin: https://pub.dev/packages/porcupine_flutter/versions/2.0.1
- UI/UX References: https://dribbble.com/search/voice-email-app
- A survey of blind users on the usability of email applications by https://link.springer.com/article/10.1007/s10209-012-0285-9
- Voice Based Mail System for Visually Impaired: https://www.ijert.org/voice-based-mail-system-for-visually-impaired
- GeekForGeeks: https://www.geeksforgeeks.org/how-to-write-a-good-srs-for-your-project/
- Perforce blog by Gerhard Krüger and Charles Lane: https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document

2. Overall Description

2.1 Product Perspective

The voice-based email application is designed to provide an alternative means of email communication for individuals with visual impairments. The application will interact with various email providers through APIs to ensure compatibility and access to existing email accounts. The application will also integrate natural language processing (NLP) and speech-to-text (STT) technology to interpret user voice commands and generate text-based emails.

- **System Interfaces**

The voice-based email application will interface with the iOS and Android operating systems and the associated software and hardware. This includes access to system libraries and resources, such as audio and speech recognition libraries.

- **User Interfaces**

The user interface of the application will be designed to be simple and intuitive, with large buttons and clear audio feedback. The interface will be accessible for visually impaired users, allowing them to navigate the app easily and use voice commands to compose and read emails.

- **Hardware Interfaces**

The application will interface with the microphone and speaker of the user's device to receive voice commands and provide audio feedback. The user will speak commands into the microphone, which will be interpreted by the app's NLP and STT technology. The app will then provide audio feedback via the speaker, reading incoming emails aloud to the user.

- **Software Interfaces**

The voice-based email application will integrate with various email providers through APIs to ensure compatibility and access to existing email accounts. The application will be designed to integrate seamlessly with popular email providers, such as Gmail and Outlook, to ensure that users can access their email accounts without having to switch to a different email client.

- **Communications Interfaces**

The voice-based email application will use Wi-Fi or cellular data to communicate with email servers and API endpoints. The app will use standard email protocols such as SMTP, POP3, and IMAP to send and receive emails. The app will also use APIs to connect to email providers and to integrate with other services such as speech-to-text and text-to-speech conversion services.

2.2 Product Functions

- Login and authentication
- Voice-based email composition
- Voice-based email reading
- Integration with popular email providers
- User interface
- Compatibility with iOS and Android devices
- Settings and customization
- Error handling and notification

2.3 User Classes and Characteristics

Different users are identified on the basis of the review process of the document. It is listed as follows:

1. Customers

a. Educational institutions: The solution can be used in educational institutions to facilitate communication between students and teachers with visual or hearing impairments.

b. Non-profit organizations: Non-profit organizations that serve the disabled community can use the solution to enhance accessibility for their clients.

2. End Users

a. Visually Impaired Users: These are the primary users of the application who have visual impairments and face challenges in accessing and using email. They may have varying degrees of visual impairments and may require different levels of assistance from the application.

b. Regular Users: These are users who do not have visual impairments but may still benefit from the application's voice-based email composition and reading functions. They may also use the application to send emails hands-free while driving or doing other tasks.

c. Technical Users: These users may have technical expertise and require advanced features of the application, such as integration with custom email providers or customization of voice commands.

2.4 Operating Environment

The voice-based email application will be designed to operate on the following environments:

a. Mobile Devices: The application will operate on mobile devices running iOS 13 or later and Android 10 or later.
b. Internet Connectivity: The application will require an active internet connection to interact with the email server and API endpoints.
c. Hardware Requirements: The application will require a microphone and speaker to enable voice-based commands and audio feedback.
d. Software Requirements: The application will require access to speech-to-text and text-to-speech engines for voice interpretation and audio feedback, respectively.
e. Power Requirements: The application will require sufficient battery power to ensure uninterrupted operation.

2.5 Design and Implementation Constraints

- Hardware Constraints
    - The application requires the microphone and speaker of the user's device to be in good working condition.
    - The application is designed to work on both iOS and Android devices.
    - The Android API level should be above 21.
    - The device must have a good camera, speaker, and microphone.
    - The application works only on Wi-Fi or mobile data.

- Regulatory Constraints
    - The application must comply with the General Data Protection Regulation (GDPR).

- Functional Constraints
    - The application must use Flutter plug-ins and the Flask framework for front-end and back-end development respectively.
    - The application must be developed using Android Studio or VS Code IDE.
    - The user interface should be simple and easy to understand, with readable text and smooth transitions.
    - The application should be space-efficient and use a readable and maintainable code.
    - The variables should be named using camel case, and the class names should start with a capital letter.
    - The application should follow proper exception handling mechanisms.

- Language Requirements
    - The front-end should be developed using Dart programming language.
    - The back-end should be developed using the Python programming language.

2.6 User Documentation

- A user guide will be distributed to customers and developers as soft-copy in pdf format.
- An online forum will be created to allow users to connect with each other and ask questions, share tips, and provide feedback on the application.
- The end users (target) would be provided a tutorial video available in the Google Play Store while installing the app.

2.7 Assumptions and Dependencies

Assumptions:

- Users have access to a compatible device with the necessary hardware (microphone, speaker, camera) and software (Android or iOS operating system, internet connectivity).
- Users have basic knowledge of using mobile devices and accessing email.
- The email service providers' APIs remain stable and compatible with the application.

Dependencies:

- Flutter framework for cross-platform mobile application development.
- Speech-to-text plugin for converting spoken words to text.
- Text-to-speech plugin for converting text to spoken words.
- tflite_flutter and tflite_flutter_helper plugins for integrating machine learning models into the application.
- Email provider APIs for accessing and managing email accounts.

3. External Interface Requirements

3.1 User Interfaces

At initial startup the homepage will contain the login to the email of users.Then on homepage will show buttons for:

Inbox
Compose
Starred
Settings

- A button to read out the text on the screen(Text to speech) and provide audio assistance will be placed on the bottom of the screen.
- A button for speech to text option will be available in the compose email section.
- On every new page open an audio will be spoken to detail the current state of the page and to ask the instruction from the user.

3.2 Hardware Interfaces

The application is user-friendly.Buttons can be used for mode of transactions.The application mainly deals with audio data inputs.To record the user's voice instructions and send them to the app for processing, microphone is used.A speaker or headphones is used to receive auditory feedback. To connect with external hardware devices like Bluetooth headsets or hearing aids, the app needs Bluetooth connectivity.

3.3 Software Interfaces

TTS flutter plugin is used for text-to-speech features.Speech_to_text plugin is used for speech-to-text conversion.

4. System Features

4.1 Integration with Email Providers

4.1.1 Description and Priority

The mobile application is designed to be compatible with two popular email providers, Gmail and Outlook. This means that users with Gmail or Outlook email accounts will be able to use the app to access and manage their email

The compatibility with email providers feature is of high priority as it is a fundamental functionality of the application. Without this feature, the application would not be able to serve its purpose of providing a user-friendly email client for visually impaired individuals.

4.1.2 Stimulus/Response Sequences

1. User launches the application and selects the option to add an email account.
2. Users are prompted to enter their email address and password through voice commands.
3. System verifies the entered credentials and fetches the user's email account information.
4. User selects the email provider from the list of supported providers.
5. System prompts the user to grant permission for the application to access their email account.
6. User grants permission through voice commands.
7. System completes the integration process and displays the user's inbox on the app's home screen.

For first-time users, the following steps are added:

1. Download and install the mobile application from the app store.
2. Launch the application and grant necessary permissions, such as microphone and accessibility access.
3. Upon opening the application, users will be prompted to enter their email credentials for Gmail or Outlook.
4. The application will verify the credentials and connect to the email account.
5. Users will be able to start managing their emails through voice commands immediately.

4.1.3 Functional Requirements

REQ-1: The app should be compatible with Gmail and Outlook email providers.
REQ-2: The app shall provide users with the option to set up their email accounts using voice commands.
REQ-3: The app shall authenticate user credentials for the email provider through the user's voice command.

4.2 Voice-based email composition

4.2.1 Description and Priority

This feature allows users to compose emails using voice commands, making it more accessible and convenient for individuals with visual impairments or those who prefer a hands-free approach to email composition.

The priority of the voice-based email composition feature is high, as it is a key functionality of the application that enables visually impaired users to compose and send emails hands-free. It is a primary reason why users would choose to use this application over other email clients.

4.2.2 Stimulus/Response Sequences

1. User says "Compose email".

2. System prompts the user to dictate the recipient's email address.
3. User dictates the recipient's email address.
4. System confirms the recipient's email address and prompts the user to dictate the subject of the email.
5. User dictates the subject of the email.
6. System confirms the subject of the email and prompts the user to dictate the body of the email.
7. User dictates the body of the email.
8. System confirms the body of the email and prompts the user to confirm whether they want to send the email.
9. User confirms they want to send the email.
10. System sends the email to the recipient's email address and confirms that the email has been sent.

4.2.3 Functional Requirements

REQ-1: The application should have a voice recognition system capable of transcribing spoken text accurately
REQ-2: The application should be able to format the transcribed text into a readable email format
REQ-3: The application should allow users to edit the transcribed text if necessary
REQ-4: The application should provide users with a confirmation option before sending the email

4.3 Text-to-speech for reading emails

4.3.1 Description and Priority

The text-to-speech feature for reading incoming emails is a high-priority requirement of the system, as it is essential for visually impaired or blind users who rely on auditory feedback to access their emails.

This feature enables the system to convert incoming email text content into audio format, which can then be read aloud to the user by the system's built-in text-to-speech engine. The user can also control the speed and volume of the speech output, as well as pause, rewind, or skip the audio playback as needed.

4.3.2 Stimulus/Response Sequences

The stimulus for this feature is an incoming email that has been received by the user's email account. When a new email arrives, the system should automatically trigger the text-to-speech feature to read the email's contents aloud to the user.

The response sequence for this feature involves several steps:

1. The system identifies the new email in the user's inbox and extracts the email's text content.
2. The system sends the email's text content to the text-to-speech engine for conversion into audio format.

3. The system plays the audio output through the user's device speakers or headphones.
4. The user can control the audio playback and adjust the speech speed and volume as needed.

4.3.3 Functional Requirements

REQ-1: The system must be able to detect new incoming emails in the user's email account and extract the text content of the email.
REQ-2: The system must have a built-in text-to-speech engine capable of converting the email's text content into an audio format.
REQ-3: The user must be able to control the speed and volume of the speech output through the system's user interface.
REQ-4: The user must be able to pause, rewind, or skip the audio playback as needed through the system's user interface.
REQ-5: The system must provide a clear and natural-sounding audio output that is easily understandable by the user.

4.4 Ability to edit and delete emails

4.4.1 Description and Priority

The ability to edit and send delete is a crucial feature of any email application. This feature allows users to manage their email communications effectively by composing new emails, modifying the content of existing emails, and deleting unwanted emails. The feature provides a user-friendly interface to perform these actions with ease and efficiency.

This feature has high priority as it is essential for the core functionality of the email application. Without this feature, the application would not be useful for managing email communications.

4.4.2 Stimulus/Response Sequences

1. When the user says "edit email," the system prompts the user to select an email to edit.
2. The user selects the email and the system opens the email for editing.
3. The user can then edit the email by using voice commands such as "insert text," "delete text," or "change text."
4. When the user has finished editing, they can say "send email" to send the updated email.

4.4.3 Functional Requirements

REQ-1: The system should have the ability to recognize voice commands for editing emails.
REQ-1: The system should provide audio feedback to confirm user actions, such as deleting or changing text.
REQ-1: The system should allow the user to undo and redo changes made to the email.
REQ-1: The system should have the ability to save drafts of emails in progress.

REQ-1: The system should provide error messages if the user tries to send an email without a recipient or with an invalid email address.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system should respond to user input within 2 seconds.
- The system should load images and data within 5 seconds.
- The system should be able to handle 100 simultaneous users without performance degradation.

5.2 Safety Requirements

- The system should not allow users to upload or access malicious files.
- The system should encrypt all user data in transit and at rest.
- The system should have a failsafe mechanism to prevent catastrophic failures.

5.3 Security Requirements

- The system should have multi-factor authentication for user accounts.
- The system should have role-based access control to restrict user access to certain features.
- The system should be compliant with GDPR and other privacy regulations.

5.4 Software Quality Attributes

- Capacity: The storage requirements of the app can be minimal since no information about the user is stored.
- Usability: The product is very simple and easy to use as it has an AI assistant to assist the disabled people. It will be required for better communication.
- Compatibility: Compatible with any version of the android which supports voice assistants.
- Response Time: Each page loads within 10-15 seconds and the output is expected to have the same response time.
- Reliability: Chances of failure or system crash is low compared to other heavy applications that require huge storage. However, the application is difficult to work without Wi-Fi or mobile data.
- Availability: The app will be available to the user at all times except times of maintenance. It will be made available in the Play Store for installation.
- Efficiency: The app is very efficient as the user can reach their goal within minimal time.

5.5 Business Rules

- Only registered users should be able to access certain features.
- Users should not be able to upload copyrighted or illegal material.
- Users should be able to report any inappropriate content or behavior.

6. Other Requirements

- Legal Requirements

The system shall comply with all applicable laws and regulations regarding user privacy and data protection, including the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA).

- Reuse Objectives

The system shall be designed with a modular architecture to facilitate code reuse and extensibility. The system shall be designed using standard software engineering principles to maximize reuse of existing software libraries and frameworks.

- Accessibility Requirements

The system shall comply with accessibility standards and guidelines, including the Web Content Accessibility Guidelines (WCAG) 2.1. The system shall be designed to be accessible to users with disabilities, including visual, auditory, and motor impairments.

- Performance Requirements

The system shall be designed to handle a high volume of concurrent users, with a target response time of 2 seconds or less for all user interactions. The system shall be scalable to support future growth and increasing user loads.

- Usability Requirements

The system shall be designed to be intuitive and easy to use, with a modern and responsive user interface. The system shall be designed to minimize user errors and to provide clear and concise error messages when errors occur.

- Maintenance Requirements

The system shall be designed to be maintainable and easy to update, with clear separation of concerns and modular design. The system shall be designed to minimize downtime during maintenance and to provide easy rollback in case of errors.

- Testing Requirements

The system shall be designed with testing in mind, with a comprehensive suite of automated tests to ensure functionality, performance, and security. The system shall be designed to be testable, with clear separation of concerns and well-defined interfaces.

Appendix A: Glossary

TTS: Text-to-Speech
STT: Speech-to-Text
API: Application Programming Interface
UI/UX: User interface and user experience