Voice-based Email System

Software Requirement Specification

Version 1.0

Prepared By
Megha K C
Priyanga P Kini
Srividya Krishnakumar
Karthik Suresh

Document Control Data Sheet

Project Code	VBES
Project Name	Voice-based Email System
Document Name	Software Requirement Specification
Version	1.0
File Name	VoiceBasedEmailSystem_SRS.doc
Classification	Confidential
Client	

	Name	**Signature & Date**
Prepared By		
Reviewed By		
Approved By		
Distribution List		

Revision History

Version	Date	Author	Reviewer	Change Description

Table of Contents

1. Introduction 5

1.1 Purpose 5
1.2 Intended Audience 5
1.3 Product Scope 5
1.4 Definitions, acronyms and abbreviations 5
1.5 References 5

2. Overall Description 6

2.1 Product Perspective 6
2.2 Product Functions 6
2.3 User Classes 6
2.4 Development Environment 7
2.5 Constraints 7
2.6 Assumptions and Dependencies 7

3. External Interface Requirements 9

3.1 User Interfaces 9
3.2 Hardware Interfaces 9
3.3 Software Interfaces 9
3.4 Communications Interfaces 9

4. System Features 9

4.1. Use cases 10
4.1.1 Use Case – 1: Login 11
4.1.2 Use Case – 2: Logout 12
4.1.3 Use Case – 3: View notifications for new emails 13
4.1.4 Use Case – 4: List Inbox mails 15
4.1.5 Use Case – 5: Listen to Individual Mail 18
4.1.6 Use Case – 6: Compose Mail 19
4.1.7 Use Case – 7: Organize Mails 22
4.1.8 Use Case – 8: View Trash 23
4.1.9 Use Case – 9: View Sent Mails 26

5. Other Nonfunctional Requirements 28

5.1 Performance Requirements 28
5.3 Security Requirements 28

1. Introduction

1.1 Purpose

The purpose of the Software Requirements Specification document is to maintain all the functions and the specifications of ‘Voice-based Email System’. Besides it contains detailed descriptions of all the requirements specified.

1.2 Intended Audience

The intended audience of VBES project include visually challenged people. Moreover the system is also open to the common people.

1.3 Product Scope

Though email is a viable form of formal communication, it is not accessible by the visually impaired. The project named ‘Voice-based Email System’ aims at implementing a natural language interface to the existing email system. The system will allow users to receive voice notifications for new emails, to listen to the content of their emails, organize their inbox, search, compose, reply or forward emails. The ultimate aim is to ensure the smooth and efficient functioning of the email system so that the visually challenged people can access their email like any other common people.

1.4 Definitions, acronyms and abbreviations

1.4.1 Abbreviations

VBES	-	Voice-based Email System
SRS	-	Software Requirement Specification

1.4.2 Definitions

Organize mails: Categorize mails as spam or create label.

1.5 References

- IEEE Recommended Practice for Software Requirements Specification-IEEE Std 830-1998.

2. Overall Description

2.1 Product Perspective

Voice-based Email System aims at creating an interface over the existing Gmail. The VBES application will ensure that visually challenged people will attain confidentiality over their received and sent mails just like any other common man.

2.2 Product Functions

The project aims at developing an interactive software interface over the existing Gmail system. The various functionalities to be dealt by the system are classified into different modules.

The proposed software shall have the following modules or functions:

- Authentication Module
- Login Module
- Forgot Password Module
- Home Page Module
- Compose mail Module
- Inbox Module
- Sent mail Module
- Organize Module
- Notification Module

2.3 User Classes

The user classes for the system being developed have been broadly classified as follows. The actors of the 8 use cases mentioned in section 4 can be classified under the 2 user classes listed below.

- Authenticated user
- Unauthenticated user

User	**Characteristics**
Authenticated user	• Login • Logout • Compose mail • View sent mails • Organize mails

	● Receive notifications ● List inbox mails ● Listen to individual mail ● List Trash
Unauthenticated User	● Register

2.4 Development Environment

Development environment is as follows. Final decision on the development environment shall be taken during the design phase.

- Gmail add-ons
- Android Studio

2.5 Constraints

a. **Regulatory Policies**: NA
b. **Hardware Limitations**: The user must have access to a fingerprint scanner.
c. **Interfaces to other application**: VBES project intends to build an interface over the existing email.
d. **Parallel operations**: NA
e. **Audit Functions**: NA
f. **Control Functions**: NA
g. **Safety and Security Considerations**: The password and a valid username are the security issues. The backup process at the server side shall satisfy data protection.
h. **Reliability Requirements**: Total number of bugs in the system shall not exceed 1% of the total line number of code, except connection reliability, which is out of range.
i. **Criticality of the Application**: NA

2.6 Assumptions and Dependencies

The following assumptions are made with regard to this project:

- The user must have access to a fingerprint scanner either in his phone or laptop etc.
- It is assumed that the duration of the project is about 5 months, this is only a rough estimate.

- All the hardware and software requirements of the team to carry out the development activities are not finalised.

3. External Interface Requirements

3.1 User Interfaces

VBES is a web based email system for visually impaired people. Users interact with the system using user friendly natural language interfaces and Graphical User Interfaces (GUI). The formats of various GUIs like screens, web pages and reports of the system shall be furnished in the design document.

3.2 Hardware Interfaces

No hardware interface is required for the system.

3.3 Software Interfaces

Voice-based Email System should integrates Gmail API to send and receive mails.

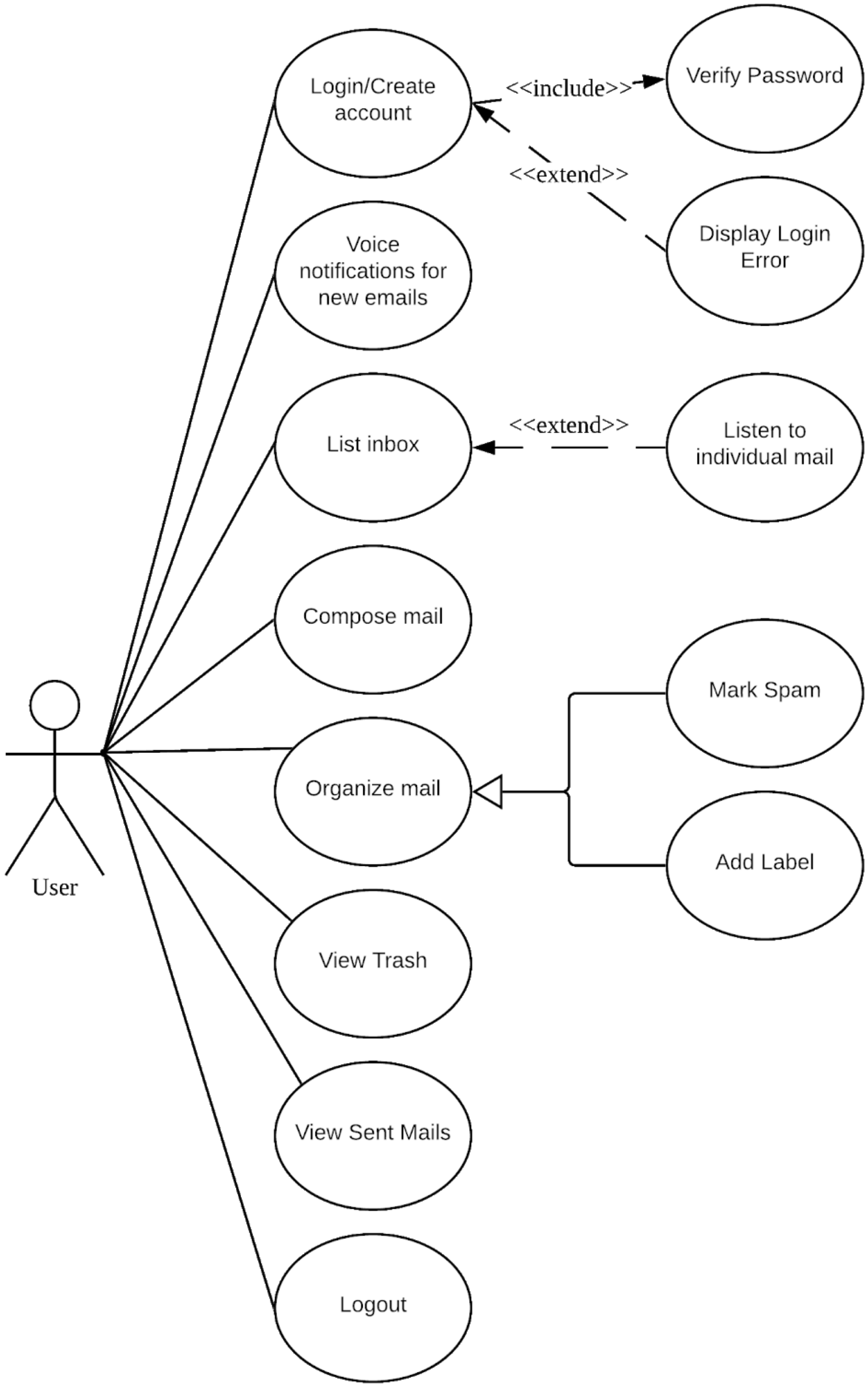
3.4 Communications Interfaces

Users will access the application using web browsers like Internet Explorer 7.0. The communication will be through standard HTTP protocol and SMTP protocol.

4. System Features

This section gives the details of system features and functions identified as different use cases relevant for various users (or actors) of the system. The following sections group and specify the use cases according to the 5 user classes identified and listed under section 2.3.

4.1 Use cases


Login/Create account
<<include>>
Verify Password
<<extend>>
Display Login Error
Voice notifications for new emails
List inbox
<<extend>>
Listen to individual mail
Compose mail
Mark Spam
Organize mail
Add Label
User
View Trash
View Sent Mails
Logout

4.1.1 Use Case – 1: Login

Description:
The user can login to the system by scanning his/her fingerprint.

Actors:
The actors include registered users.

Precondition:
The User should be registered to the system.

Main Flow of Events:

1. The user scans his/her fingerprint.
2. System validates username and fingerprint.
3. If the details are valid, user logs in to the system.

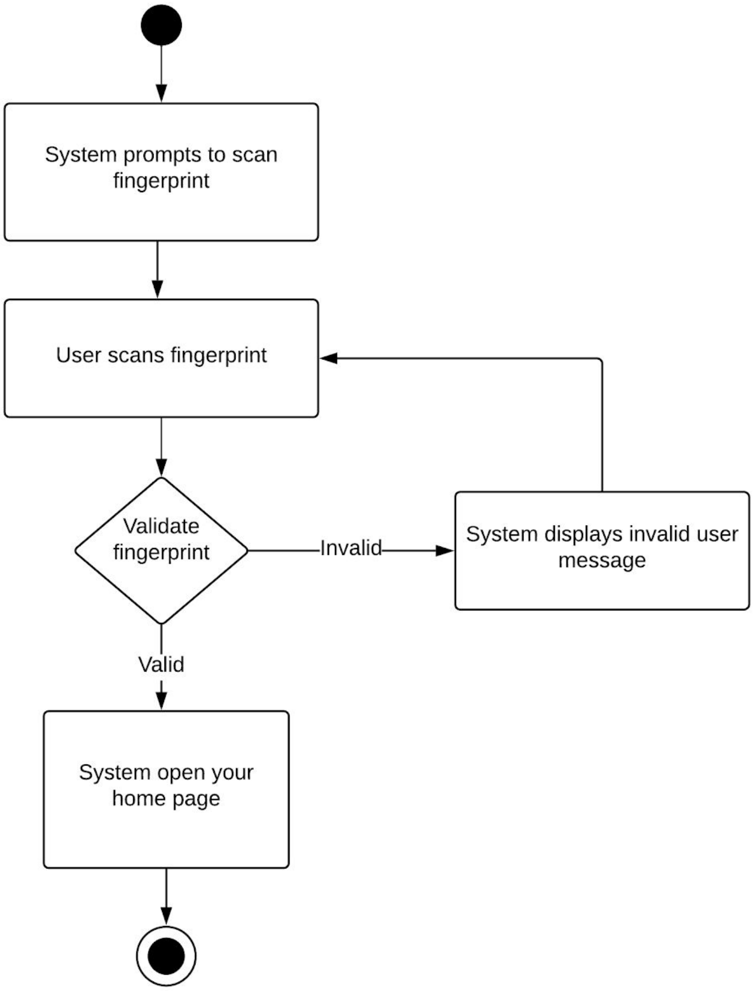
Alternate Flow of Events:

3.a. If the details are not valid, the system displays invalid user message.

Post condition:
The system displays user’s home page.

Relationships:
NA

Activity Diagram:


System prompts to scan fingerprint
User scans fingerprint
Validate fingerprint
Invalid
System displays invalid user message
Valid
System open your home page

Special Requirements:
NA

4.1.2 Use Case – 2: Logout

Description:
The User can logout from system.

Actor:
The actors include the registered user.

Precondition:
The User should be logged in to the system.

Main Flow of Events:

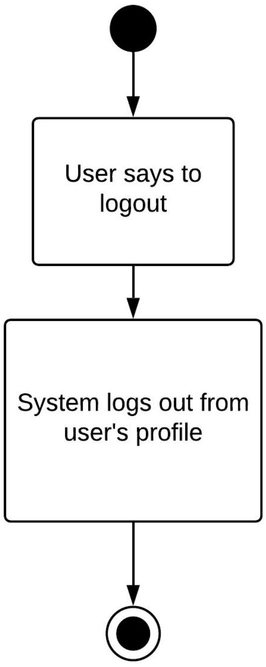
1. The user selects logout option.
2. System logs out from user’s profile.

Alternate Flow of Events:
NA

Post condition:
NA

Relationships:
NA

Activity Diagram:



Special Requirements:
NA

4.1.3 Use Case – 3: Voice notifications for new emails

Description:
The User will receive voice notifications for new emails.

Actors:
The actors include the registered users.

Precondition:

The User should be logged in to the system.

Main Flow of Events:

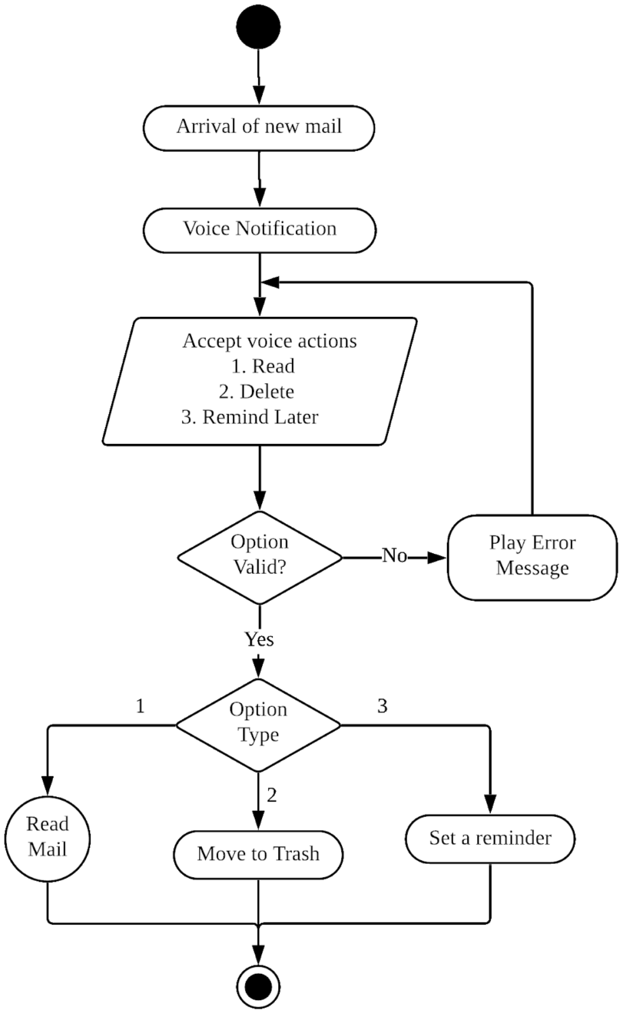
1. The user receives voice notifications for new emails.
2. The user can choose to read (hear) it immediately, delete it, or set a reminder to read later.
3. If user chooses to read it immediately, the system executes the read mail module.
4. If user chooses to delete it, the mail is moved to trash.
5. If the user chooses to set a reminder to read later, the system prompts to set the date and time at which the mail should be read.

Alternate Flow of Events:
NA

Post condition:
NA

Relationships:
NA

Activity Diagram:



Special Requirements:
NA

4.1.4 Use Case – 4: List inbox mails

Description:
System lists inbox mails.

Precondition:
The User should be logged in to the system and choose the list inbox action.

Main Flow of Events:

1. User chooses list inbox action.
2. Ask user if he needs to read the next mail.
3. If yes, system reads the address and subject of mails in the user's inbox in reverse chronological order.
4. If no, the user is directed to his dashboard.

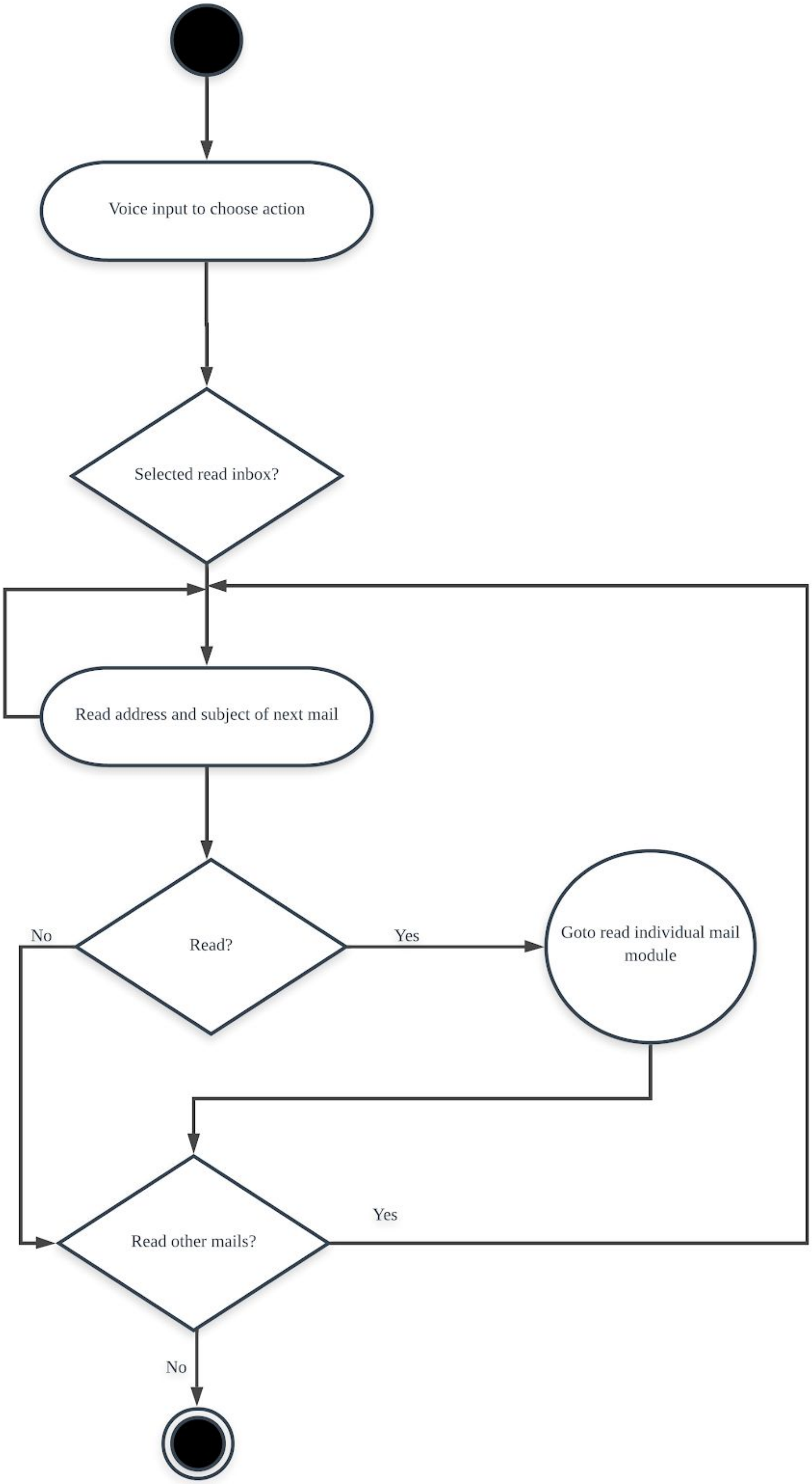
5. Ask if user needs to read that particular mail.
6. If no, then goto 2.
7. If yes, direct user to read individual mail module to read that mail.

Alternate Flow of Events:
Directed to read individual mail module if user chooses to read a particular mail.

Post condition:
NA

Relationships:
Excludes: Read Individual Mail

Activity Diagram:


Voice input to choose action
Selected read inbox?
Read address and subject of next mail
No
Read?
Yes
Goto read individual mail module
Yes
Read other mails?
No

Special Requirements:
NA

4.1.5 Use Case – 5: Listen to individual mail

Description:
User can listen to the individual mail.

Actors:
The actors include the registered user.

Precondition:
The User should have opted to listen to the mail.

Main Flow of Events:

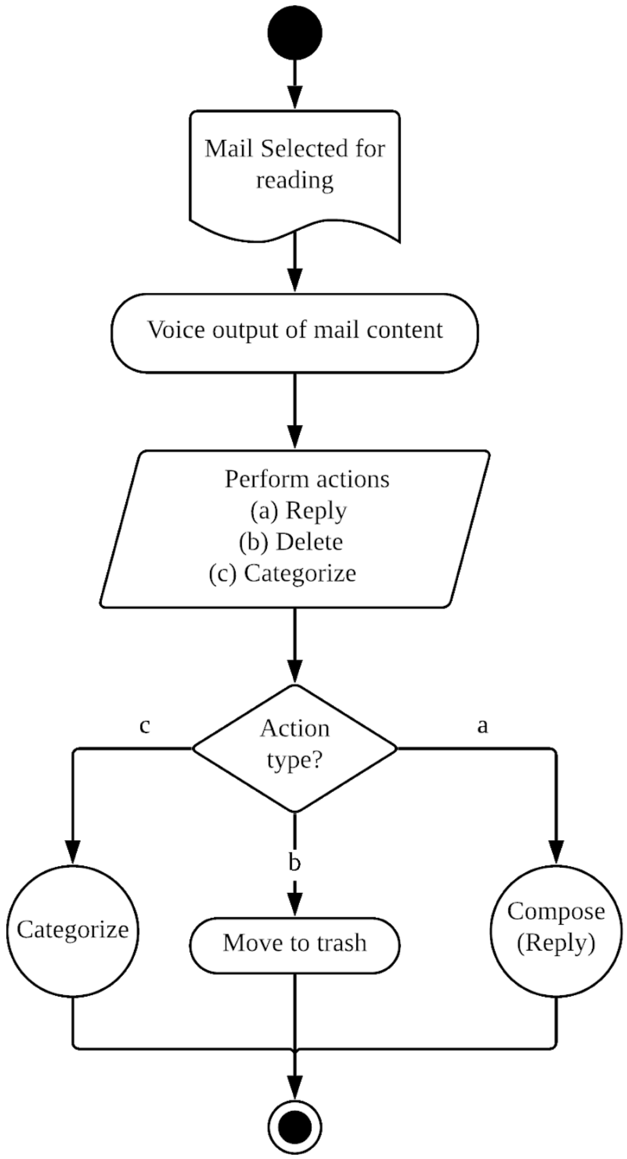
1. User opts to listen to the mail.
2. System redirects to the individual mail interface.
3. The sender, subject and the contents of the mail are read out.
4. The user can perform any action with mail, such as reply, delete or categorize it.
5. System performs the requested operation.
6. Returns to inbox.

Alternate Flow of Events:
NA

Post condition:
User receives confirmation of the task performed.

Relationships:
NA

Activity Diagram:



Special Requirements:
NA

4.1.6 Use Case – 6: Compose mail

Description:
The user can create mails by using speech which is later converted to text.

Actor:
The actors include the registered users.

Precondition:
The User should be logged in to the system and choose the compose mail action.

Main Flow of Events:

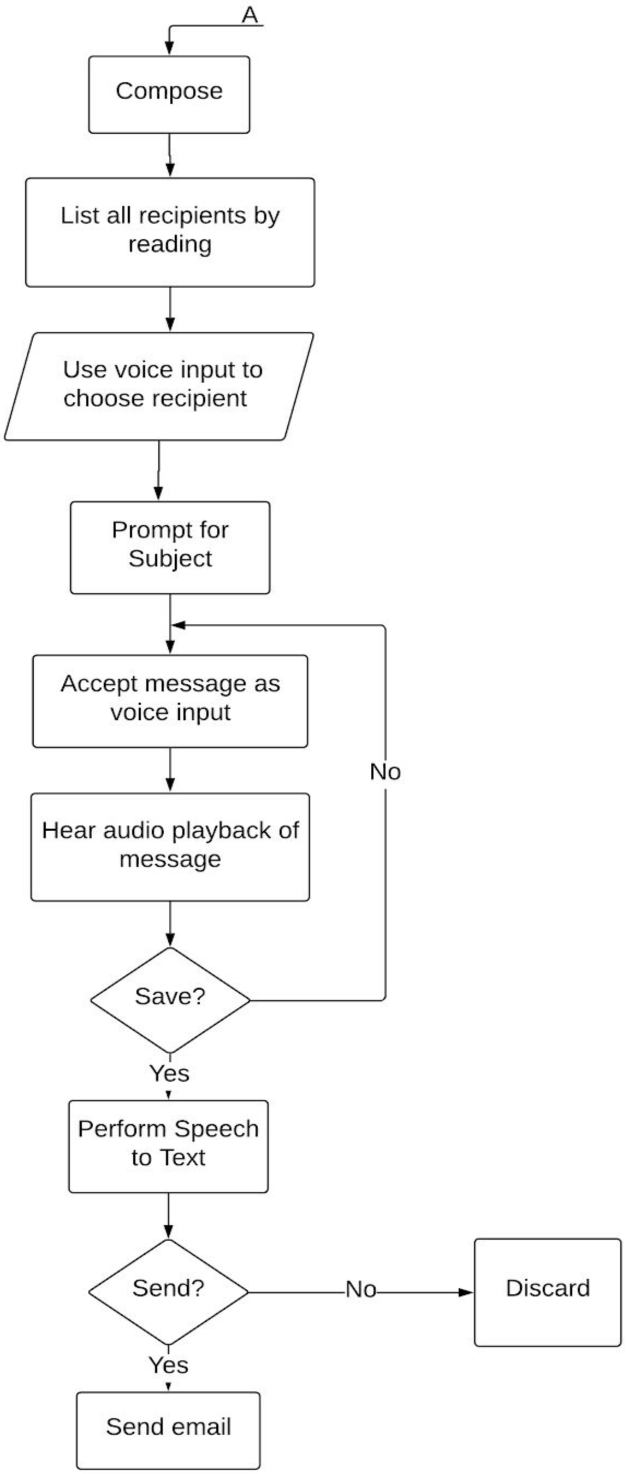
1. The user selects compose mail option.
2. System lists all recipients by reading.
3. User chooses the recipient(s) of his mail by means of voice input.
4. System prompts the user for subject.
5. System accepts message as voice input.
6. User listens to the audio playback of message.
7. User chooses to save if he is satisfied, else goto 5.
8. Perform Speech to Text.
9. User chooses to send or discard the mail.

Alternate Flow of Events:
NA

Post condition:
The mail if sent, gets added to the list of sent mails; else gets discarded.

Relationships:
NA

Activity Diagram:


A
Compose
List all recipients by reading
Use voice input to choose recipient
Prompt for Subject
Accept message as voice input
Hear audio playback of message
Save?
No
Yes
Perform Speech to Text
Send?
No
Discard
Yes
Send email

Special Requirements:

NA

4.1.7 Use Case – 7: Organize mails

Description:
The User can organize mails as spam or create labels.

Actor:
The actors include the registered users.

Precondition:
The User should be logged in to the system and choose the categorize mail action.

Main Flow of Events:

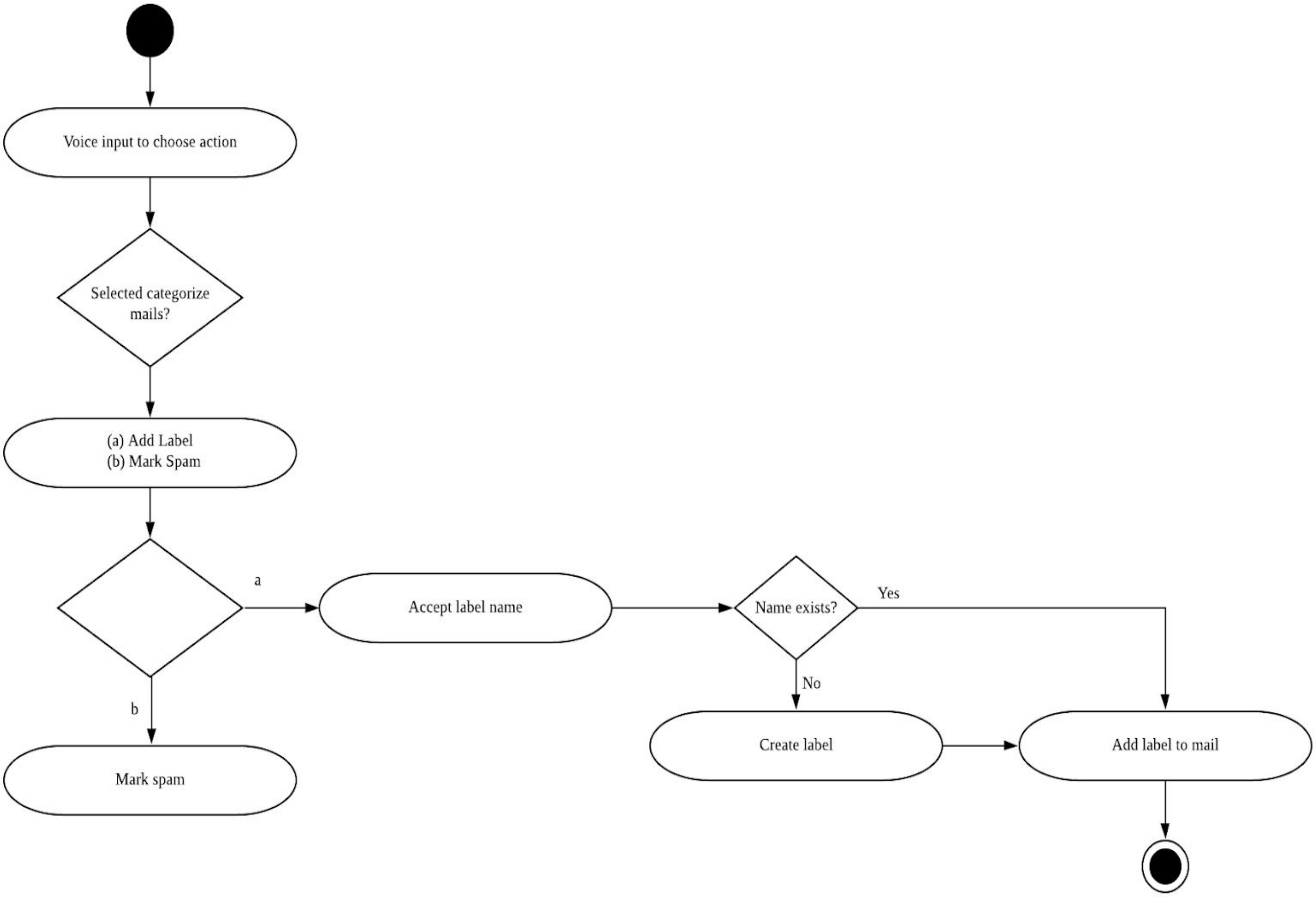
1. The user selects categorize mail option.
2. User either selects add label or mark as spam.
3. Accept label name if user selects “Add label”.
4. If label name already exists then add label to mail else create label name and then add.
5. Mark mail as spam if user selects mark as spam.

Alternate Flow of Events:
NA

Post condition:
Mails are categorized with specific labels or are marked as spam.

Relationships:
NA

Activity Diagram:



Special Requirements:
NA

4.1.8 Use Case – 8: View Trash

Description:
User can listen to the emails which are moved to trash and can restore them or delete them forever.

Actors:
Actors include the registered user.

Precondition:

The User should have moved any emails to trash.

Main Flow of Events:

1. User selects to view the trash.
2. System reads the deleted mails (if any).
3. User can choose to restore the mails or delete them forever.
4. Exit.

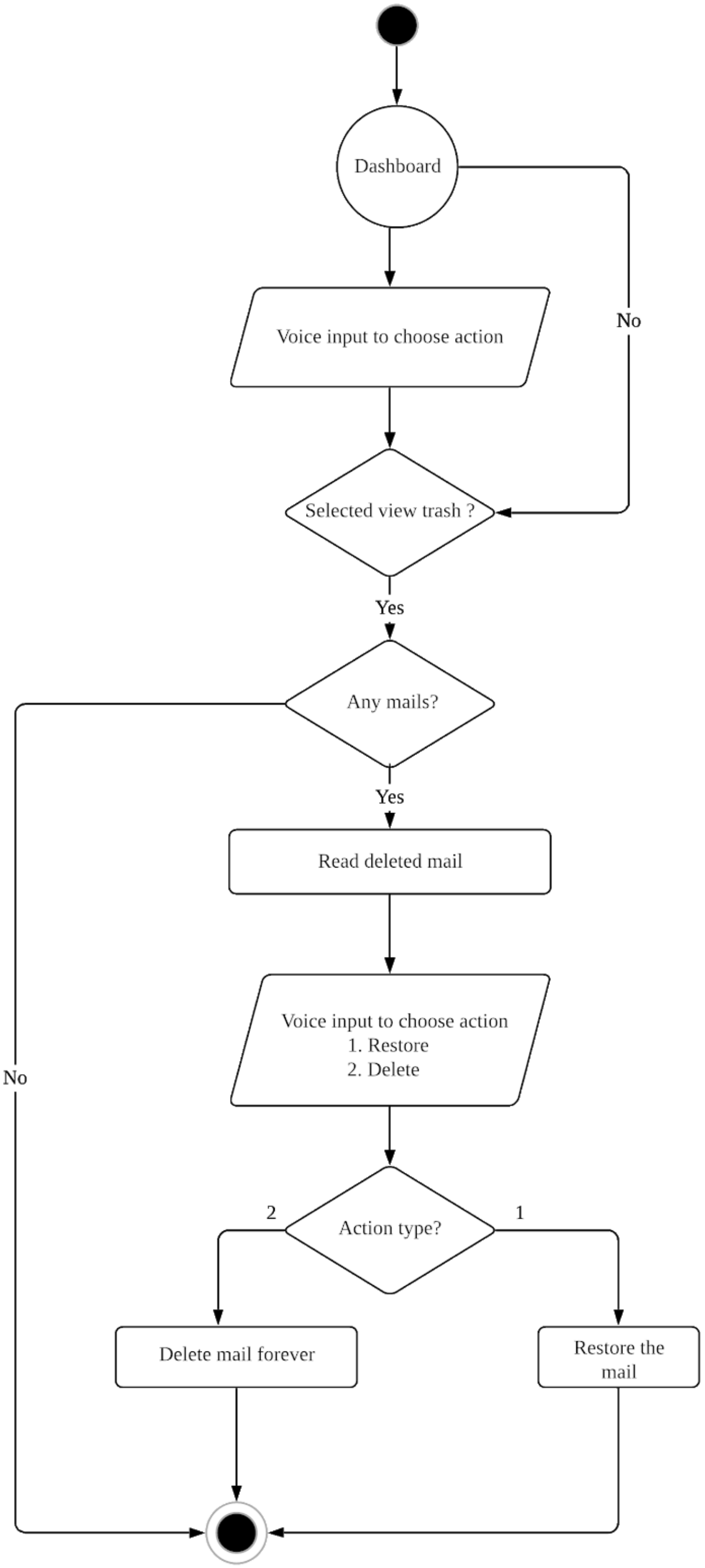
Alternate Flow of Events:

4.a. System reads the next mail in trash.
4.b. goto 3

Post condition:
NA

Relationships:
NA

Activity Diagram:



Special Requirements:

NA

4.1.9 Use Case – 9: View Sent Mails

Description:
User can view the sent mails.

Actors:
Actors include registered user.

Precondition:
The User should have sent at least one mail.

Main Flow of Events:

1. User selects "View sent mail".
2. System checks if there are any sent mails.
3. If yes, system reads sent mail.
4. Exit.

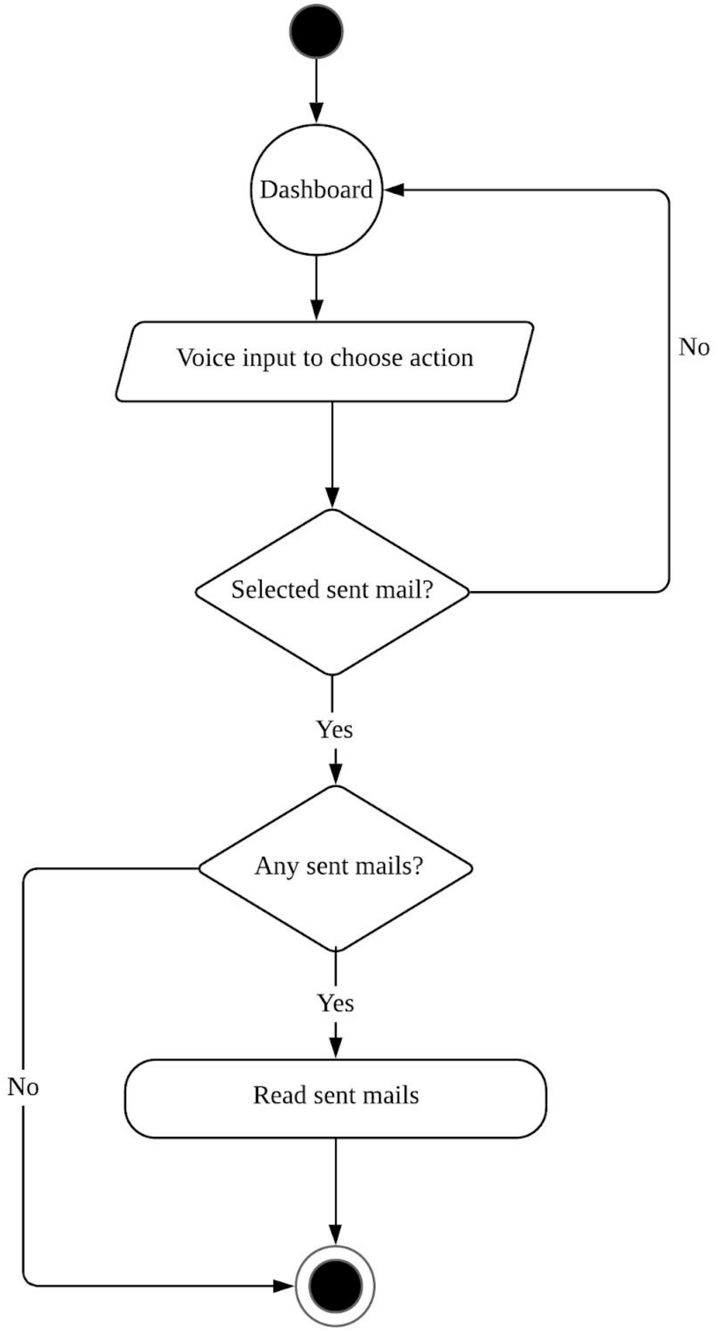
Alternate Flow of Events:

3.a. If no sent mail, goto 4.

Post condition:
NA

Relationships:
NA

Activity Diagram:



Special Requirements:

NA

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Major performance requirements are:

- All responses should be given without any latency.
- The system should be able to recognize different accents, and produce good speech to text output.

5.2 Security Requirements

Access to VBES will be restricted to the authorized users who have registered with their fingerprints. Guest users can register or sign up to the system with a valid mail id. The users will be able to login to the application using the username and password and access the modules or functions based on their requirements.