PROJECT SECRETARY

Final Software Report

FARHAN SAYED - *5573889*, FAIZA RAHMAN - *5580651*, ARSHA SHAJU - *5941611,* KESHAV MATHUR - *5773878*

Table of Contents

1. Introduction

Background

Virtual assistants have come a long way in assisting users with accomplishing tasks in every aspect of their lives. The major voice assistants are currently Siri and Google Assistant that serve approximately 3.5 billion users worldwide (Statista 2020), and an approximate of one billion devices that offer voice assistant services (Bera, 2019). However, despite the significant value that it has provided over the years, the voice assistants so far have only been able to aid users predominantly in personal aspects of their lives and some aspects of their work. With our Project, we plan on helping people by providing them with a platform that organizes information for them, reminds them of their schedule and aids them in tasks that are workplace oriented. Additionally, since it is evident that amongst 4 billion email users (Statista,2020), "Radicati's data shows that an email user has roughly two accounts on average. a high percentage of users have a minimum of three email accounts, all the way up to nine, according to a Microsoft study. Apart from separating work and personal emails, people often have a separate account for filtering out spam messages, too" (99firms.com, 2019). From this, we also see that people keep separate accounts primarily for privacy-oriented reasons and for filtering out spam messages. So, with our project, we also intend on encouraging users to keep or create a separate workplace-oriented account whilst working with our Web application to ensure the safety of the user's personal/sensitive information.

Problem statement

Users have access to a virtual secretary that is available to them whenever they want as opposed to having an actual secretary that must be paid and is available for a certain time period.

Users will be notified beforehand as to who will be attending a certain meeting which after knowing that not enough individuals are attending, the meeting can be canceled well ahead before the rest of those who were planning on attending spend time and resources to get to the venue.

Users can store files and documents using the file system that the Web application provides and have them sorted and retrieved in almost no time as opposed to having an actual secretary who will take significantly more time especially when there are too many documents to sort and retrieve.

Our solution

An AI-based Office Secretary mobile application to facilitate and organize their office tasks and other responsibilities in a systematic fashion. The application utilizes a chat-bot integrated with a voice assistant to aid the user in organizing their work-life. Based on the command given to the application, it can execute the tasks that the user has for the day, set reminders, alarms, tasks and set appointments. Additionally, the application would also allow the user to make reservations for an Uber. The application also accesses the contacts, email and calendar of the user. The application will also provide a file system that can store and sort files for the user. The user can then use a search function to retrieve whatever he/she wishes to and view them with cluster visualization. Finally, they can record their meeting audio which will be then converted to text and then summarized. The chat bot provides easy navigation for the user making the experience even more relaxing and easy. Usability goals and principles were kept in mind while building the GUI creating a smooth, easy and safe application for the users.

Target niche

This application will be targeting offices and workplace environments. This Web application is not restricted to age, location, gender, income level. Educational level, occupation and ethnicity. This Web application can be used by any employee in any workplace or office environment. Since it's an application, it is portable and easily accessible on any device.

Technologies

To build the application we used Android Studio as our core platform where we used Java to develop the functionality of our application and XML to design the application. We used Python 3.7, flash and Node.JS programming language to develop the filesystem, python 3.7 and rasa for the voice assistant and Ngrock to host the bot publicly. Android's google speech to text and text to speech API was used to convert speech to text and text to speech. To build the note maker python 3.7 programming language was used along with Firebase as the real time cloud database and Google APIs to convert the recordings to text. Firebase authentication was used to authenticate the user Log in and Sign Up.

2. Feasibility

Project charter

The purpose of the project is to develop an office oriented virtual assistant, Sarah that can assist employees with their daily office tasks while reducing operational cost and time. The project started on February 5th, 2020 and will commence on the 9th of December 2020. The success of the project is based on developing the promised features and deliverables such as design document, requirements document, test plan etc., after getting the approval from the stakeholders which will include, the panel members of the department of Information Technology and Computer science at the University of Wollongong, team members (Arsha Shaju, Keshav Mathur, Farhan Sayed & Faiza Rahman) and the users. At the end of the project an android based mobile application will be successfully build that will carry out all the promised features and services.

Scope statement

The world has always wondered how AI robots or intelligence would contribute to the success and empowerment of the world. According to an article by (Nanalyze, 2017) in the U.S., there are almost four million workers which fall into the general category of secretaries and administrative assistants. Bureau of Statistics for Employment (BLS). Around 2014 and 2024, the BLS expects sluggish job growth ahead at 3 percent. They are required to coordinate appointments, take meeting notes, manage email, organize files and produce reports. Which can all be taken over by Artificial Intelligence soon. AI can work all the time and does not require breaks or rests, working efficiently compared to a human secretary. Additionally, it is very cost-efficient as an average secretary has a salary of 31,000 US dollars and does not require any additional benefits (Meet Sally, 2017). The purpose of this project is to facilitate the smooth running of offices and save them a lot of time and money.

Goals and objectives

Our goal is to develop an android mobile application which can provide office secretary like assistance. It has various features which include automated summarization of meeting notes, voice response to emails, ordering food, fixing rides, setting automated reminders and automatically organizing & managing their file systems all using a simple voice command. This will aid in increasing operational and working efficiency of employees in the working sector by reducing the cost and saving the time of the users. This project is developed for the office community because it provides them an A.I Secretary that will carry out the works of a real-life secretary and get daily assistance.

PESTEL analysis

- Political: Upon publishing the app on play store, it must follow the rules, regulations and policies of play store. There is a need to maintain a firm standard of the laws set out by the play store with consistency. Also, many different markets across the world have different set of regulations that are either relaxed or are either stringent. The app must be customized to satisfy the rules of various countries or release different versions in different countries. The competitor solution comes free when buying any android or iOS device while our application has some features that can only be unlocked while using the premium package where ads will be removed. Most importantly, cross border situations are starkly different therefore Sarah must stay in line with all those policies and changes so that they can adapt to all those changes accordingly.
- Economical: the recent economic downturn has plagued the economy, companies had to restructure their sales and marketing campaigns greatly. Also, with diminishing profits they had to undergo downsizing internally and re-think upon how to penetrate the market. Economic conditions have the highest influence on a business, regardless of what trade it is in. Though, in Sarah’s favor, the economic downturn that started in 2020 will result in increased sales as the app can replace office secretaries which will reduce company cost

- Social: Sarah must communicate its image as a global brand so that the people can associate it with themselves as something that connects the various offices together. The need to develop and maintain customer loyalty is important. Replacing office secretaries with our application might be presumed as a risk and would create uncertainties, there is a need to educate the users on the application and how it can make a difference in their organization.
- Technological: With the advent of the new age in technology, companies have completely integrated themselves with all the recent changes that have taken place. To mention a recent trend that has greatly picked up and something that almost every business is turning toward is Social Media. The social media explosion has allowed for increasingly interactive engagement with the consumers with real time results so Sarah must stay ahead of all the developments that take place with keeping in view how the youth of today utilizes technology for their benefit and how can Sarah reach them in order to keep on increasing brand recall and brand engagement.
- Environmental: The use of Renewable technologies and lower level of battery usage can create a positive brand image as there is a higher interest in preserving the environment.
- Legal: There can be many legal implications upon the technology industry due to loss of data, privacy leaks, hackers. Most of the user data stored in the database is confidential and important which can create many legal implications if fallen into the wrong hands. Sarah must run multiple tests to find any loopholes or flaws in security and fix it immediately.

Competitive analysis

Our main competitors are Siri, Alexa, Bixby and Google assistant. This section will analyze the competitor solutions, limitations and how our product is different.

Existing Solutions

- Siri: Siri is an intelligent personal assistant, voice-controlled, accessible to Apple users. The goal is to speak to her like you'd be a buddy and she's aiming to help you get things done, whether it's making a reservation for dinner or sending a message. She can read your last email briefly, phone your friend to tell them you're running late, change your road trip playlist on Apple Music, let you know what movies are playing today, locate a three table in London or call your dad at college. She can also inform you were a good restaurant is nearby, flip a coin, locate a certain author's books, set an alarm, give you directions and even set a reminder that will only activate if it knows that you are in or missed a certain place (for example, as you leave work, pick up dry cleaning). She's going to tell you what the weather is like tomorrow, switch events, tell you when the next game is for your party, and you can even get her to roll a die, along with several other things including showing her how to say your name or asking her which family members are your contacts. Even though Siri displays a lot of features and has many advantages it comes with limitations which include:

    - Siri will not work without an internet connection: One must be either wired to a Wi-Fi network or to cellular data. This is one of the speech assistant's biggest flaws. The whole story behind voice assistants is quite complicated, although everything seems quite plain. It all depends on cloud computing, and when you're down, you can't use Siri.
    - Siri has performance flaws when it comes to listening to the voice command: Siri functions perfectly fine most of the time, but to get Siri to hear you and really do what you want, you need to talk out plainly. Siri won't understand you if you're moving too fast, or with a strong accent.

(O'Boyle 2019 & Zujic, 2020).

- Alexa: Alexa Voice Service (AVS) currently lives in the cloud. Amazon's AVS is advanced voice recognition and a tool for interpreting natural language. The app can be used for any connected device that has a microphone and speaker to allow speech. "With machine learning, Alexa is always getting smarter with new capabilities," reads Amazon's Developer section. The number of instructions that Alexa will comprehend continues to expand every day. Amazon considers those "skills," and now with Amazon Blueprints, you can even develop your own skills. Since testing Amazon, you can even post your own abilities for other users to use on their Alexa apps in the Amazon Abilities Store. Alexa also can set up "routines," where a single order you create — say, "Alexa, goodnight,"— shuts off the lights, locks your front door, sets an alarm for a period you specified, and schedules your coffee pot to turn on at some point. Alexa comes with many smart functions and features, but it also has limitations. They are:
    - Updating a new version of the program onto the computer takes several days.
    - Many privacies flaw
    - People can identify who is who at home based on private echo interactions.
    - Others have links to echo talks, too.
    - Users can use different techniques to use the position to offer location-based services.
    - Since Alexa is a cloud-based service any issues or software malfunctions in the cloud can directly affect the functions of Alexa.
    - To use echo must be plugged in as it does not have an internal battery.
    - Amazon maintains digital recordings of audio spoken by customers after the phrase "wake up".

(Wetzel & Rawes 2020 & Ratnesh 2018)

- Bixby: Samsung Bixby, the artificially intelligent agent used on the company's flagship phones, is no question somewhat distinct from Apple's Siri, Alexa's Amazon, and Google Assistant. Attempts by Samsung to predict the wishes and provide contextualized alerts. Bixby filters images, sends messages and command-composes e-mails. Everything that you do via the contact on your screen, you should be able to do with Bixby that voice. It can also monitor certain smart home appliances and devices running the Internet of Things (IoT), including smart fridges, televisions, and hundreds of other Samsung devices. Samsung claims Bixby follows more than 3,000 commands from Twitter, Snapchat, YouTube, Uber, Gmail, Google Maps and more app-specific commands. It even integrates with the SmartThings software from Samsung to provide buttons that monitor smart light bulbs and door locks, shortcuts to Web applications that you use most often, and more — everything based on the time of day, place, and other variables. Bixby also has a routine feature like Alexa (Jansen 2019). According to an article by (Rodriguez 2017) Bixby is a smart system but comes with its limitation:

- Server less Extensibility: Compared to its competitors, Samsung's server fewer computing capabilities are very limited and that could influence the long-term growth of Bixby.
- Lack of NLP platform: One drawback of Bixby is the lack of a Windows LUIS or Google's API.ai-like transparent NLP-AI interface that can be used to introduce
- complex conversational interfaces and add new capabilities that can be learned and improved over time.
- Cognitive services: Technologies such as Cortana, Alexa or Google Assistant not only have strong NLP features but also a range of cognitive tools in relevant fields such as image processing, speech analysis or video analytics. Samsung needs to find a way to catch up in this area.

- Google Assistant: Google Assistant is the speech adjutant for Android. Google Assistant was an expansion to Android Now when it debuted, intended to be personal and building on the "OK Google" voice controls on Google. Google Assistant provides voice commands, voice search and voice-activated system access, helping you to perform several tasks after you've said the words "OK Google" or "Hi, Google" wake up. It has many features like:
  - Google Home.
  - View schedule records and other personal information.
  - Find online content, from restaurant bookings to maps, weather and news.
  - Guide your music Play material on your Chromecast or other apps that are compatible.
  - Control timers and callbacks.
  - Make appointments and submit updates.
  - Open Web applications installed on your device
  - Read the notifications

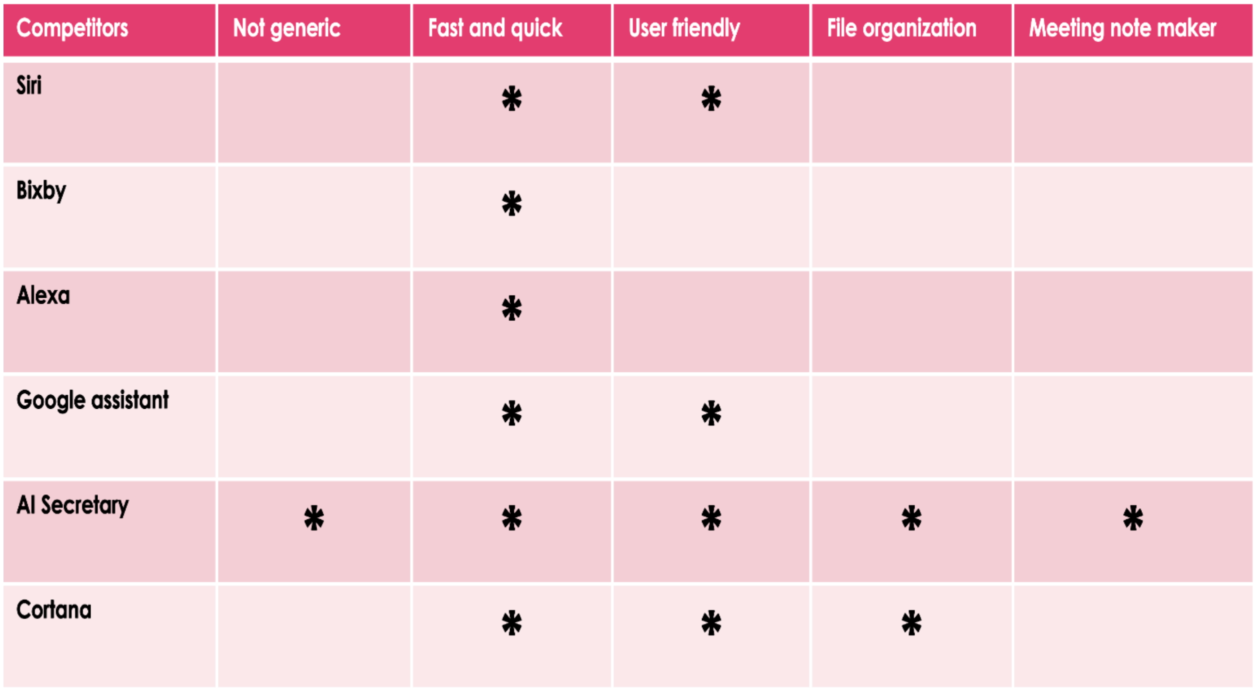
It also can check in on your trip (airline and destination dependent), as well as the ability to reserve a space with some friends, and Google Home products have an Interpreter Mode and smart screens as well. With that, you might ask the Google Assistant to help you speak in dozens of languages. (O'Boyle & Tillman 2019)

This perfect-looking system also comes with limitations. According to (AEO 2019) they are:

- It takes up a lot of data.
- Constant monitoring and lack of privacy.
- Google will have to be permitted to view all your details, including travel schedule, images, videos, calendar schedule and more, in full so that it can continue to complete the basic tasks for you.

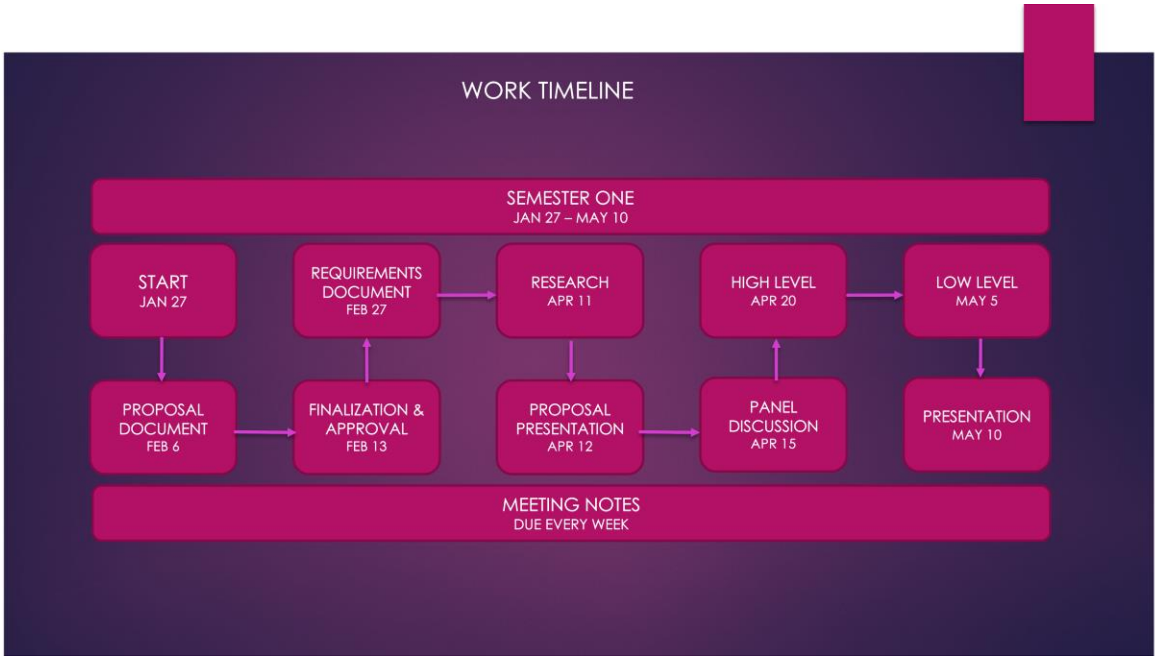
Our solution:

In comparison to the above solutions our product is not generic, its fast and quick, User friendly. The user interface is build following all the usability goals and principles and customized for an office environment with the use or relaxing animation, colors and fonts. Our solution enhances the working experience by making the process smooth while reducing cost and saving time. It has an in-build file system where the user can upload their files, view them by using cluster visualization. Now they can search for files based on content rather than on file names. The app also has an in-build meeting summarizer that can summarize important or any meetings audio to text and into summarized points. It also offers services such as Uber, Contact, Calendar and Email.



Competitors	Not generic	Fast and quick	User friendly	File organization	Meeting note maker
Siri		✱	✱		
Bixby		✱			
Alexa		✱			
Google assistant		✱	✱		
AI Secretary	✱	✱	✱	✱	✱
Cortana		✱	✱	✱	

Work breakdown structure



WORK TIMELINE

SEMESTER ONE
JAN 27 – MAY 10

START
JAN 27

PROPOSAL DOCUMENT
FEB 6

FINALIZATION & APPROVAL
FEB 13

REQUIREMENTS DOCUMENT
FEB 27

RESEARCH
APR 11

PROPOSAL PRESENTATION
APR 12

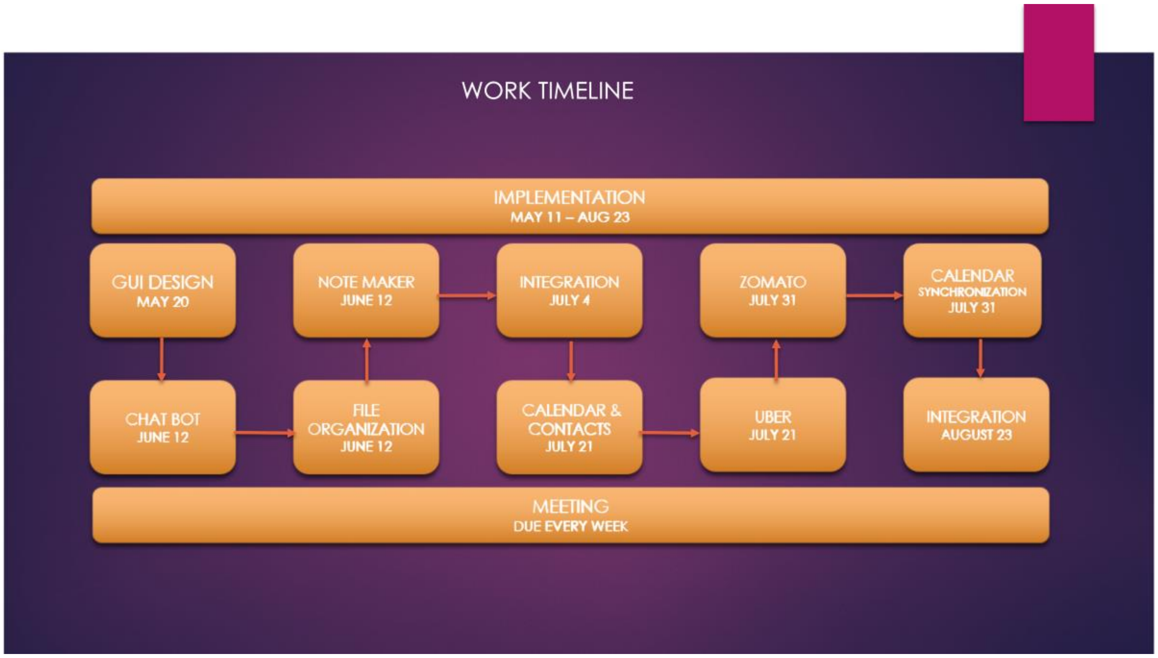
PANEL DISCUSSION
APR 15

HIGH LEVEL
APR 20

LOW LEVEL
MAY 5

PRESENTATION
MAY 10

MEETING NOTES
DUE EVERY WEEK



WORK TIMELINE

IMPLEMENTATION
MAY 11 – AUG 23

GUI DESIGN
MAY 20

CHAT BOT
JUNE 12

FILE ORGANIZATION
JUNE 12

NOTE MAKER
JUNE 12

INTEGRATION
JULY 4

CALENDAR & CONTACTS
JULY 21

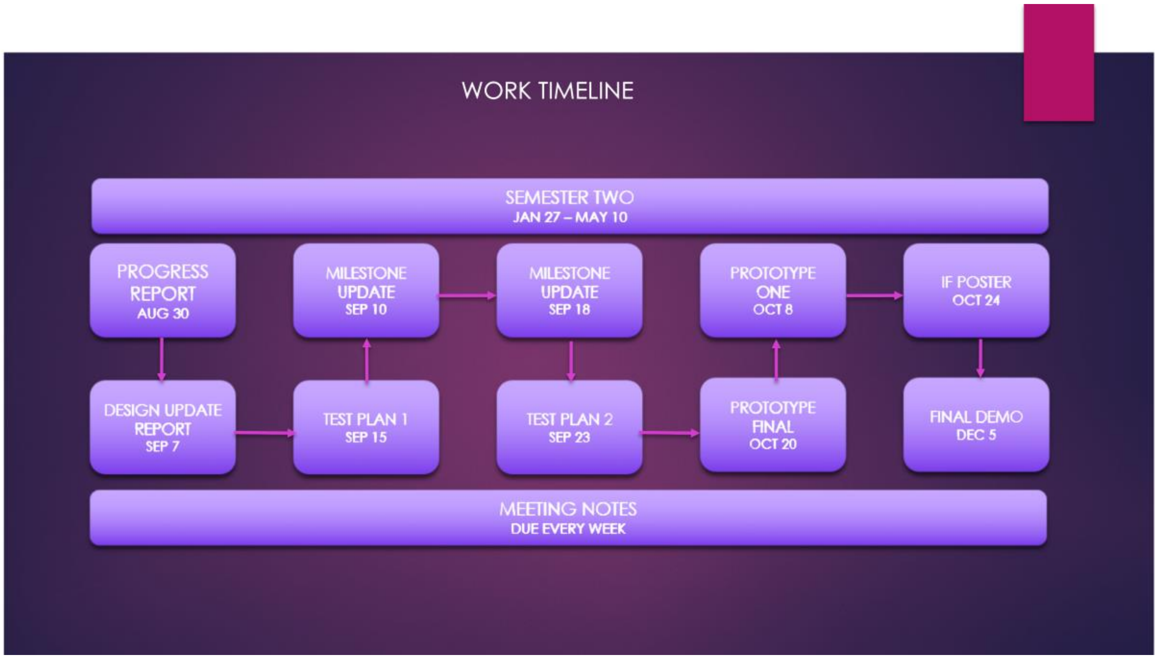
UBER
JULY 21

ZOMATO
JULY 31

CALENDAR SYNCHRONIZATION
JULY 31

INTEGRATION
AUGUST 23

MEETING
DUE EVERY WEEK



WORK TIMELINE

SEMESTER TWO
JAN 27 – MAY 10

PROGRESS REPORT
AUG 30

MILESTONE UPDATE
SEP 10

MILESTONE UPDATE
SEP 18

PROTOTYPE ONE
OCT 8

IF POSTER
OCT 24

DESIGN UPDATE REPORT
SEP 7

TEST PLAN 1
SEP 15

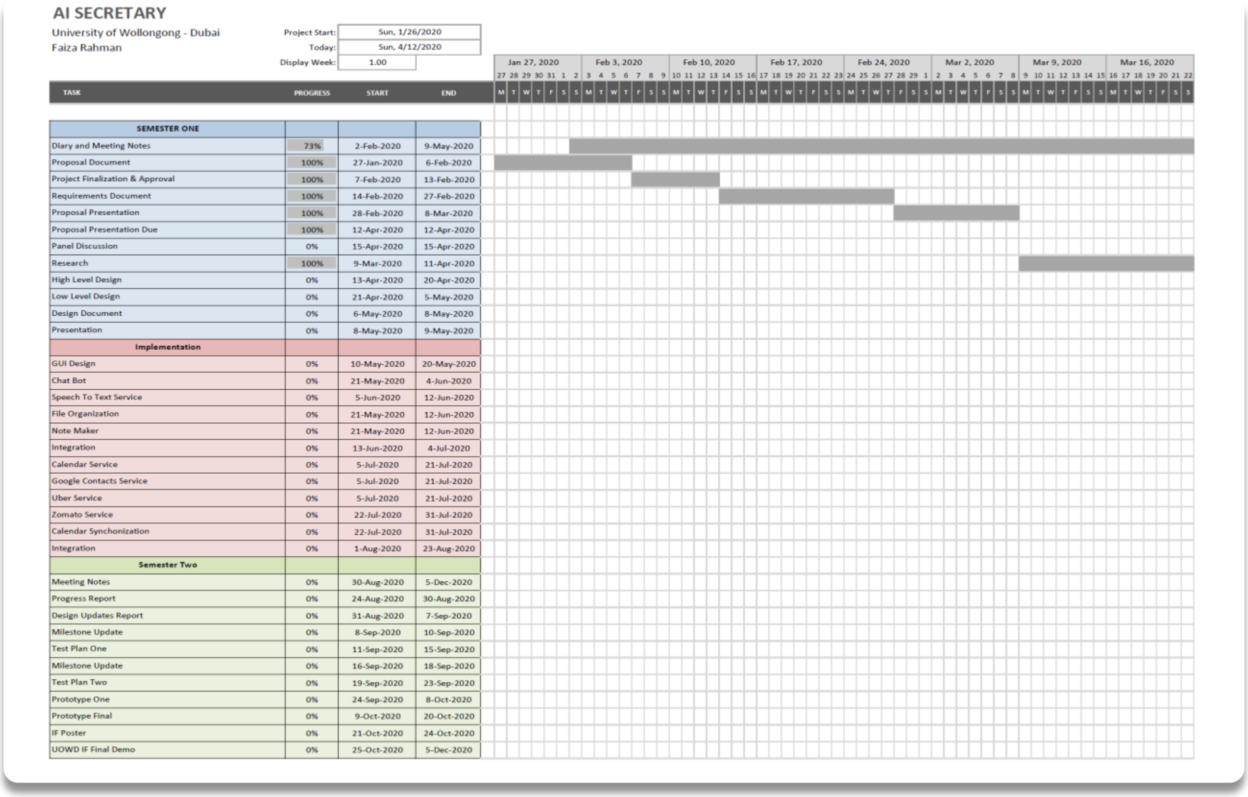
TEST PLAN 2
SEP 23

PROTOTYPE FINAL
OCT 20

FINAL DEMO
DEC 5

MEETING NOTES
DUE EVERY WEEK

Milestones and division of work among team members

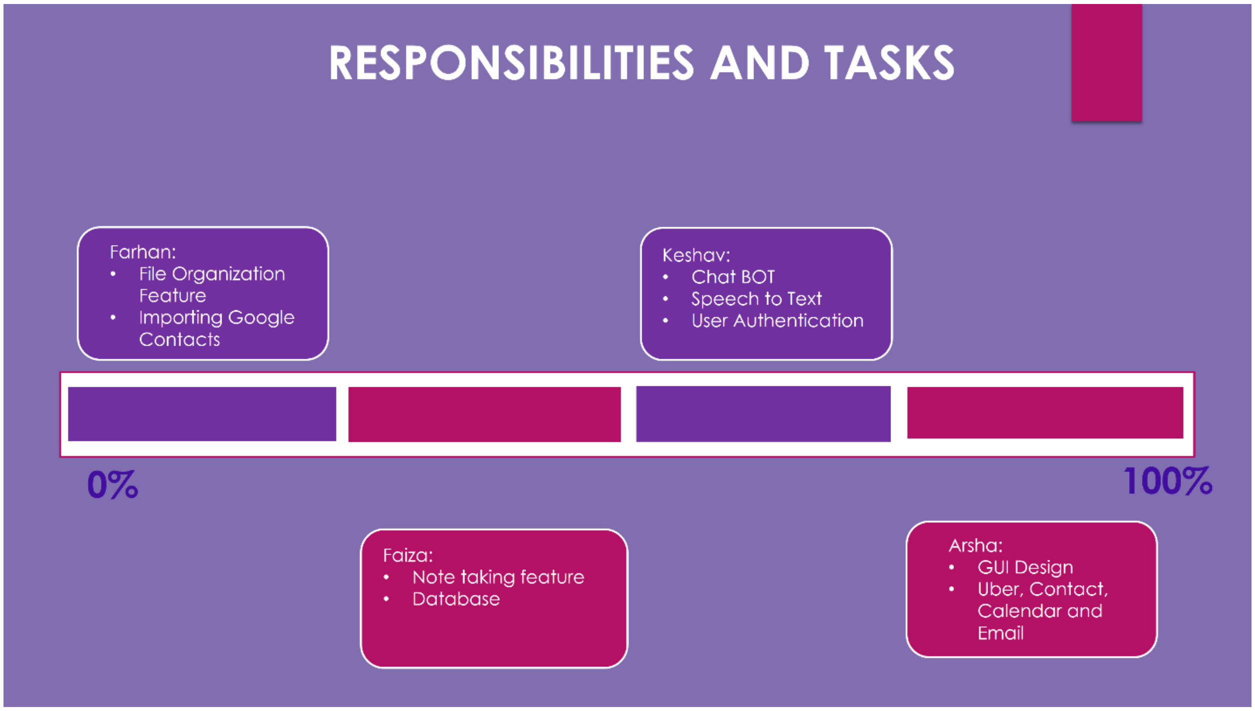


AI SECRETARY

University of Wollongong - Dubai
Faiza Rahman

Project Start: Sun, 1/26/2020
Today: Sun, 4/12/2020
Display Week: 1.00

TASK	PROGRESS	START	END
SEMESTER ONE			
Diary and Meeting Notes	73%	2-Feb-2020	9-May-2020
Proposal Document	100%	27-Jan-2020	6-Feb-2020
Project Finalization & Approval	100%	7-Feb-2020	13-Feb-2020
Requirements Document	100%	14-Feb-2020	27-Feb-2020
Proposal Presentation	100%	28-Feb-2020	8-Mar-2020
Proposal Presentation Due	100%	12-Apr-2020	12-Apr-2020
Panel Discussion	0%	15-Apr-2020	15-Apr-2020
Research	100%	9-Mar-2020	11-Apr-2020
High Level Design	0%	13-Apr-2020	20-Apr-2020
Low Level Design	0%	21-Apr-2020	5-May-2020
Design Document	0%	6-May-2020	8-May-2020
Presentation	0%	8-May-2020	9-May-2020
Implementation			
GUI Design	0%	10-May-2020	20-May-2020
Chat Bot	0%	21-May-2020	4-Jun-2020
Speech To Text Service	0%	5-Jun-2020	12-Jun-2020
File Organization	0%	21-May-2020	12-Jun-2020
Note Maker	0%	21-May-2020	12-Jun-2020
Integration	0%	13-Jun-2020	4-Jul-2020
Calendar Service	0%	5-Jul-2020	21-Jul-2020
Google Contacts Service	0%	5-Jul-2020	21-Jul-2020
Uber Service	0%	5-Jul-2020	21-Jul-2020
Zomato Service	0%	22-Jul-2020	31-Jul-2020
Calendar Synchonization	0%	22-Jul-2020	31-Jul-2020
Integration	0%	1-Aug-2020	23-Aug-2020
Semester Two			
Meeting Notes	0%	30-Aug-2020	5-Dec-2020
Progress Report	0%	24-Aug-2020	30-Aug-2020
Design Updates Report	0%	31-Aug-2020	7-Sep-2020
Milestone Update	0%	8-Sep-2020	10-Sep-2020
Test Plan One	0%	11-Sep-2020	15-Sep-2020
Milestone Update	0%	16-Sep-2020	18-Sep-2020
Test Plan Two	0%	19-Sep-2020	23-Sep-2020
Prototype One	0%	24-Sep-2020	8-Oct-2020
Prototype Final	0%	9-Oct-2020	20-Oct-2020
IF Poster	0%	21-Oct-2020	24-Oct-2020
UOWD IF Final Demo	0%	25-Oct-2020	5-Dec-2020



RESPONSIBILITIES AND TASKS

Farhan:
- File Organization Feature
- Importing Google Contacts

Keshav:
- Chat BOT
- Speech to Text
- User Authentication

Faiza:
- Note taking feature
- Database

Arsha:
- GUI Design
- Uber, Contact, Calendar and Email

SWOT analysis

Strengths:

- Sarah utilizes a voice assistant to access many of its features which are inaccessible by other voice assistants which not only brings added value to our product but also makes our product easier to use and navigate across.
- Sarah's file system utilizes a cluster-based approach which enables the user to search for documents based on its content unlike other file managing application which only permits the user to search based on the name of the file itself.
- Sarah's note taking feature enables the user to record the meeting and have the application provide summarized bullet points of the meeting back to the user. Unlike other note taking applications in which the user notes down what he/she believes is important. As a result, we can see that our application in a nutshell takes notes for the user instead of the user having to do it manually.
- The domestic market in which Assistants is operating is both a source of strength and roadblock to the growth and innovation of the company. Assistant Voice can easily grow in its domestic market without much innovation but will require further investment into research and development to enter international market. Sarah has a great future in the domestic market.

Weakness:

- Since we have implemented a web-service architecture, setting up each of the components of the application can be a rather long and a tedious task.
- Our application's voice assistant is limited in terms of mainstream functionalities that other voice-assistants such as Google, Siri and Alexa possess.
- The file system of our application may not always be accurate in terms of search results simply because the application cross-checks documents containing the words which have the same TF-IDF Score as the word searched instead of utilizing a string-matching technique.
- Does not allow password reset via SMS
- Does not have access to wider market segments
- Need for excessive and costly testing

Opportunities:

- The products and brand portfolio of Sarah is enabling it to target various segments in the domestic market at the same time. This will enable Sarah to build diverse revenue source and profit mix.
- Use of various social media platforms data to target perfect customers for the application, promote the need for the applications, lower marketing costs.
- Expansion into international markets, larger customer pool.
- Developments in artificial intelligence, Iot (Internet of things) and Iob(Internet of bodies) which will assist in enhancing the product meeting customer needs and demands.

Threats:

- Sarah operates in an environment where it faces numerous regulations and government laws. Sarah needs to navigate environment by building strong relationship with lobby groups and political network.
- Government rules, laws, regulations and bureaucracy.
- Loss of intellectual and confidential property.
- Security hacks and issues.
- High competition in the market segment with regard to cost and features.
- Loopholes can be found in code.

Risk management plan

The project team will ensure that risks are actively identified, analyzed, and managed throughout the life of the project. Risks will be identified as early as possible in the project to minimize their impact. Risk identification will involve the project team, appropriate stakeholders, and will include an evaluation of environmental factors, organizational culture and the project management plan including the project scope. Careful attention will be given to the project deliverables, assumptions, constraints, WBS, cost/effort estimates, resource plan, and other key project documents. All risks identified will be assessed to identify the range of possible project outcomes. Qualification will be used to determine which risks are the top risks to pursue and respond to and which risks can be ignored. The probability and impact of occurrence for each identified risk will be assessed by the project team, with input from the project team based on the probability and impact the risk can cause.

- High Risk – Greater than 70% probability of occurrence. Risk that has the potential to greatly impact project cost, project schedule or performance
- Medium Risk – Between 30% and 70% probability of occurrence. Risk that has the potential to slightly impact project cost, project schedule or performance
- Low Risk – Below 30% probability of occurrence. Risk that has relatively little impact on cost, schedule or performance

Risks that fall within the high and medium zones will have risk response planning which may include both a risk mitigation and a risk contingency plan. Analysis of risk events that have been prioritized using the qualitative risk analysis process and their effect on project activities will be estimated, a numerical rating applied to each risk based on this analysis, and then documented in this section of the risk management plan. Each major risk (those falling in the High & Medium zones) will be assigned to a project team member for monitoring purposes to ensure that the risk will not "fall through the cracks". For each major risk, one of the following approaches will be selected to address it by eliminating the threat by eliminating the cause, identifying ways to reduce the probability or the impact of the risk, accepting the outcome and transfer the responsibility of the risk using insurance or outsourcing. For each risk that will be mitigated, the project team will identify ways to prevent the risk from occurring or reduce its impact or probability of occurring. This may include prototyping, adding tasks to the project schedule, adding resources, etc. For each major risk that is to be mitigated or that is accepted, a course of action will be outlined for the event that the risk does materialize in order to minimize its impact. The level of risk on a project will be tracked, monitored and reported throughout the project lifecycle.

A risk list will be maintained by the project team and will be reported as a component of the project status reporting process for this project. All project change requests will be analyzed for their possible impact to the project risks.

3. Requirement analysis

Definitions and assumptions

The mobile application will be built for android mobiles using android studio with JAVA and XML as the language. XML will be used to create responsive, interactive elements for each activity pages, enhancing the user experience. it will be used to create presentation of the mobile application, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens or small screens. It is dependent on JAVA for making the applications function well. Firebase will be used for storage and authentication purposes. The project has duration from 2nd February 2020 – 9th December 2020 and the work are distributed evenly among the team members based on their skills and areas of expertise. Any necessary skills or additional requirements will be learned along the way to successfully complete the project

Chat Bot table

Field	Type	Description
requestID	String	Primary Key
userRequest	String	The request sends in by the user
Functionality to be called	String	The function the user wants to open

File Table

Field	Type	Description
ID	String	Primary Key
size	String	The size of the file
fileName	String	The name of the file
type	String	The type of the file

Cluster Table

Field	Type	Description
ID	String	Primary Key
documentName	String	The name of the document

type	String	The type of document

Folder Table

Field	Type	Description
ID	String	Primary Key
size	String	The size of the folder
folderName	String	The type of folder

Design constraints

The application should be build using JAVA, python, flask, XML and NodeJS programming languages, hosted on the internet using ngrock, supported by RASA and Google API and user data and app data is stored on Firebase's real-time database. The application will be built on Android Studio and released on play store upon completion. Google's material design will be used while designing the application. The application will be built following usability goals and principles.

Programming Language	Servers	API	Database
Java	ngrock	rasa	Firebase (real-time)
Python		Google API	
flask			
XML			
NodeJS			

Functional requirements

Login

Login	
Function	Login
Description	Allows user to log in
Inputs	Username and password
Source	An android device.
Outputs	Access to the application
Destination	Cloud Storage, Application, Firebase
Action	A function that allows the user to login to the system using the previously registered username and password
Requires	the username and password must watch
Pre-condition	The user should have a valid username and password
Post-condition	The user is granted access
Side-effects	N/A

Use case	

Signup

Sign Up	
Function	Sign Up
Description	Allows user to sign up
Inputs	Username, password, email and phone number
Source	An android device.
Outputs	Access to the application
Destination	Cloud Storage, Application and Firebase
Action	A function that allows the user to login to the system using the previously registered username and password
Requires	the username and password must watch
Pre-condition	The user should have a valid username and password
Post-condition	The user is granted access
Side-effects	N/A
Use case	

Permissions

Access microphone and audio	
Function	Permission to access the device microphone and audio
Description	Sends the user a request to access the microphone and audio
Inputs	Accept or decline
Source	An android device.
Outputs	Accept– access granted, decline – access denied
Destination	Cloud Storage, Application
Action	A function that requests the user to grant the application permission to access their microphone and audio
Requires	A working microphone
Pre-condition	The user’s device should have a microphone that functions well
Post-condition	The access is granted
Side-effects	N/A
Use case	

Access storage	
Function	Permission to access the device storage
Description	Sends the user a request to access the storage
Inputs	Accept or decline
Source	An android device.
Outputs	Accept– access granted, decline – access denied
Destination	Cloud Storage, Application
Action	A function that requests the user to grant the application permission to access their storage
Requires	A storage system on users' device
Pre-condition	The user's device should have a storage system
Post-condition	The access is granted
Side-effects	N/A
Use case	

Access files	
Function	Permission to access the device files
Description	Sends the user a request to access the files
Inputs	Accept or decline
Source	An android device.
Outputs	Accept– access granted, decline – access denied
Destination	Cloud Storage, Application
Action	A function that requests the user to grant the application permission to access their files
Requires	File contents
Pre-condition	The user's device should have some files
Post-condition	The access is granted
Side-effects	N/A
Use case	

Voice Assistant

Speech to Text Converter

Function	Speech to text converter.
Description	Converts user speech to text and display the text into the user edit box.
Inputs	Pressing the record button.
Source	Microphone of the android device.
Outputs	Text into the user edit box.
Destination	User’s edit box in the voice assistant layout.
Action	A function that records audio from the microphone android device when the user grants permission and then converts it into text format.
Requires	Permission to access the microphone of the android device.
Pre-condition	The user is within the application via an android device.
Post-condition	The speech is converted to text and displayed in the edit box of the layout.
Side-effects	While the audio is being recorded, the user cannot access any other features of the application.
Use case	

Text to Speech Converter	
Function	Text to speech converter.
Description	Converts bot text to speech and display in the conversation text box in the chat bot layout.
Inputs	Bot response.
Source	Microphone of the android device.
Outputs	Convert bot response in text to speech.
Destination	Voice assistant layout.
Action	A function that converts the response of the bot into speech.
Requires	Bot’s text input
Pre-condition	The user must have sent a query to get a bot response to convert to speech.
Post-condition	The bot’s response is converted to speech.
Side-effects	
Use case	

Chat Bot	
Function	Chat bot.
Description	Takes user input and processes the intent of the user query and respond according to the user query.

Inputs	User query.
Source	User query from the chat bot layout
Outputs	Bot response based on user query.
Destination	Text to speech converter and text box in the assistant layout.
Action	A function that processes user query and responds accordingly.
Requires	User query, wi-fi connection.
Pre-condition	The chat bot must be publicly hosted over the internet.
Post-condition	Creates a response based on user query.
Side-effects	No side effects.
Use case	

Contacts

Record audio	
Function	Record user command.
Description	Records audio when the user grants permission.
Inputs	Pressing the record option.
Source	Microphone of the android device.
Outputs	Audio recorded from the device.
Destination	Application.
Action	A function that records audio from the microphone android device when the user grants permission to access user's contact app.
Requires	Permission to access the microphone of the android device.
Pre-condition	The user is within the application via an android device.
Post-condition	The audio is recorded.
Side-effects	While the audio is being recorded, the user cannot access any other features of the application.
Use case	

Speech to text converter	
Function	Speech to text converter.
Description	Converts the speech from the audio recorded to text.
Inputs	Audio recorded from the android device.
Source	Microphone of the android device.

Outputs	Audio recorded from the device converted into text.
Destination	Application.
Action	A function that converts the audio recorded from the microphone of the android device into text in order to use this text to redirect to the contact app
Requires	Audio recorded from the android device.
Pre-condition	The user is within the application via an android device.
Post-condition	The audio is converted to text.
Side-effects	N/A
Use case	

Access contact app	
Function	Access contacts on user’s device
Description	The user’s contact is viewed through the app
Inputs	Grant permission to access the contacts.
Source	user device.
Outputs	The user’s contact list.
Destination	application
Action	The user’s contacts are now accessible to the application.
Requires	Contact app installed on user device
Pre-condition	The user should have their contacts on their device
Post-condition	The user’s contacts can now be viewed and used.
Side-effects	N/A
Use case	

File system

Create folder	
Function	Create folder.
Description	Allows user to create folders.
Inputs	Pressing the create folder option.
Source	An android device.
Outputs	A folder.
Destination	Cloud Storage, Application.

Action	A function that allows the user to create a folder which he/she can store Files into and name the folder in such a way that could possibly aid the user In locating it when the user is trying to look for it.
Requires	Cloud storage, folder must not already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The folder is created with a default name.
Side-effects	Once the folder is created, no other folder can be named like it.
Use case	

Delete folder	
Function	Delete folder.
Description	Allows user to delete folders.
Inputs	Long Pressing the folder and then pressing the delete option.
Source	An android device.
Outputs	Folder removed/deleted.
Destination	Cloud Storage, Application.
Action	A function that allows the user to delete a folder which he/she originally stored Files into which the user may not need anymore and could possibly help the user in freeing some space up in his/her cloud storage.
Requires	Folder must already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The folder is deleted.
Side-effects	Once the folder and its contents are deleted, it frees up storage space in the cloud.
Use case	

Rename folder	
Function	Rename Folder.
Description	Allows user to Edit folder Name.
Inputs	Long Pressing the folder and then pressing the rename option.
Source	An android device.

Outputs	A new folder name assigned by user.
Destination	Cloud Storage, Application.
Action	A function that allows the user to edit a folder name in such a way that it could possibly aid the user In locating it when the user is trying to look for it.
Requires	Folder must already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The folder is renamed.
Side-effects	Once the folder is renamed, no other folder can be named like it.
Use case	

Open folder	
Function	Open folder.
Description	Allows user to open folders.
Inputs	tapping the folder.
Source	An android device.
Outputs	A folder.
Destination	Cloud storage, Application.
Action	A function that allows the user to open a folder that the user may have created beforehand. Once the folder is opened, the application provides the user with a view of the contents of the folder which could be crucial photos and documents.
Requires	Cloud storage, folder must already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The folder is opened.
Side-effects	Once one folder is opened. The user cannot view any other folder.
Use case	

Copy folder	
Function	Copy folder.
Description	Allows user to copy folders.
Inputs	Long pressing the folder and then Pressing the copy option.
Source	An android device.
Outputs	Folder copied to clipboard.

Destination	Cloud Storage, Application.
Action	A function that allows the user to copy a folder and its contents to another location/directory in the cloud storage that may aid him/her in locating it with greater ease.
Requires	Cloud storage, folder must already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The folder is copied to clipboard waiting to be pasted in another directory.
Side-effects	N/A.
Use case	

Cut folder	
Function	Cut folder.
Description	Allows user to cut folders.
Inputs	Long pressing the folder and then Pressing the cut option.
Source	An android device.
Outputs	Folder copied to clipboard.
Destination	Cloud Storage, Application.
Action	A function that allows the user to move a folder and its contents to another location/directory in the cloud storage that may aid him/her in locating it with greater ease.
Requires	Cloud storage, folder must already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The folder is copied to clipboard waiting to be pasted in another directory.
Side-effects	The folder is no longer accessible in its original directory.
Use case	

Paste folder	
Function	Paste folder.
Description	Allows user to paste folders.
Inputs	Pressing the paste option.
Source	An android device.

Outputs	A folder.
Destination	Cloud Storage, Application.
Action	A function that allows the user to paste a folder and its contents to a location/directory that a user desire.
Requires	Cloud storage, folder must not already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The folder is pasted.
Side-effects	Once the folder is created, no other folder can be named like it.
Use case	

Rename file	
Function	Rename File.
Description	Allows user to Edit file Name.
Inputs	Long Pressing the file and then pressing the rename option.
Source	An android device.
Outputs	A new file name assigned by user.
Destination	Cloud Storage, Application.
Action	A function that allows the user to edit a file name in such a way that it could possibly aid the user In locating it when the user is trying to look for it.
Requires	File must already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The file is renamed.
Side-effects	Once the file is renamed, no other file can be named like it.
Use case	

Upload file	
Function	Upload file.
Description	Allows user to upload files.
Inputs	Selecting a file from the local directory of the android device.
Source	An android device.
Outputs	File uploaded.
Destination	Cloud Storage, Application.
Action	A function that allows the user to upload a file from the local directory of the

	user’s device onto the cloud storage of the application.
Requires	File must already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The file is uploaded
Side-effects	Once the file is uploaded, it takes up storage space in the cloud.
Use case	

Delete file	
Function	Delete file.
Description	Allows user to delete files.
Inputs	Long Pressing the file and then pressing the delete option.
Source	An android device.
Outputs	File removed/deleted.
Destination	Cloud Storage, Application.
Action	A function that allows the user to delete a file that he/she originally uploaded which the user may not need anymore and could possibly help the user in freeing some space up in his/her cloud storage.
Requires	File must already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The file is deleted.
Side-effects	Once the file is deleted, it frees up storage space in the cloud.
Use case	

Search file	
Function	Search file.
Description	Allows user to search for a file.
Inputs	Typing the possible words of the file on the search bar.
Source	An android device.
Outputs	Results relevant to the search returned (i.e., files which contain the words typed on the search bar).
Destination	Cloud Storage, Application.
Action	A function that allows the user to search for the file.

Requires	File must already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The search results are returned.
Side-effects	Once the application returns the results, the user can only view the contents of the search results and nothing more.
Use case	

Cluster files	
Function	Cluster files.
Description	Groups files together that are of a similar search result.
Inputs	The search queries.
Source	An android device.
Outputs	The group of files that belong to a certain cluster.
Destination	Cloud Storage, Application.
Action	A function that takes input from the search bar and uses that as a keyword to browse through the files and check for files that contain that specific keyword. Once these files are found, the function then retrieves them as one whole group of Files that fall under the same keyword.
Requires	Input from the search bar.
Pre-condition	The user is within the application via an android device.
Post-condition	The files are clustered.
Side-effects	N/A
Use case	

Display cluster	
Function	Display cluster.
Description	Displays the group of files clustered.
Inputs	The group of files clustered.
Source	An android device.
Outputs	The group of files that belong to a certain cluster.
Destination	Cloud Storage, Application.
Action	A function that takes the cluster of files that contains the group of files and displays them in an appealing fashion in the form of bubbles.

Requires	The search result clustered.
Pre-condition	The user is within the application via an android device.
Post-condition	The files are clustered and displayed to the user.
Side-effects	The user can only view the clustered result at this time and nothing more.
Use case	

Notes

Create Note	
Function	Create Note.
Description	Allows user to take notes.
Inputs	Pressing the create note option and then using the android device's in-built Keyboard to type in whatever is necessary or record using voice.
Source	An android device.
Outputs	A note.
Destination	Cloud Storage, Application.
Action	A function that allows the user to create a note of important things that may have occurred. Once the note is created, the user can either type in what he/she remembers or have it recorded and then let the application summarize it for the user and return the key points that the user needs.
Requires	Cloud storage.
Pre-condition	The user is within the application via an android device.
Post-condition	The note is created.
Side-effects	Once the note is created, it takes up storage space in the cloud.
Use case	

Delete Note	
Function	Delete note.
Description	Allows user to delete notes.
Inputs	Long Pressing the note and then pressing the delete option.
Source	An android device.
Outputs	Note removed/deleted.

Destination	Cloud Storage, Application.
Action	A function that allows the user to delete a note that he/she originally created Which the user may not need anymore and could possibly help the user in freeing some space up in his/her cloud storage.
Requires	Note must already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The note is deleted.
Side-effects	Once the note is deleted, it frees up storage space in the cloud.
Use case	

Edit Note	
Function	Edit Note.
Description	Allows user to Edit contents of the note.
Inputs	In-built keyboard of the android device.
Source	An android device.
Outputs	An edited note.
Destination	Cloud Storage, Application.
Action	A function that allows the user to edit a note and its contents including the title of the note using the In-built keyboard of the android device.
Requires	Note must already exist.
Pre-condition	The user is within the application via an android device.
Post-condition	The note is edited.
Side-effects	Once the note is edited, it may take more space in the cloud or free up space depending on how the user has edited it.
Use case	

Record audio	
Function	Record audio.
Description	Records audio when the user grants permission.
Inputs	Pressing the record option.
Source	Microphone of the android device.
Outputs	Audio recorded from the device.
Destination	Cloud Storage, Application.

Action	A function that records audio from the microphone android device when the user grants permission in order to aid the user in taking notes by simply recording audio and then presenting the key points of the audio clip back to the user.
Requires	Permission to access the microphone of the android device.
Pre-condition	The user is within the application via an android device.
Post-condition	The audio is recorded.
Side-effects	While the audio is being recorded, the user cannot access any other features of the application.
Use case	

Speech to text converter	
Function	Speech to text converter.
Description	Converts the speech from the audio recorded to text.
Inputs	Audio recorded from the android device.
Source	Microphone of the android device.
Outputs	Audio recorded from the device converted into text.
Destination	Cloud Storage, Application.
Action	A function that converts the audio recorded from the microphone of the android device into text in order to then summarize the note and represent the key points of the audio clip.
Requires	Audio recorded from the android device.
Pre-condition	The user is within the application via an android device.
Post-condition	The audio is converted to text.
Side-effects	N/A
Use case	

Text Summarizer	
Function	Text Summarizer.
Description	Summarizes text of the audio clip.
Inputs	Text of the audio clip.
Source	Microphone of the android device.
Outputs	Summarized text.
Destination	Cloud Storage, Application.

Action	A function that summarizes the text of the audio clip by using the technique of extractive summarization and presents the user the key points of the audio clip.
Requires	The audio converted to text format.
Pre-condition	The user is within the application via an android device.
Post-condition	The text is summarized
Side-effects	N/A
Use case	

Display summarized text	
Function	Display summarized text
Description	Displays the text that has been summarized
Inputs	The summarized text of the audio clip.
Source	Microphone of the android device.
Outputs	Display of the Summarized text.
Destination	Cloud Storage, Application.
Action	A function that displays the summarized text of the audio clip recorded from the android device of the user.
Requires	The summarized text.
Pre-condition	The user is within the application via an android device.
Post-condition	The summarized text is displayed.
Side-effects	N/A
Use case	

Uber

Record audio	
Function	Record user command.
Description	Records audio when the user grants permission.
Inputs	Pressing the record option.
Source	Microphone of the android device.
Outputs	Audio recorded from the device.
Destination	Application.

Action	A function that records audio from the microphone android device when the user grants permission in order to book a ride to a specific destination.
Requires	Permission to access the microphone of the android device.
Pre-condition	The user is within the application via an android device.
Post-condition	The audio is recorded.
Side-effects	While the audio is being recorded, the user cannot access any other features of the application.
Use case	

Speech to text converter	
Function	Speech to text converter.
Description	Converts the speech from the audio recorded to text.
Inputs	Audio recorded from the android device.
Source	Microphone of the android device.
Outputs	Audio recorded from the device converted into text.
Destination	Application.
Action	A function that converts the audio recorded from the microphone of the android device into text in order to use this text on the Uber application's search bar to book rides.
Requires	Audio recorded from the android device.
Pre-condition	The user is within the application via an android device.
Post-condition	The audio is converted to text.
Side-effects	N/A
Use case	

Redirect to Uber application	
Function	Redirected to Uber application
Description	The application redirects to Uber app to book a ride.
Inputs	N/A
Source	An android device.
Outputs	The Uber application
Destination	The Uber application
Action	A function that uses the text from the user's speech command and uses it to search in the Uber app and book a ride there.

Requires	The user command must be converted from speech to text
Pre-condition	The user is within the application via an android device.
Post-condition	Order is placed in Uber app.
Side-effects	None
Use case	

Calendar

Record user command	
Function	Record user command.
Description	Records audio when the user grants permission.
Inputs	Pressing the record option.
Source	Microphone of the android device.
Outputs	Audio recorded from the device.
Destination	Application.
Action	A function that records audio from the microphone android device when the user grants permission to view and access calendar
Requires	Permission to access the microphone of the android device.
Pre-condition	The user is within the application via an android device.
Post-condition	The audio is recorded.
Side-effects	While the audio is being recorded, the user cannot access any other features of the application.
Use case	

Speech to text converter	
Function	Speech to text converter.
Description	Converts the speech from the audio recorded to text.
Inputs	Audio recorded from the android device.
Source	Microphone of the android device.
Outputs	Audio recorded from the device converted into text.
Destination	Application.
Action	A function that converts the audio recorded from the microphone of the android device into text in order to use this text to redirect to calendar application.
Requires	Audio recorded from the android device.

Pre-condition	The user is within the application via an android device.
Post-condition	The audio is converted to text.
Side-effects	N/A
Use case	

Redirect to calendar	
Function	Access calendar on user's device
Description	The user's calendar is viewed through the app
Inputs	Grant permission to access the calendar.
Source	user device.
Outputs	The user's calendar
Destination	application
Action	The user's calendar is now accessible to the application.
Requires	Calendar app installed on user device
Pre-condition	The user should have their calendar on their device
Post-condition	The user's calendar can now be viewed and used.
Side-effects	N/A
Use case	

Emails

Record User Command	
Function	Record user command.
Description	Records audio when the user grants permission.
Inputs	Pressing the record option.
Source	Microphone of the android device.
Outputs	Audio recorded from the device.
Destination	Application.
Action	A function that records audio from the microphone android device when the user grants permission to view GMAIL
Requires	Permission to access the microphone of the android device.

Pre-condition	The user is within the application via an android device.
Post-condition	The audio is recorded.
Side-effects	While the audio is being recorded, the user cannot access any other features of the application.
Use case	

Speech to text converter	
Function	Speech to text converter.
Description	Converts the speech from the audio recorded to text.
Inputs	Audio recorded from the android device.
Source	Microphone of the android device.
Outputs	Audio recorded from the device converted into text.
Destination	Application.
Action	A function that converts the audio recorded from the microphone of the android device into text in order to use this text to redirect to GMAIL
Requires	Audio recorded from the android device.
Pre-condition	The user is within the application via an android device.
Post-condition	The audio is converted to text.
Side-effects	N/A
Use case	

Redirect to Gmail	
Function	Access GMAIL on user’s device
Description	The user’s GMAIL is viewed through the app
Inputs	Grant permission to access the GMAIL.
Source	user device.
Outputs	The user’s GMAIL list.
Destination	application
Action	The user’s GMAIL are now accessible to the application.
Requires	GMAIL app installed on user device
Pre-condition	The user should have their GMAIL on their device
Post-condition	The user’s GMAIL can now be viewed and used.

Side-effects	N/A
Use case	

Non-functional requirements

- Usability:

A.I.V.A.S uses two main techniques to improve the usability of the application – the touch and the voice. The touch allows the user to manually control the application to achieve the intended tasks. Voice allows the user to speak to the application about the task they would like the application to do for them.

Moreover, to improve the usability of the application a small tab with all the voice commands provided in the application is a guideline to help the user understand the flow of interaction between the application and the user. To help complete their tasks manually without the use of voice assistant, consistent icons and labels will be used throughout the application to help the user navigate and achieve their tasks.

- Reusability:

The user can reuse the voice assistant to achieve multiple tasks that are offered by the application. For example, a user can use the speech to text component to fix a meeting, or to set a reminder. Another example would be to consider the statement "Set a reminder and book a meeting at 12:45 PM on Tuesday". This is another form of reusability where the user achieves multiple tasks in one statement where the user is re-using the voice assistant component to set a reminder and to book a meeting in one statement.

- Reliability:

If the voice assistant could not understand the instructions, it would ask the user whether it's understanding understood was correct? If yes, then the assistant would go ahead and perform the task, otherwise it would ask the user to repeat their query. This would not only help reliability but also help the usability drastically as it removes the potential of user to have to repeat their queries multiple times.

The application will contain a feedback system through notifications that would notify the user of any errors that have occurred while trying to achieve a task. For example, if a user books a meeting in their calendar but could not be due to unavailability of internet connection, the user will be informed through the notifications center and the same theory would be applied to any functionality that fails during execution. The application is expected to execute any given task under a minute that does not depend on other user's replies, and at the same time must not exceed 4 minutes to execute a task upon user's reply. If the application exceeds the average task execution time, it will alert the user with a choice whether to wait to respond or to cancel the operation.

- Performance:

The backend system would be on an optimized cloud system with optimized query results called Firebase. Therefore, extracting any information of the given user is expected to take no more than a minute due to the file organization functionality. The GUI would have a minimal aesthetic and a modern look with lots of interactivity and hence, loading the GUI components of the application is expected to load within 30s. The memory taken by the application is expected to not concede 50MB.

- Supportability:

The user will be given constant support by means of F.A.Q, regarding the use of the application. The application will also keep sending regular updates which would mainly improve the voice assistant component.

4. High-level Design

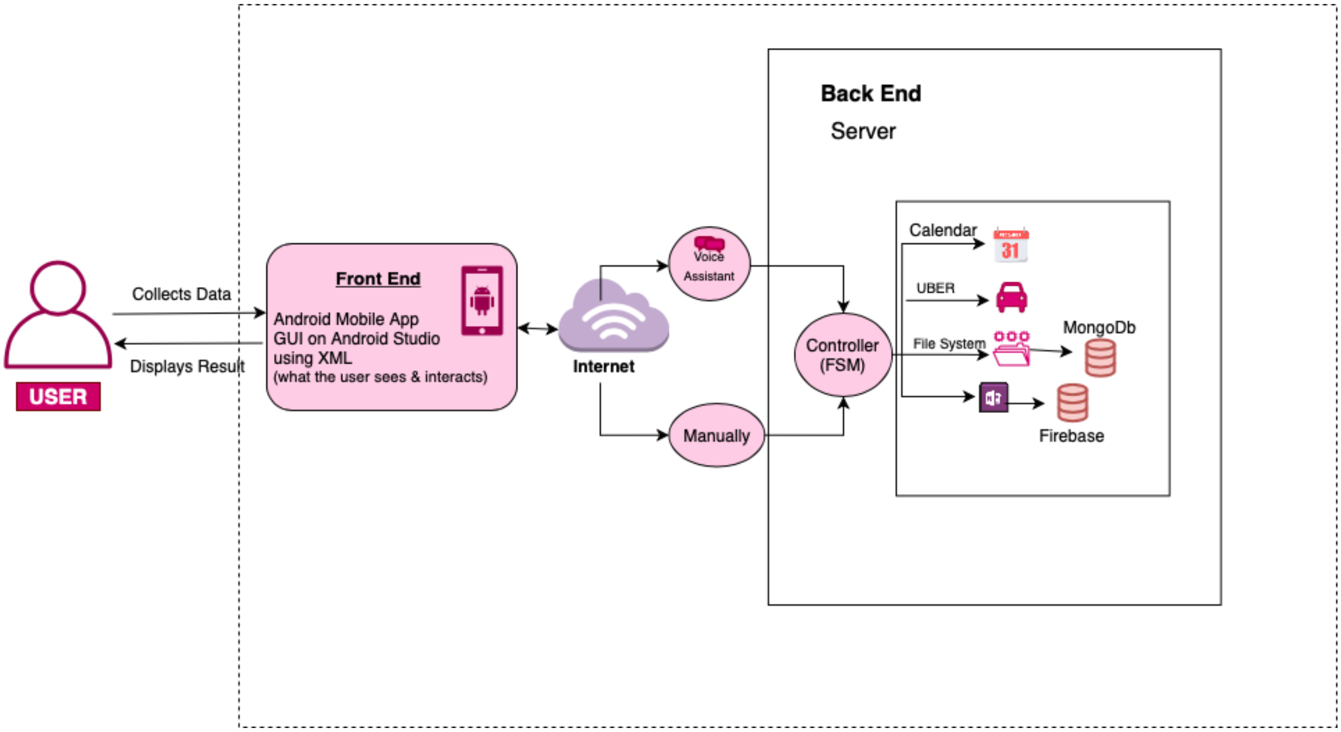
System overview and design considerations

The system architecture is the overview of how the entire system functions together. The user can either give voice commands or manually access the application. The data from the user is collected from the front end which is what the users sees and interacts with. The front end is going to be designed using XML on android studio. For the application to function completely it should have secured internet connection at all time. The backend for our application would be running with a controller. The controller is dedicated to listening to user requests. We need a server to handle the users voice commands which will pass through a finite state machine (FSM) which will be in a while loop, transitioning from one state to another in response to the inputs. A request is translated to command, and the command leads to FSM. Depending on the input event that is presented, The FSM will perform a predetermined sequence of actions. The file system and note taking feature talks directly with the database.

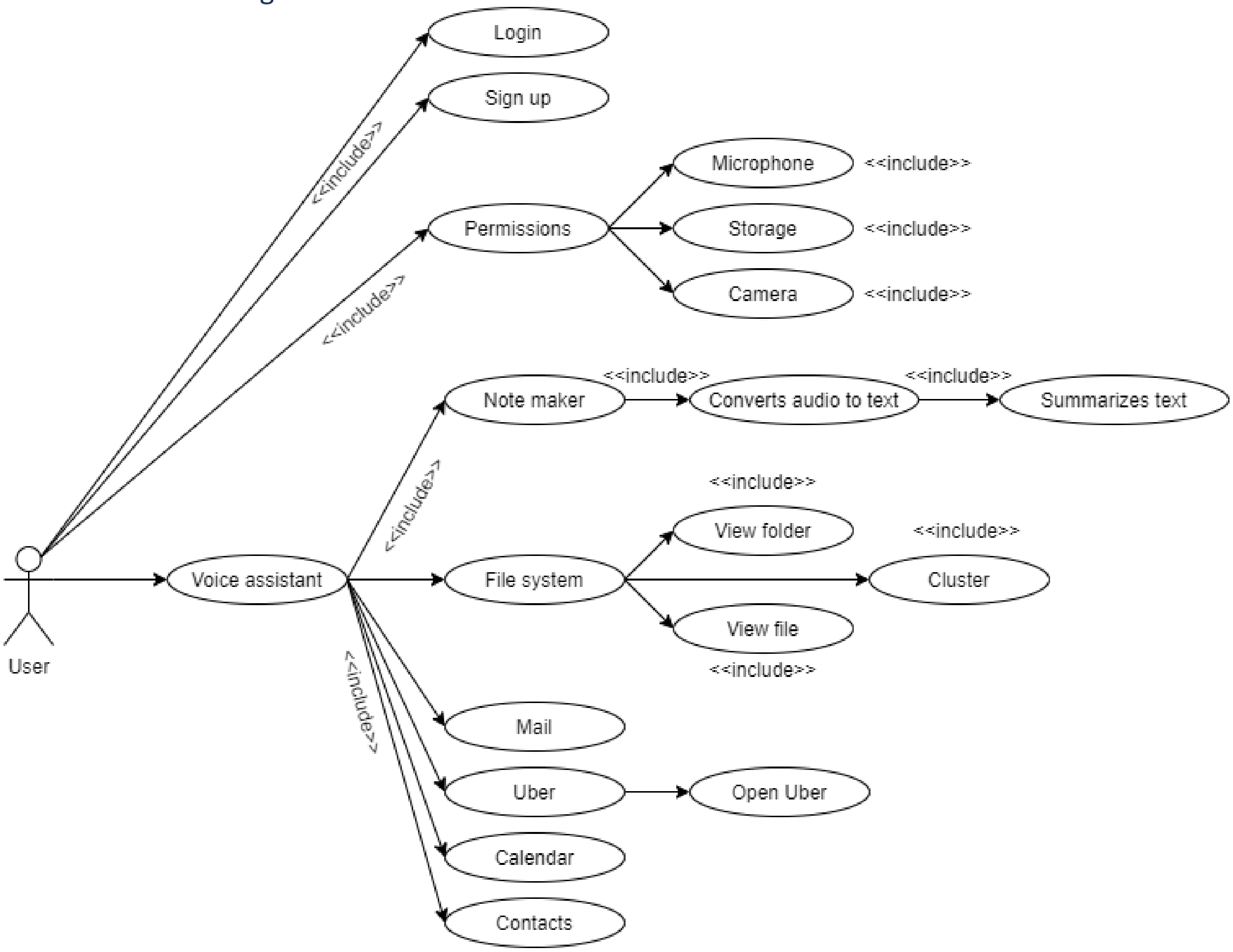
Working can be stressful especially the long hours in front of a screen doing various tasks and responsibilities. Our app has been designed with colors, animations, themes, music and navigations which eases, calms the mind of the user which in turn increases productivity and popularity of the app among the working sectors. The app is designed to easily work with other things. The contents of the application such as the icons, images, texts etc. are aligned well following the typography rules. The orientation of the elements is precise and easy to follow. The design is customized based on the device used but the results are always the same. The app can be easily modified and maintained. The performance rate and quality of the app is good with respect to speed and accuracy. This app is built to be accessible to people with visual impairments as we have integrated voice to text and speech to text where the user can hear or dictate commands. The functions and features are automated which creates a better user experience. The color and themes used in the design are all soothing, aesthetic and pleasing to the eyes of the user. The color theme used here is: (Blue:White:Black:Grey). The material and design were built following the design goals, usability goals and principles. All resources were already available therefore there was no cost. The designs are easy to take apart, fix and reuse. It is reusable, extensible and durable. The app makes efficient use of energy and resources, creates easy navigation being efficient to any user. All the features are under one platform, all essential features for an office environment have been added in the application. All design standard was followed while designing and developing the application increasing its style and usability.

System architecture and use cases

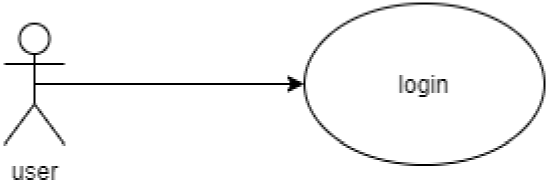
System architecture



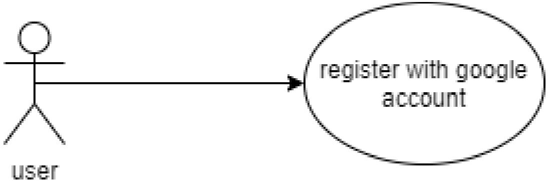
Overall Use case diagram:



- Login

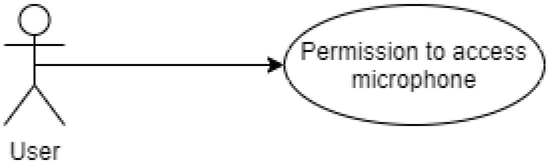


- Sign Up

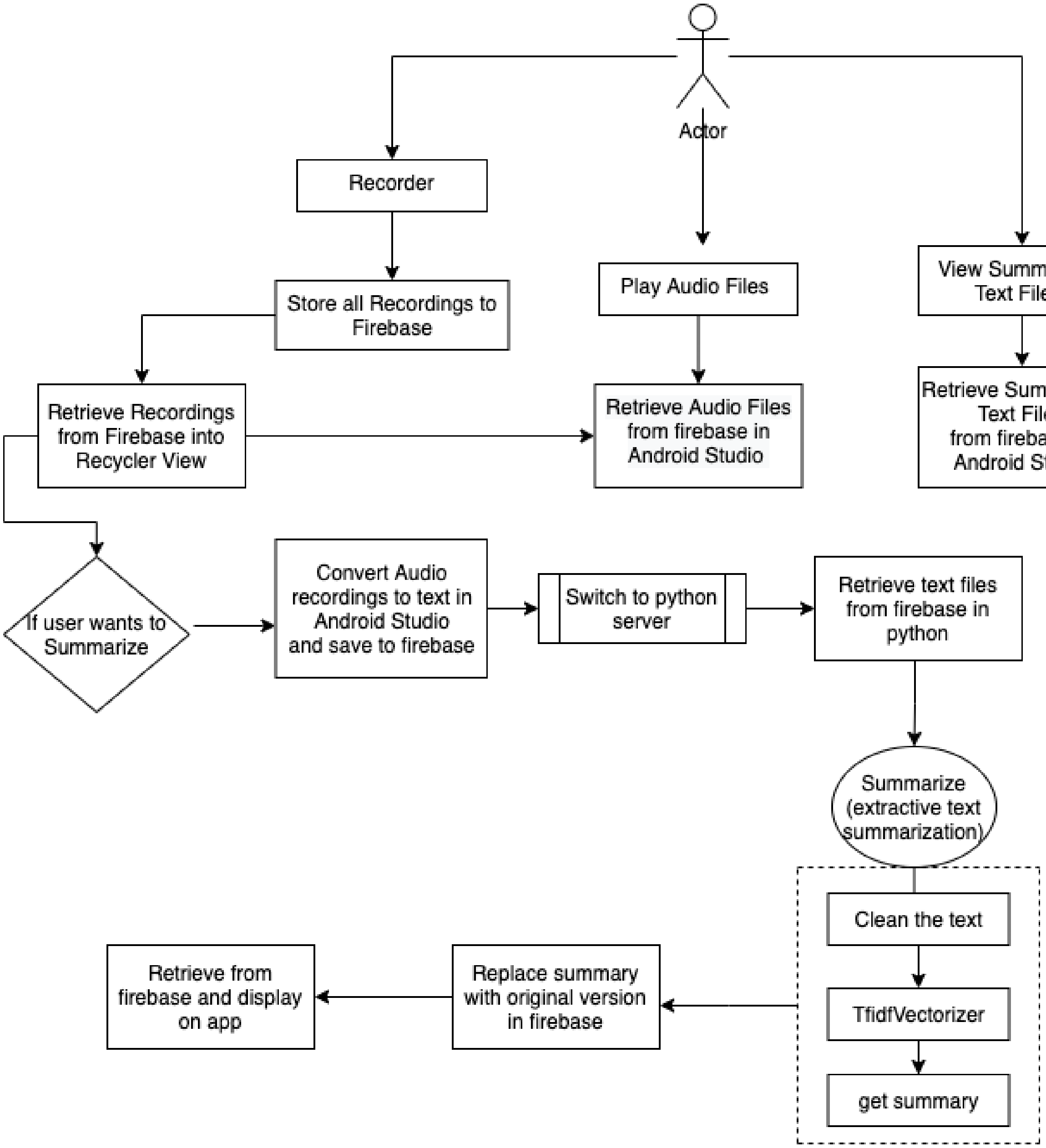


- Permissions

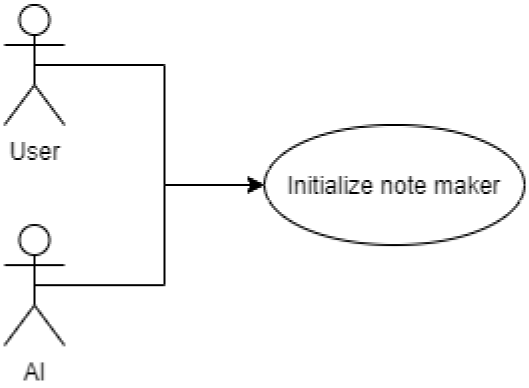
Access Microphone



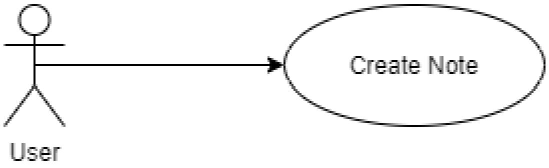
- Note Maker


Actor
Recorder
Store all Recordings to Firebase
Retrieve Recordings from Firebase into Recycler View
Play Audio Files
Retrieve Audio Files from firebase in Android Studio
View Summarized Text Files
Retrieve Summarized Text Files from firebase in Android Studio
If user wants to Summarize
Convert Audio recordings to text in Android Studio and save to firebase
Switch to python server
Retrieve text files from firebase in python
Summarize (extractive text summarization)
Clean the text
TfidfVectorizer
get summary
Replace summary with original version in firebase
Retrieve from firebase and display on app

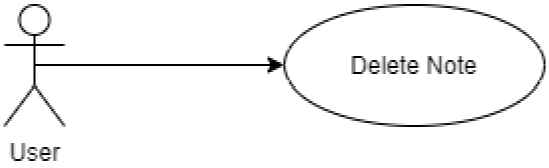
Initialize note feature



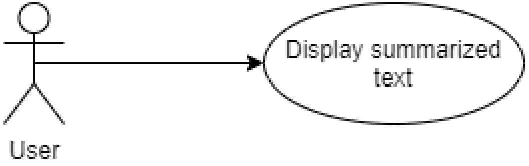
Create note



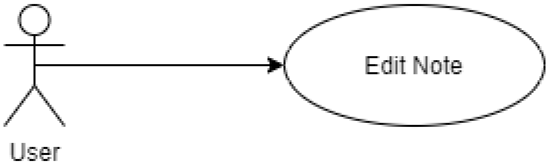
Delete note



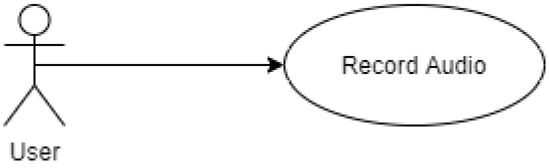
Display summarized text



Edit note

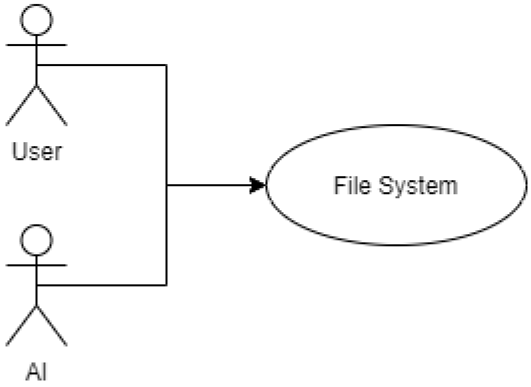


Record audio

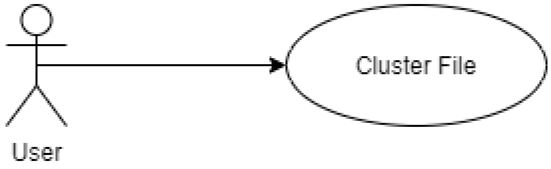


- File System

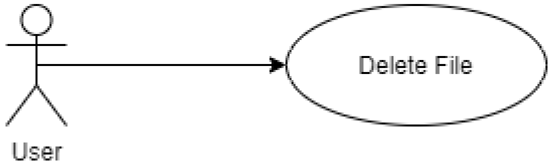
Initialize File System



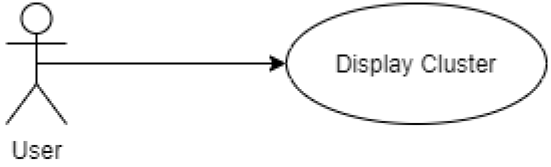
Cluster file



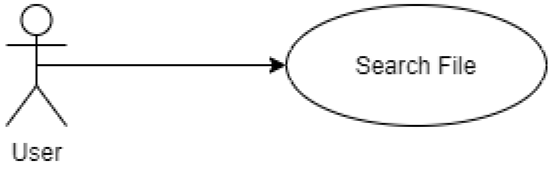
Delete file



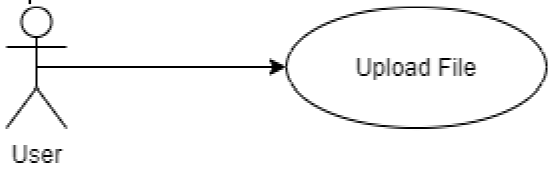
Display cluster



Search file

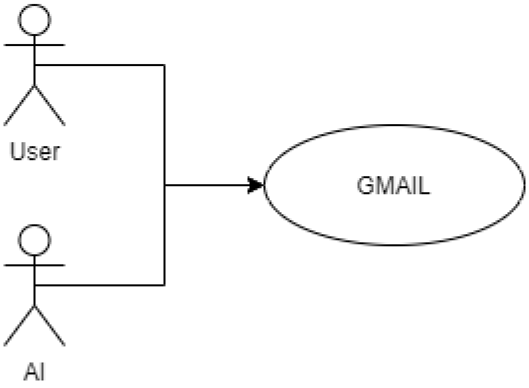


Upload file



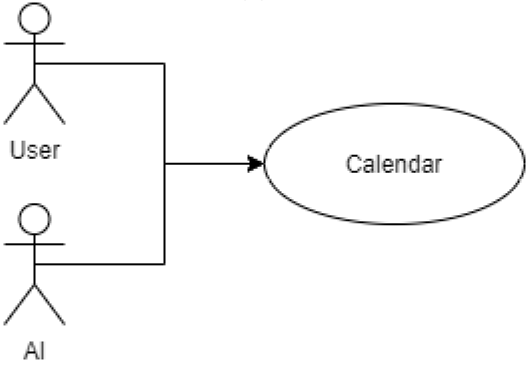
- Mail

Redirect to GMAIL



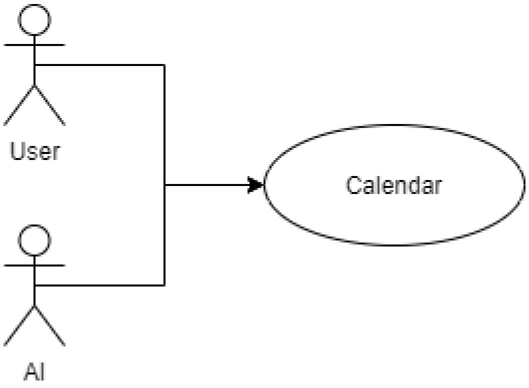
- Uber

Redirect to Uber application



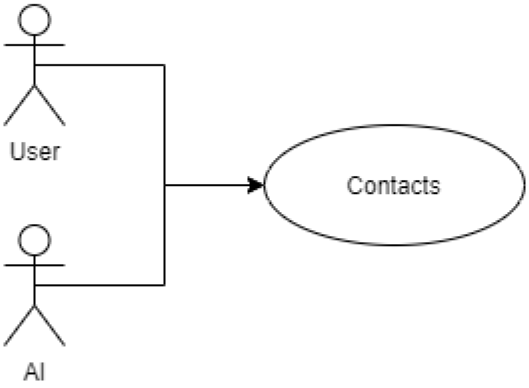
- Calendar

Redirect to calendar

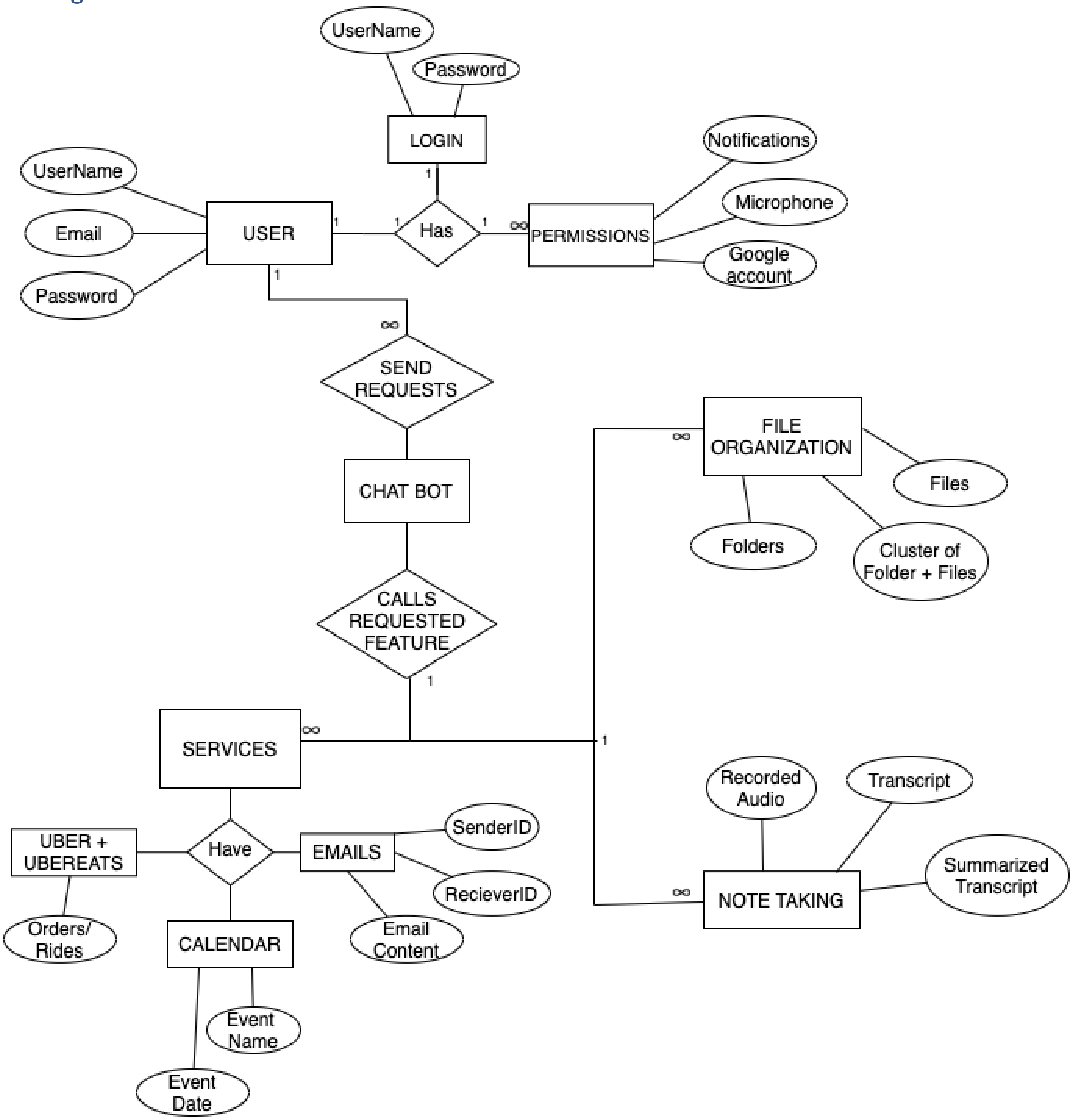


- Contacts

Redirect to contacts

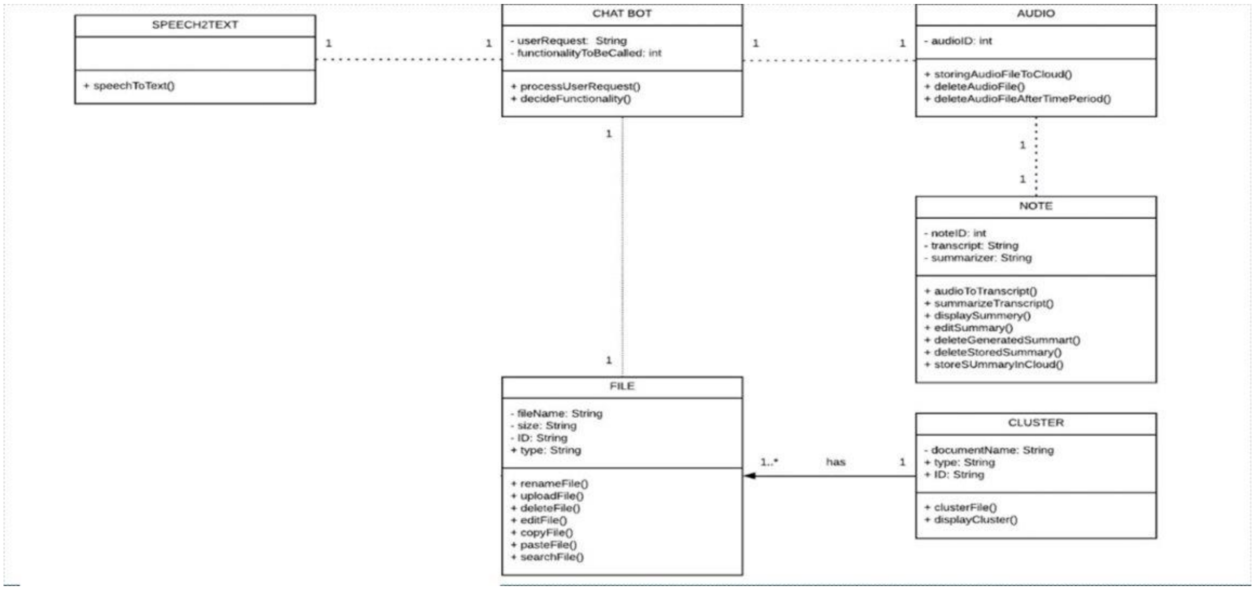

User
AI
Contacts

ER diagram

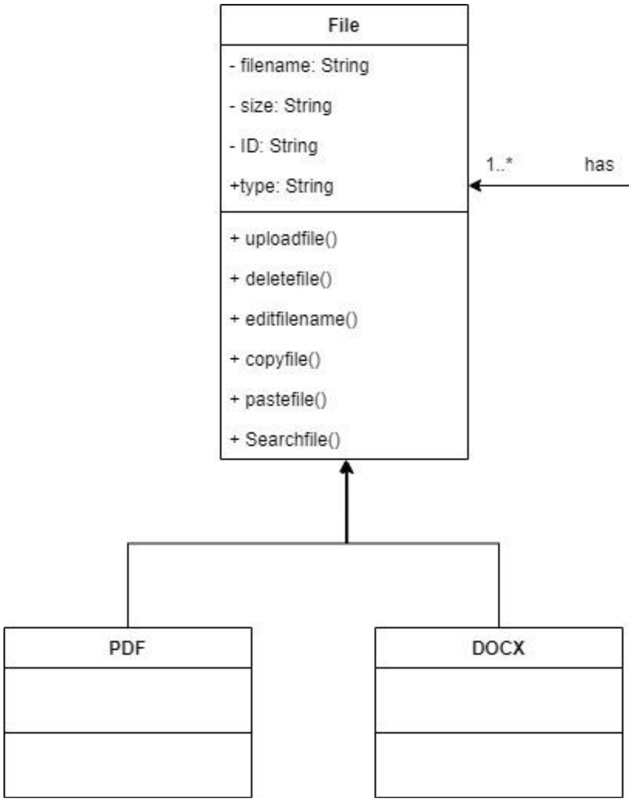


Class diagram

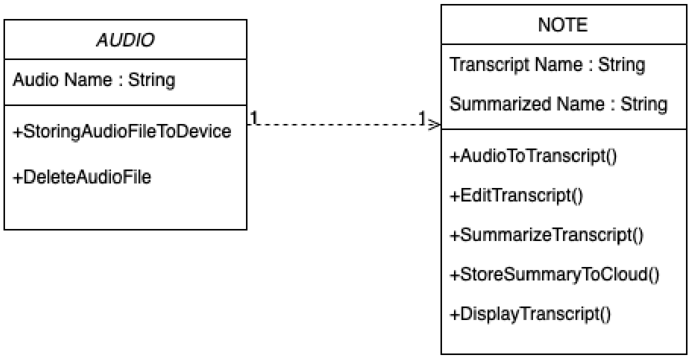
Overall Class Diagram



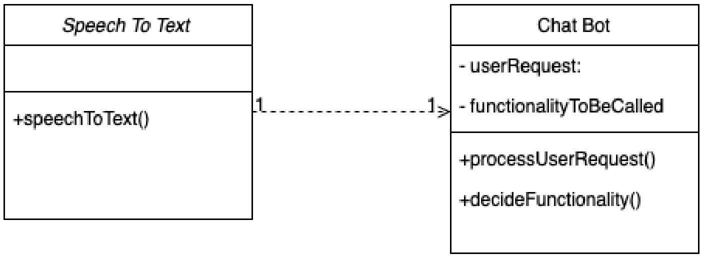
- File System



- Note Maker



- Chatbot

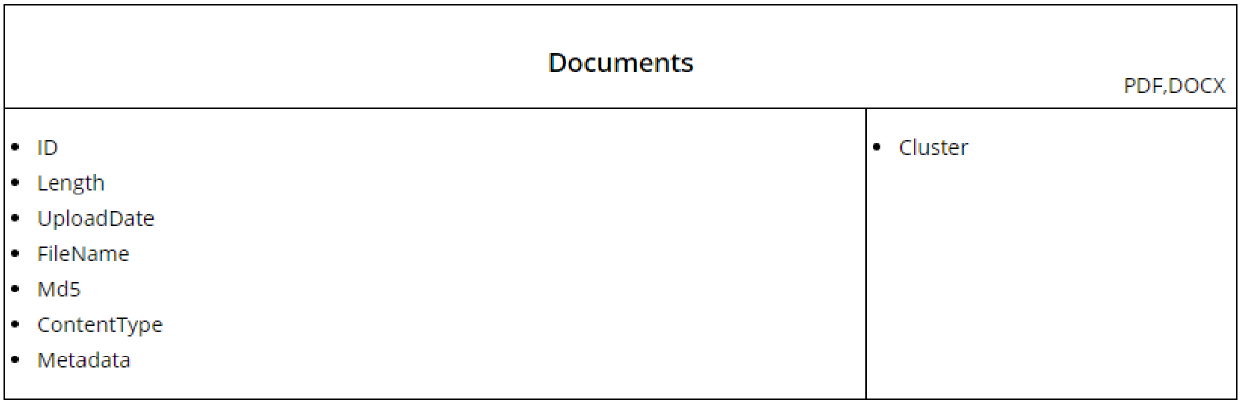


CRC cards

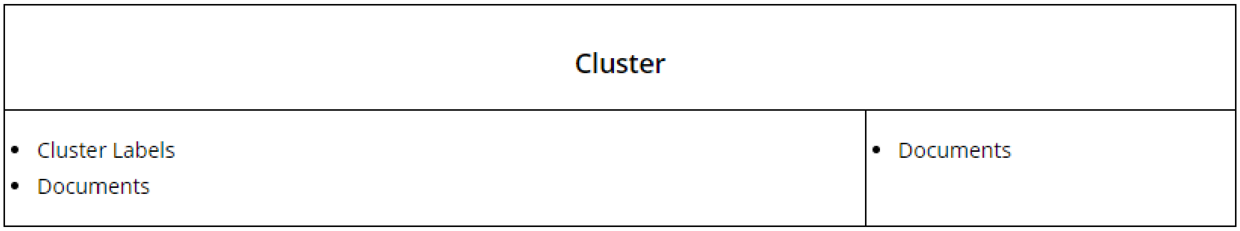
- Voice assistant

Voice Assistant	
Responsibilities	Collaborators
Get user query as text or speech. Understand and predict the intent for user query. Respond according to user intent through text and speech.	Initialize: • File system • Note maker • Contacts • Uber • Calendar • Email Report: • Date and time • Weather

- File System



Documents	PDF,DOCX
• ID • Length • UploadDate • FileName • Md5 • ContentType • Metadata	• Cluster

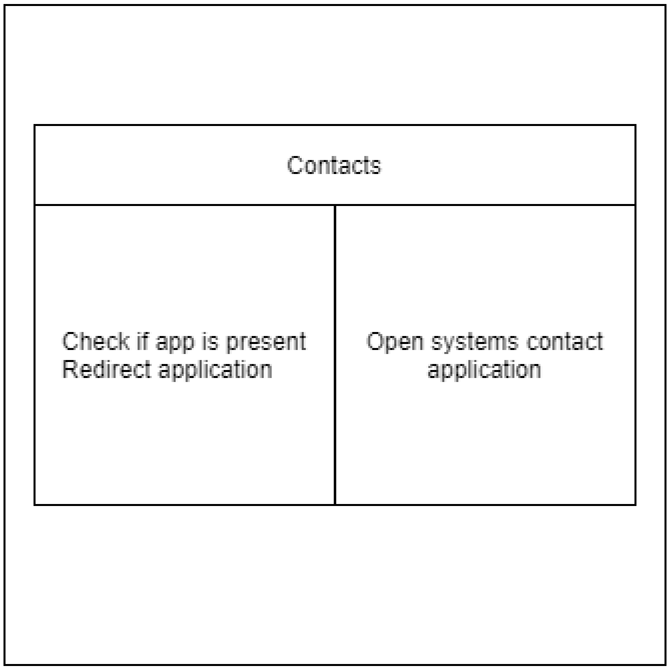


Cluster	
• Cluster Labels • Documents	• Documents

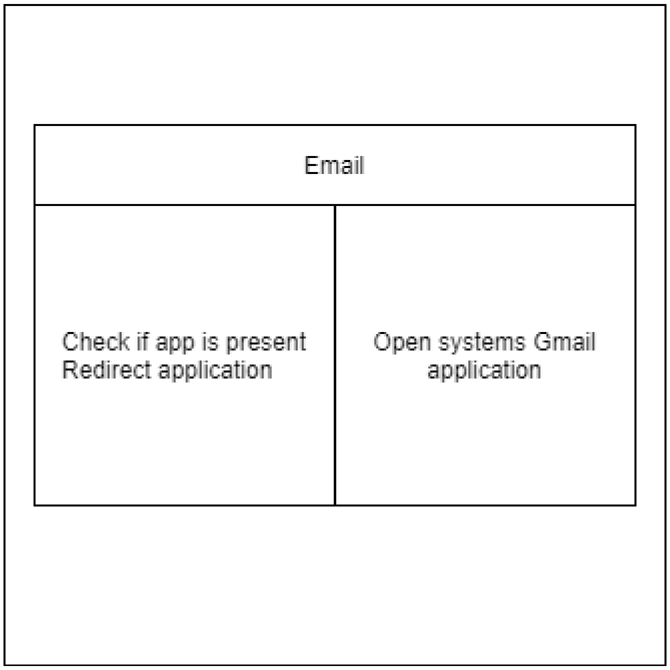
- Note maker

Note Maker	
Responsibilities	Collaborators
• Records audio • Convert audio to Text in real-time • Store in cloud storage • Display Summary	• Summarizer in Python

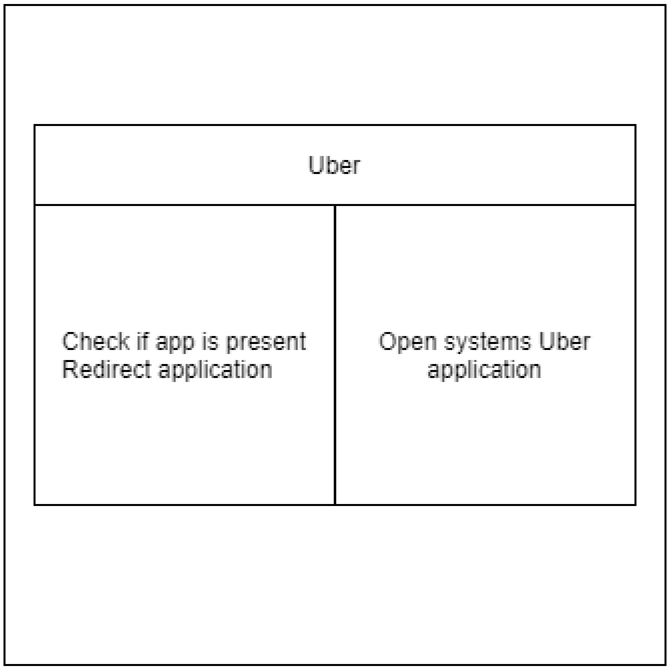
- Contacts


Contacts
Check if app is present
Redirect application
Open systems contact application

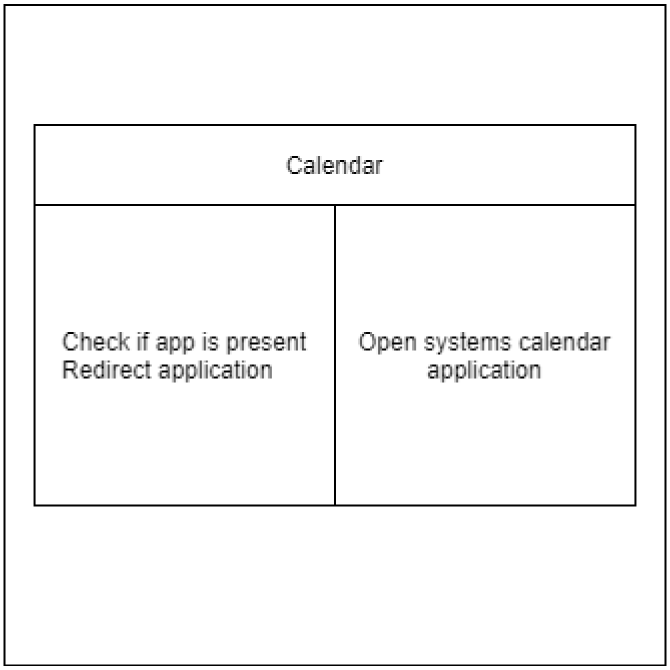
- Email


Email
Check if app is present
Redirect application
Open systems Gmail application

- Uber



- Calendar



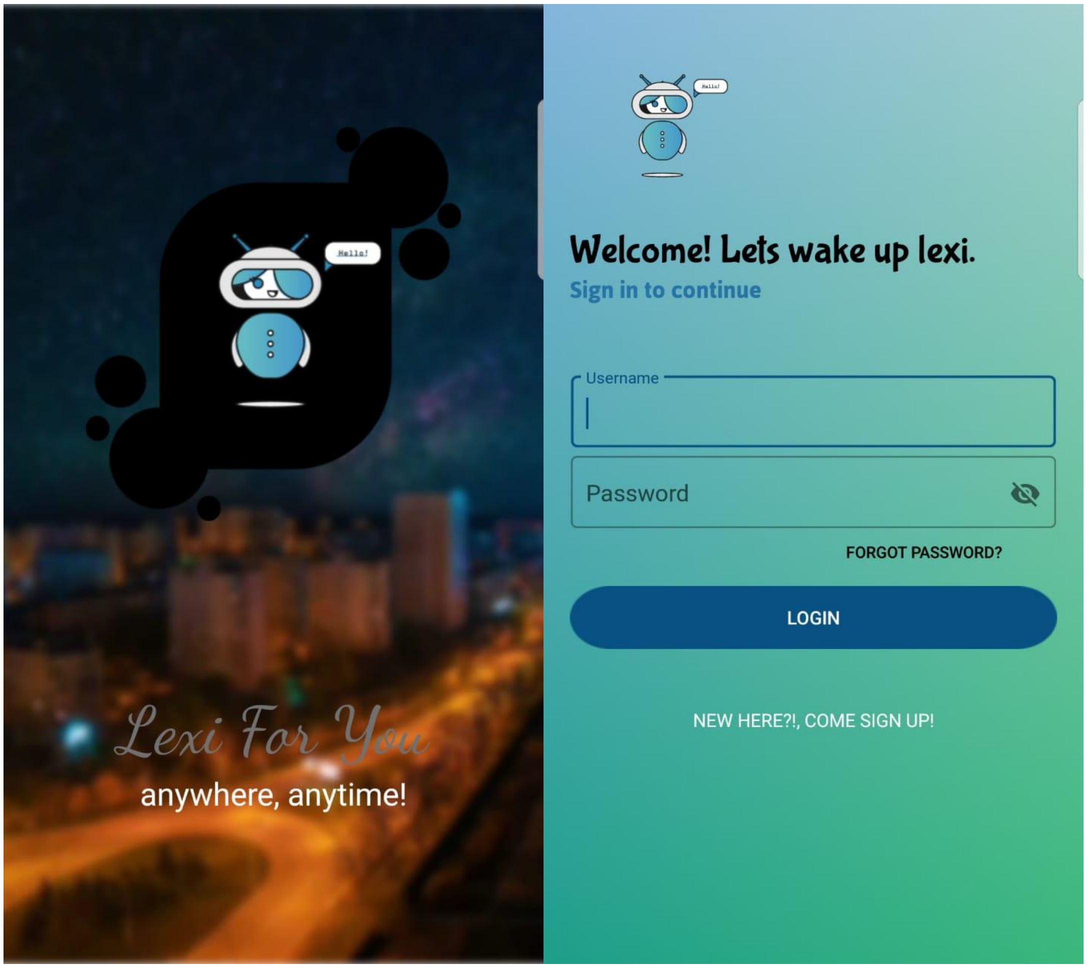
User interfaces

Virtual Assistance is a mobile application that the users can download from play store into their android devices and use it as a daily virtual office assistant. The user logs in or registers with their google account. Pop up permissions asking to access the user’s camera, microphone and storage will be visible to the user to which the user can click “ok” or “cancel”. He/she is then redirected to the home page with all the menu features displayed. The main features of the application include file organization, meeting note maker and the virtual assistant. It also contains services such as sending emails using voice command, ordering food via Uber Eats, fixing rides via Uber, synchronizing calendar events and importing contacts. When the user opens the file system, the user can search and view their files and folders using a cluster visualization. The user can also create, edit and delete their files and folders. When the user opens the meeting note maker function the user will be asked to record an audio or import an audio file. If the user opts to record the audio, after the audio is recorded its converted to text and the text is summarized and displayed to the user. if the user opts to import a recording, after the user imports the file, the audio will be converted to text and the text will be summarized and saved in the user's cloud database. The user will then have the option to delete the audio or save it for 3 months and then it will automatically have cleared. The user can control and navigate through the application using the voice assistant chat bot.

The user interfaces aim to allow the general users of secretary to use the application in two ways which is the voice and the manual touch input. The application aims to provide an intuitive and beautiful looking application by using the research that has been done by the material design on the following:

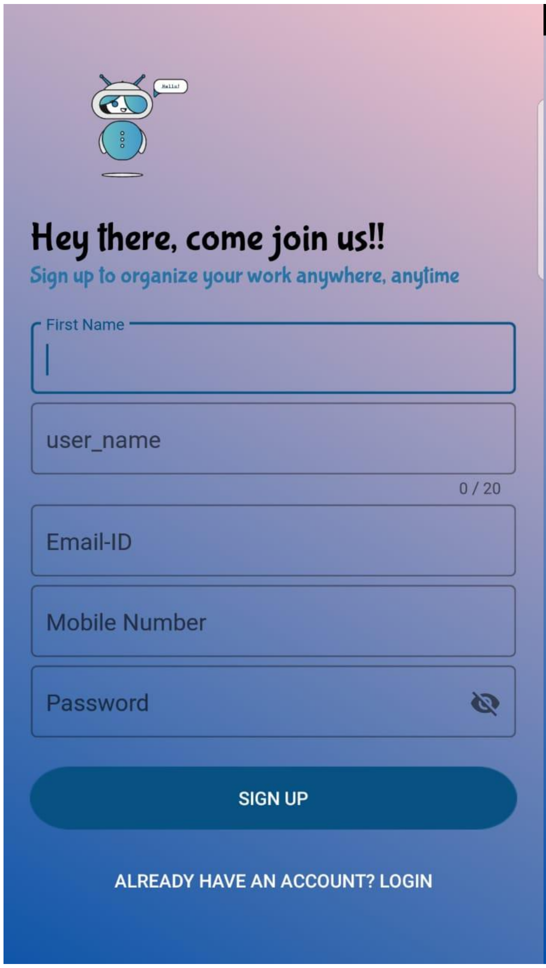
- Material design guidelines: Focuses on the color pallet and theme of the application such as the color of the text and the color of the page. The application aims to use the material dark theme for its UI.
- Components: They define the interactive building blocks for creating a user interface such as application bars, banners, buttons, alert boxes etc.
- Icons: This research tells the developer about using consistent icons across different applications and at the same time tweaking in a way that it goes with the theme of the application. The application aims to use the system icons for its UI.

Screens Images:



Welcome! Lets wake up lexi.

Sign in to continue



Hey there, come join us!!

Sign up to organize your work anywhere, anytime

First Name

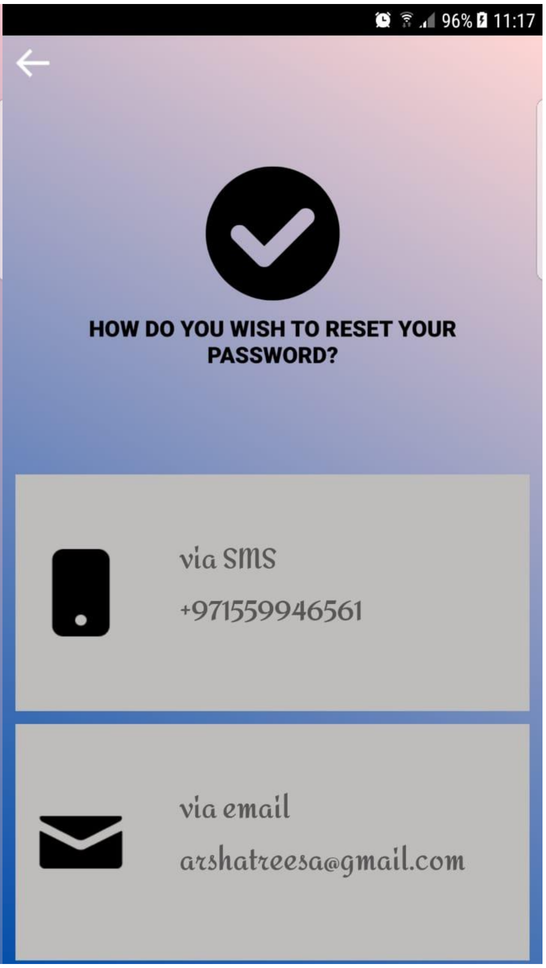
user_name

0 / 20

Email-ID

Mobile Number

Password



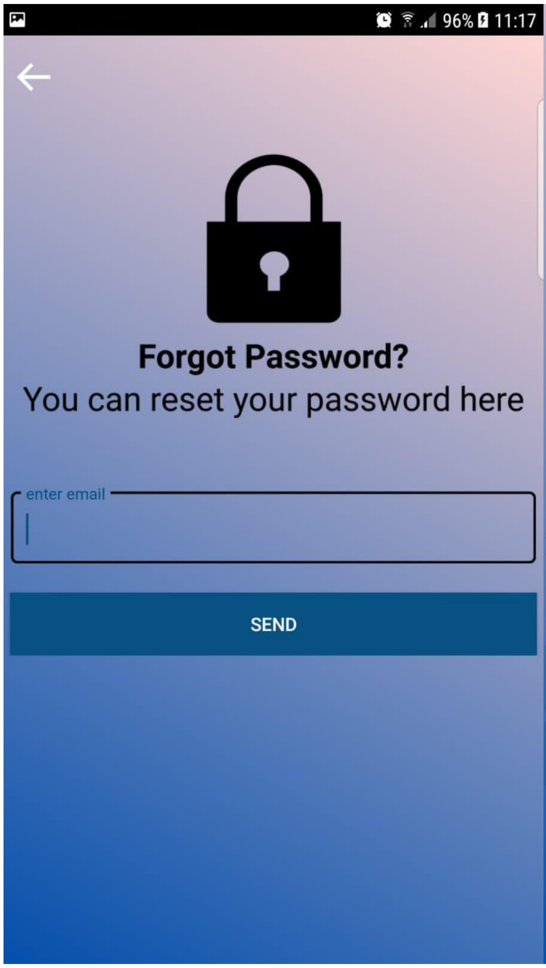
HOW DO YOU WISH TO RESET YOUR PASSWORD?

via SMS

+971559946561

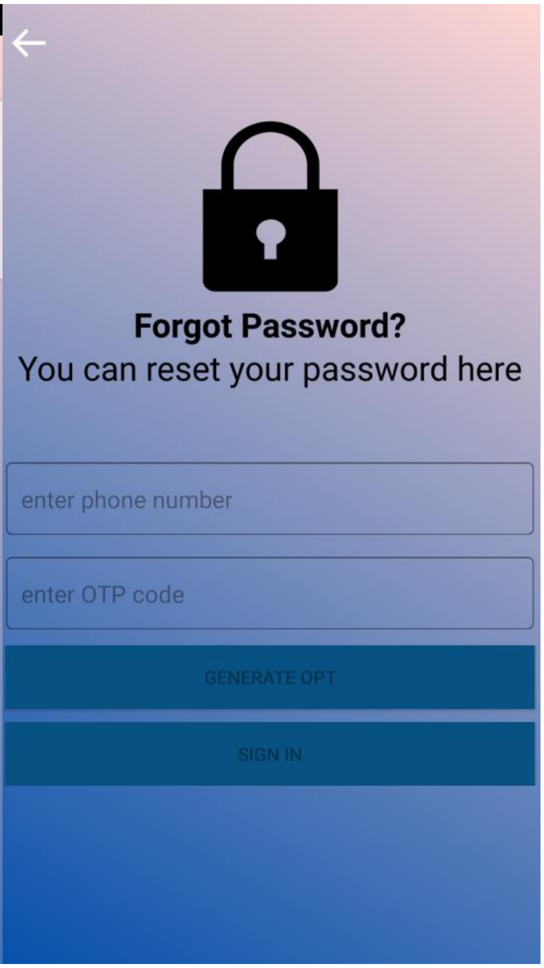
via email

arshatreesa@gmail.com



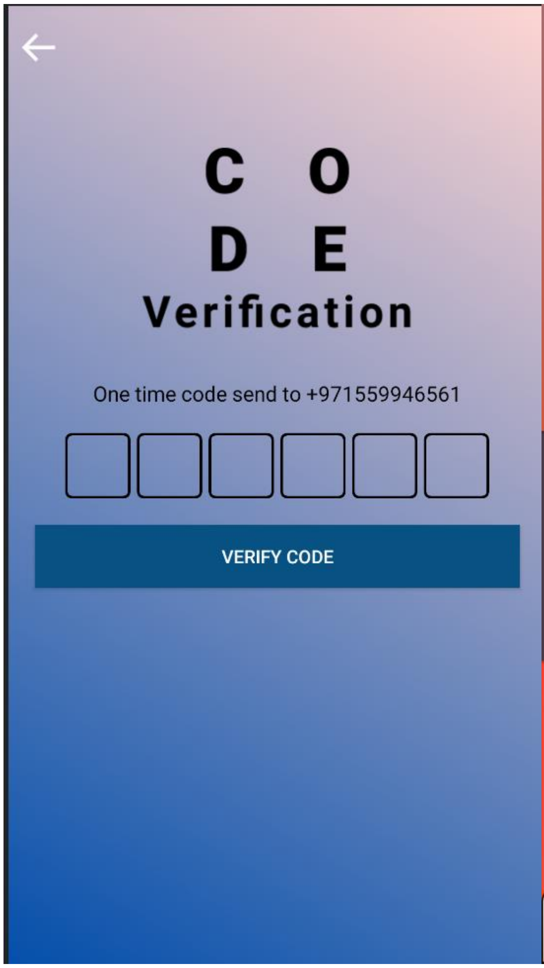
Forgot Password?

You can reset your password here



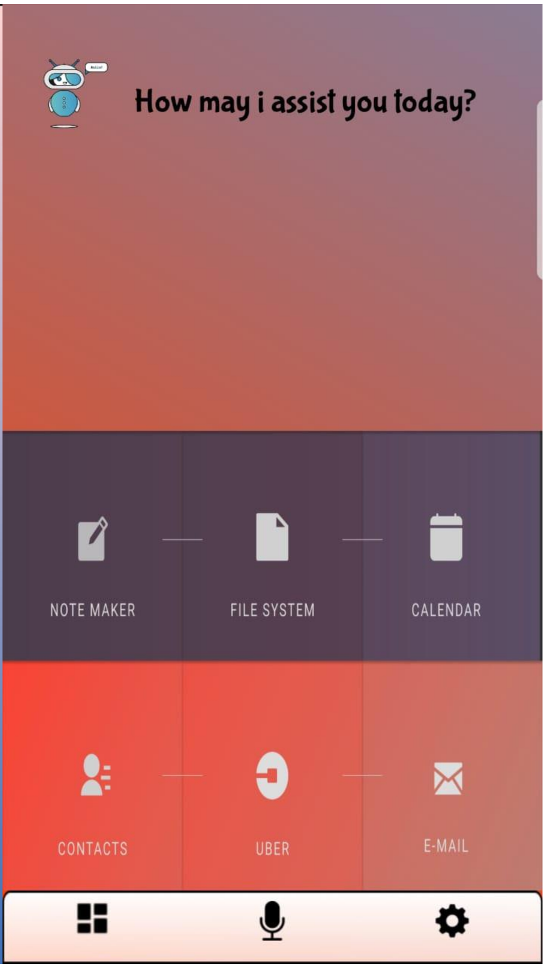
Forgot Password?

You can reset your password here

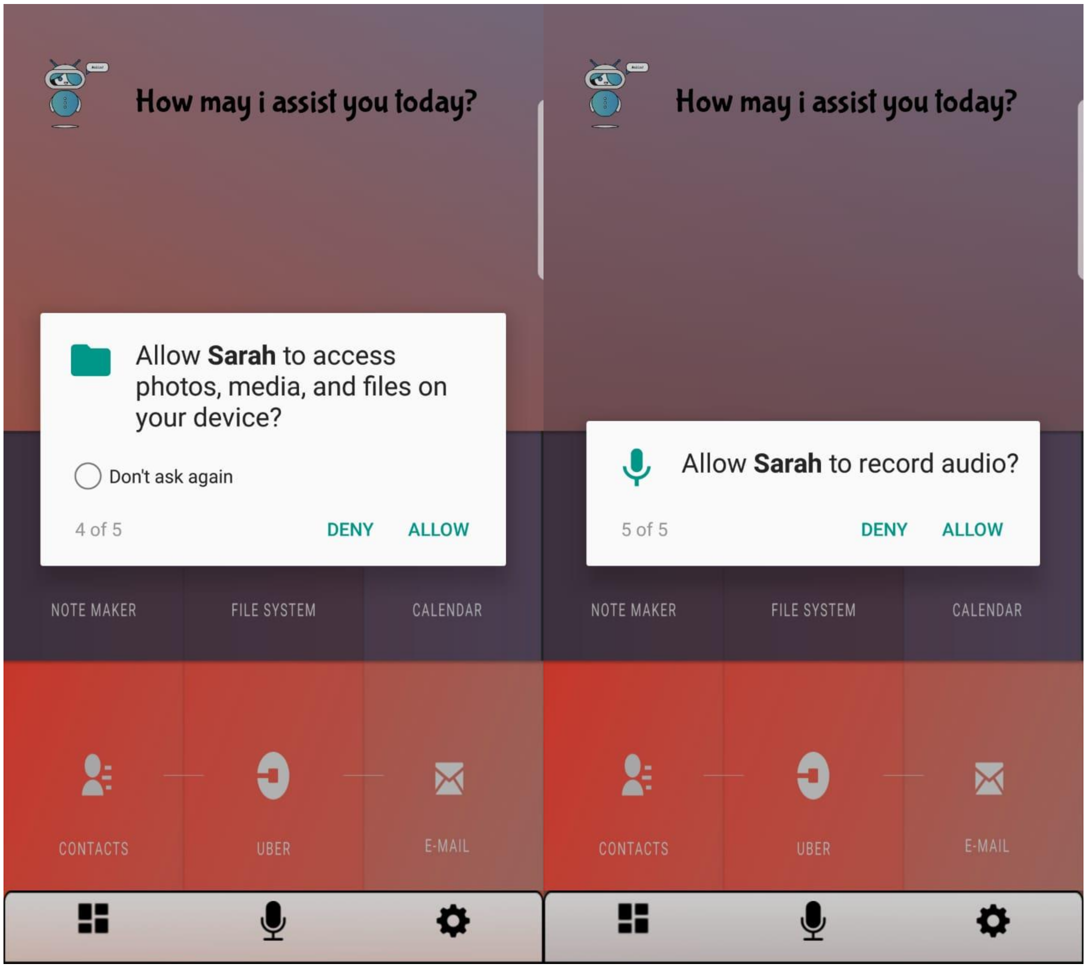


CODE Verification

One time code send to +971559946561



How may i assist you today?



How may i assist you today?

Allow **Sarah** to access photos, media, and files on your device?

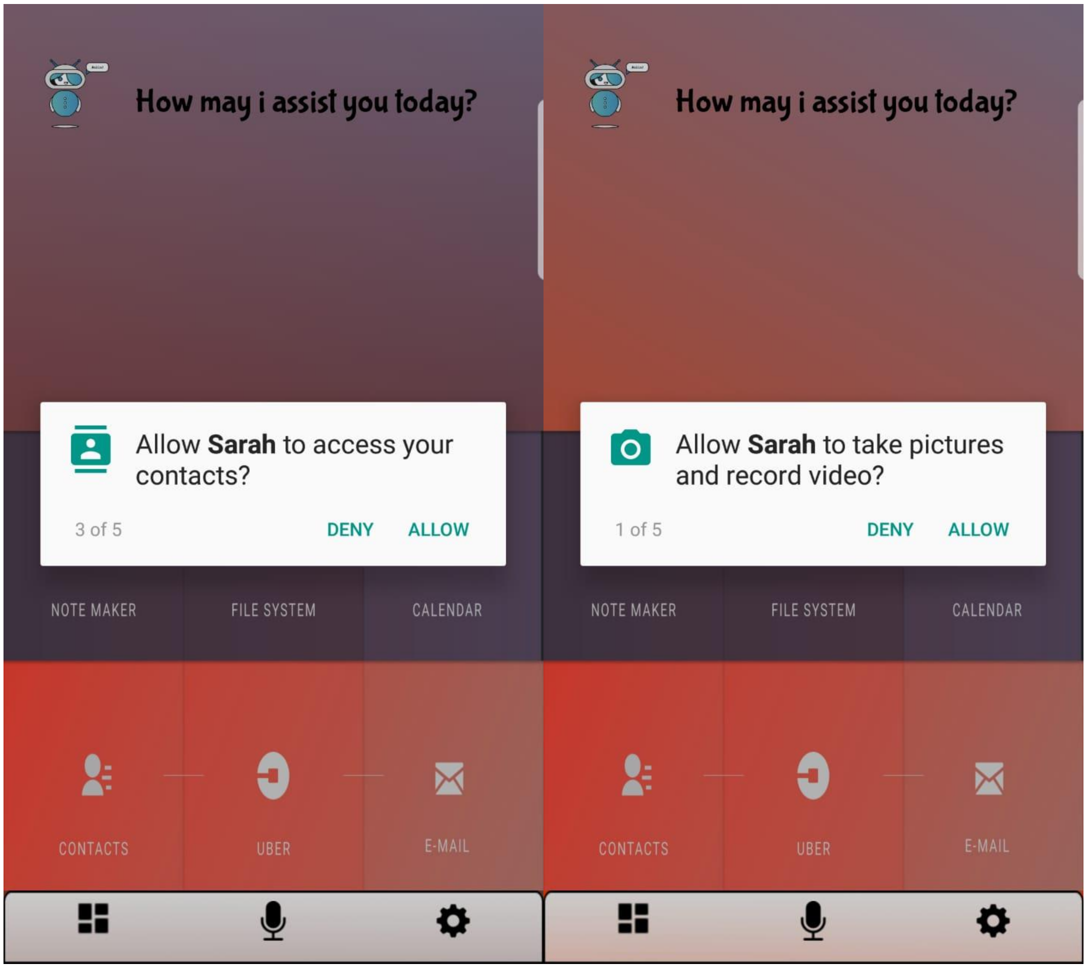
○ Don't ask again

4 of 5

How may i assist you today?

Allow **Sarah** to record audio?

5 of 5



How may i assist you today?

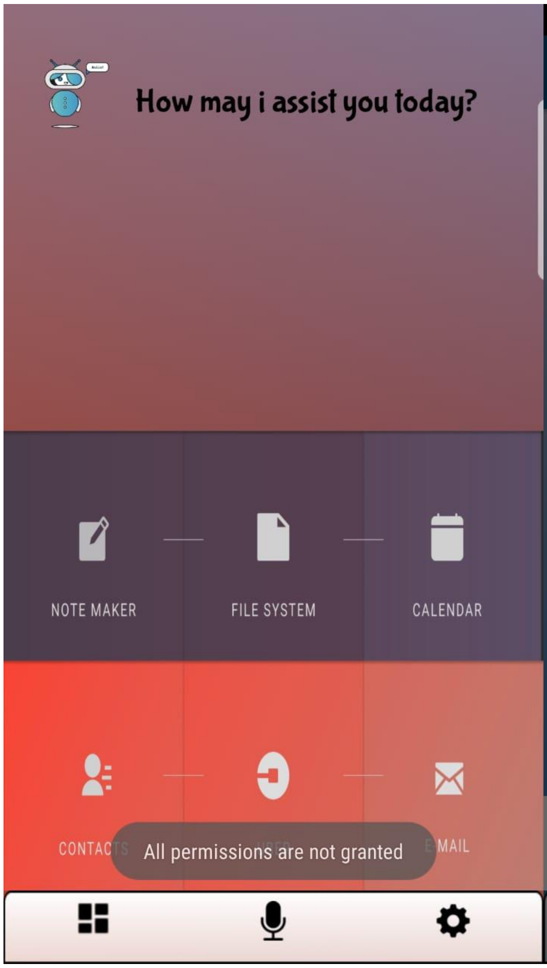
Allow **Sarah** to access your contacts?

3 of 5

How may i assist you today?

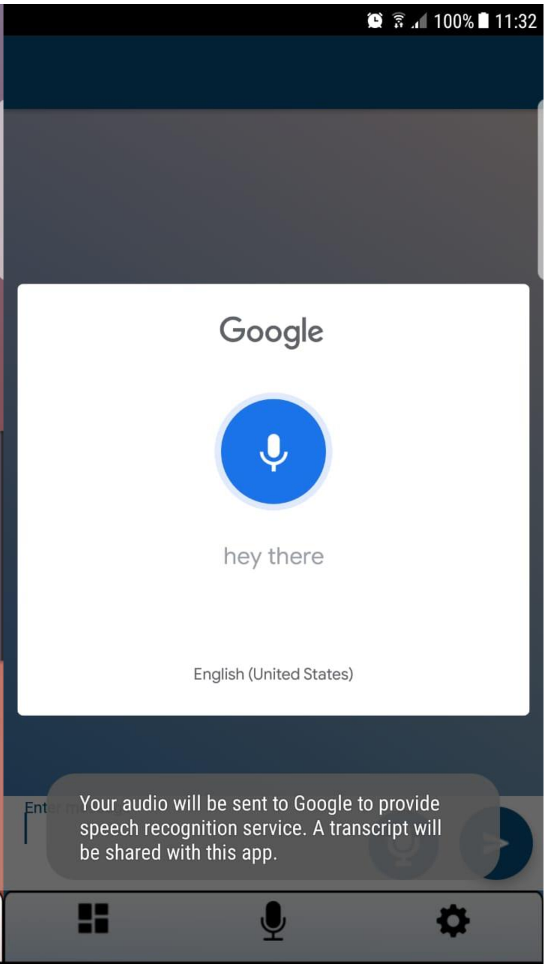
Allow **Sarah** to take pictures and record video?

1 of 5



How may i assist you today?

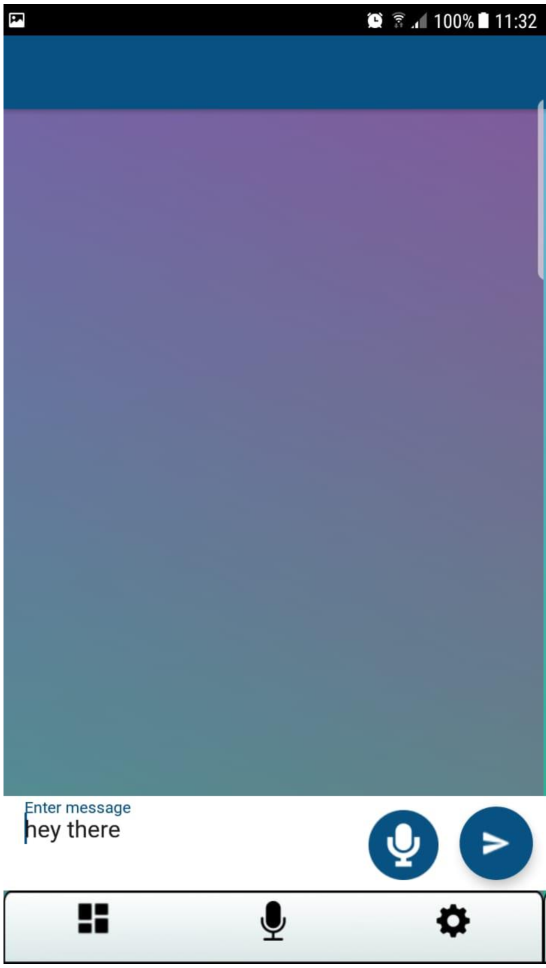
All permissions are not granted



hey there

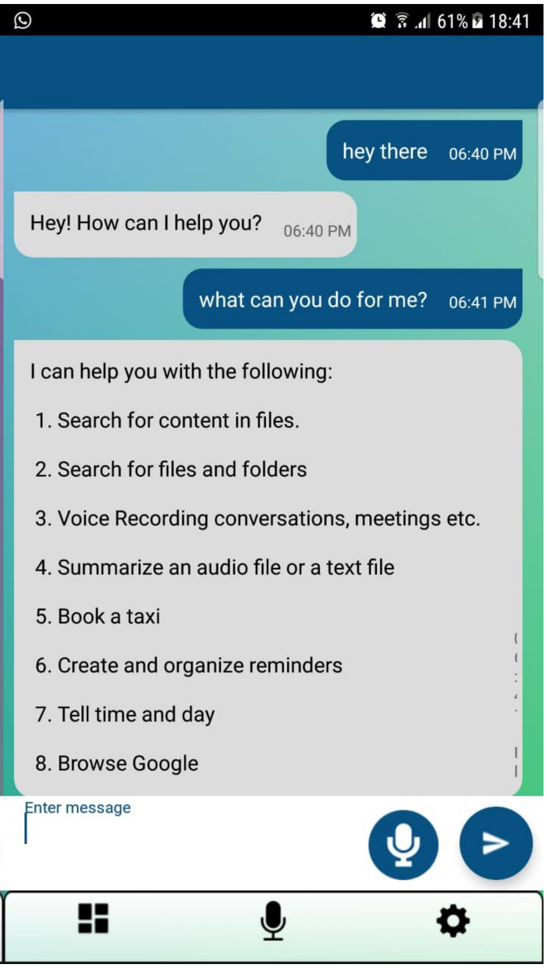
English (United States)

Your audio will be sent to Google to provide speech recognition service. A transcript will be shared with this app.



Enter message

hey there



hey there 06:40 PM

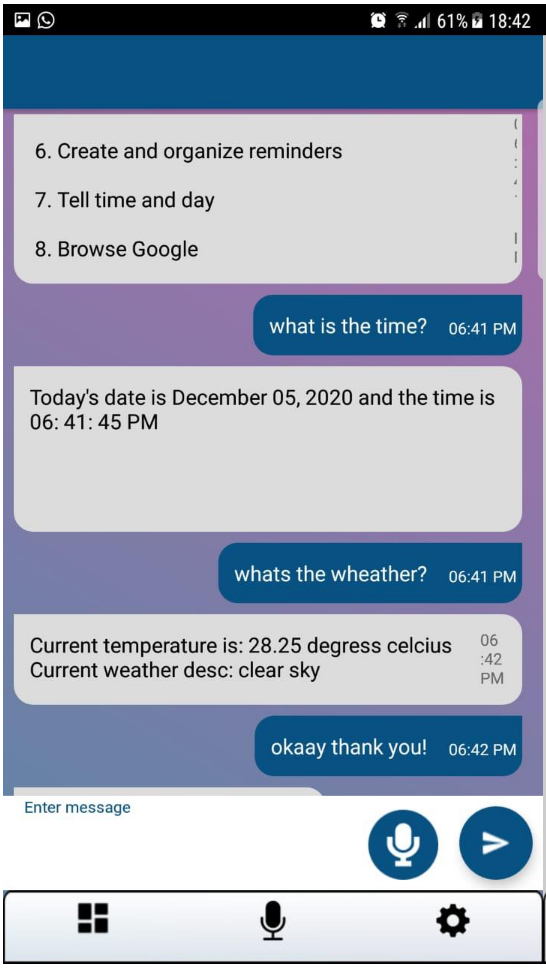
Hey! How can I help you? 06:40 PM

what can you do for me? 06:41 PM

I can help you with the following:

1. Search for content in files.
2. Search for files and folders
3. Voice Recording conversations, meetings etc.
4. Summarize an audio file or a text file
5. Book a taxi
6. Create and organize reminders
7. Tell time and day
8. Browse Google

Enter message



6. Create and organize reminders

7. Tell time and day

8. Browse Google

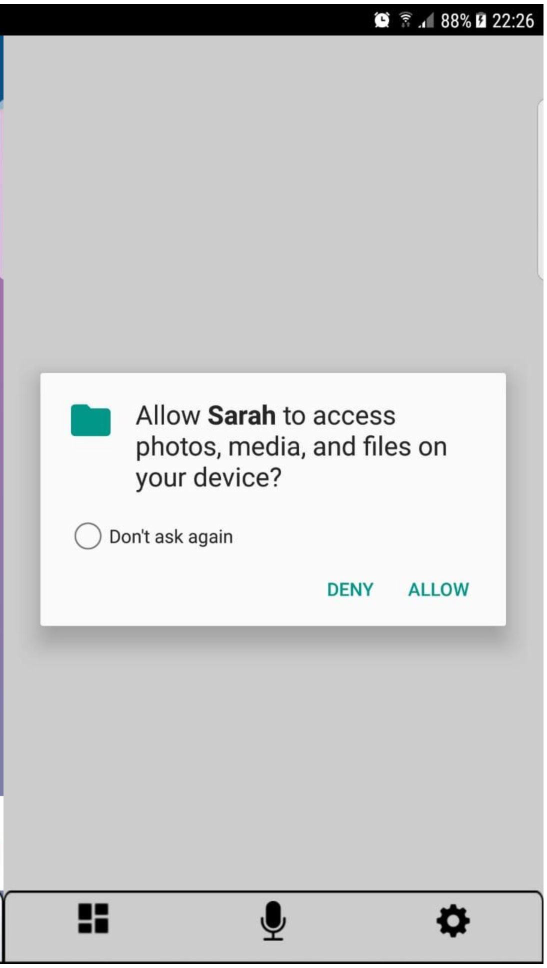
what is the time? 06:41 PM

Today's date is December 05, 2020 and the time is 06: 41: 45 PM

whats the wheather? 06:41 PM

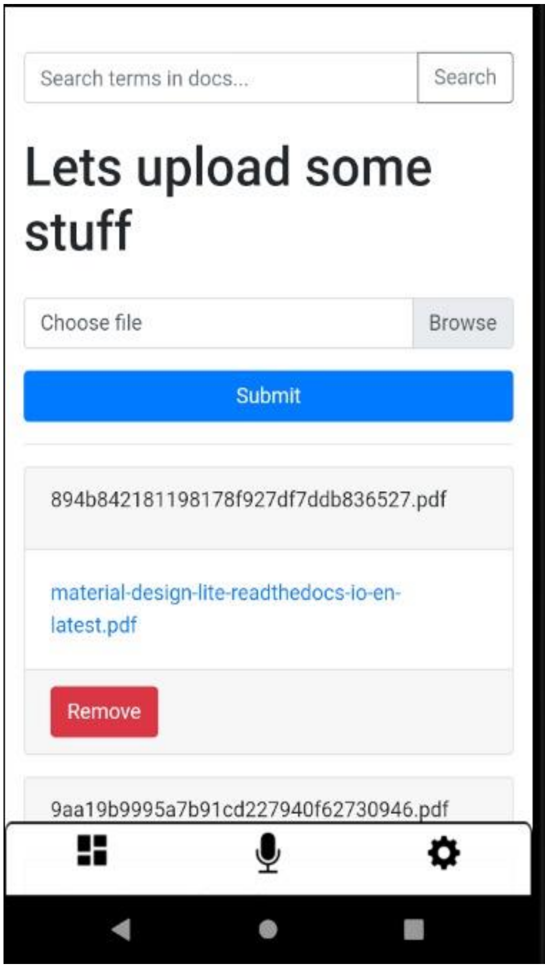
Current temperature is: 28.25 degress celcius
Current weather desc: clear sky 06 :42 PM

okaay thank you! 06:42 PM



Allow **Sarah** to access photos, media, and files on your device?

○ Don't ask again

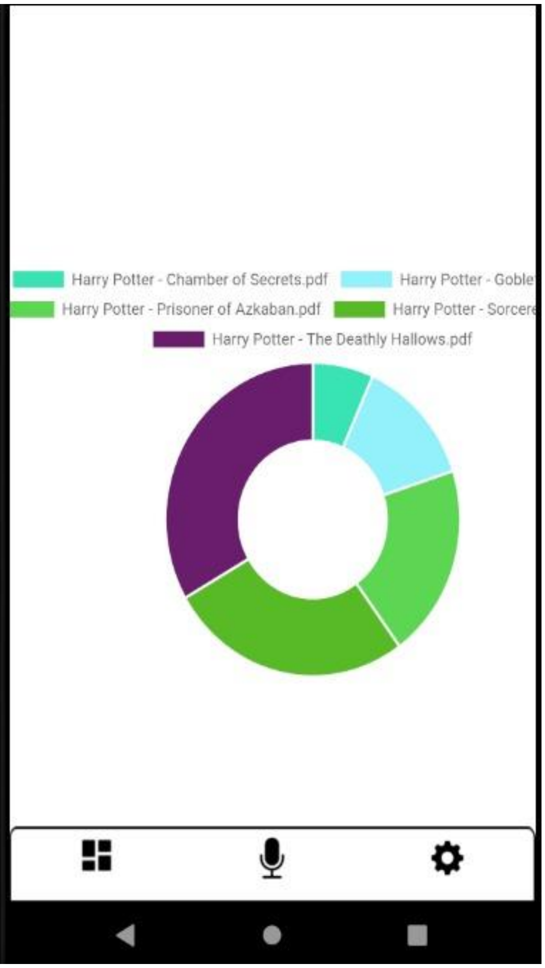


Lets upload some stuff

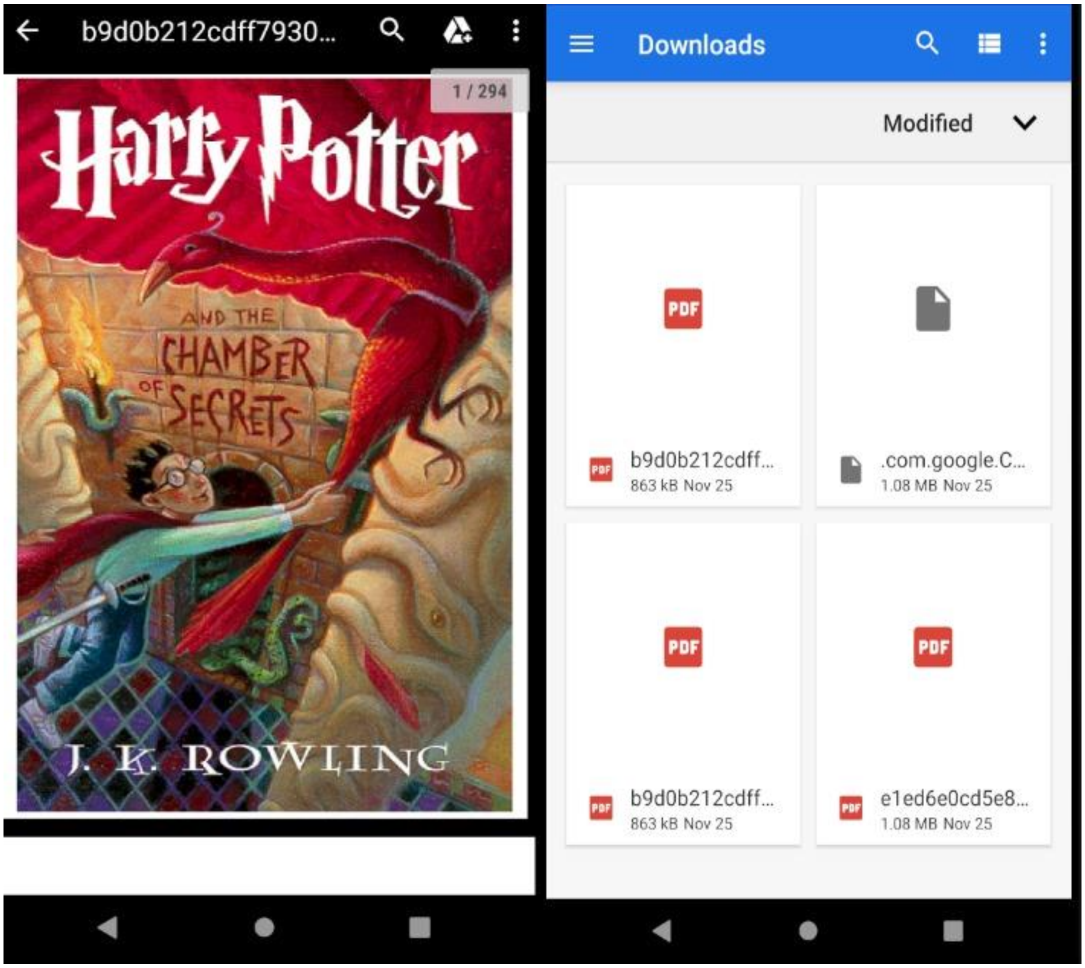
894b842181198178f927df7ddb836527.pdf

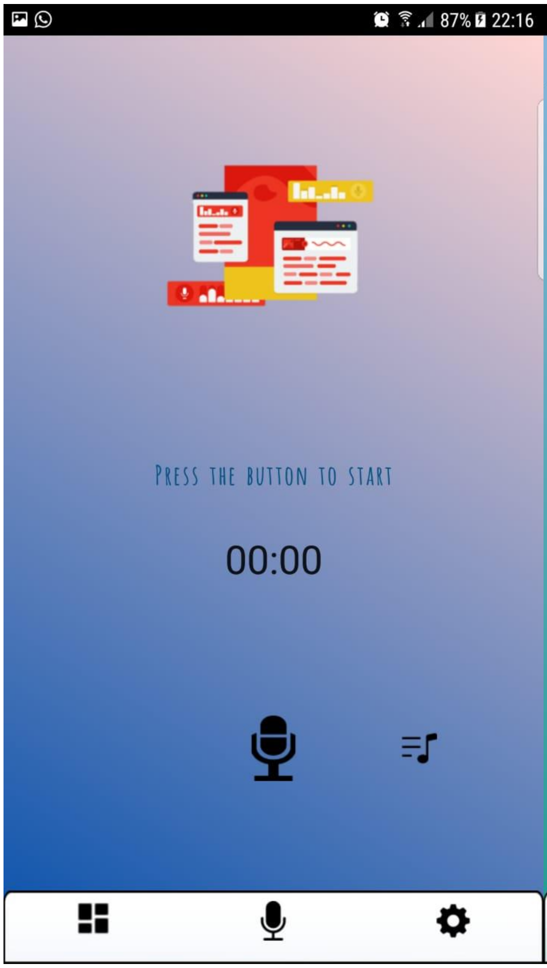
material-design-lite-readthedocs-io-en-latest.pdf

9aa19b9995a7b91cd227940f62730946.pdf

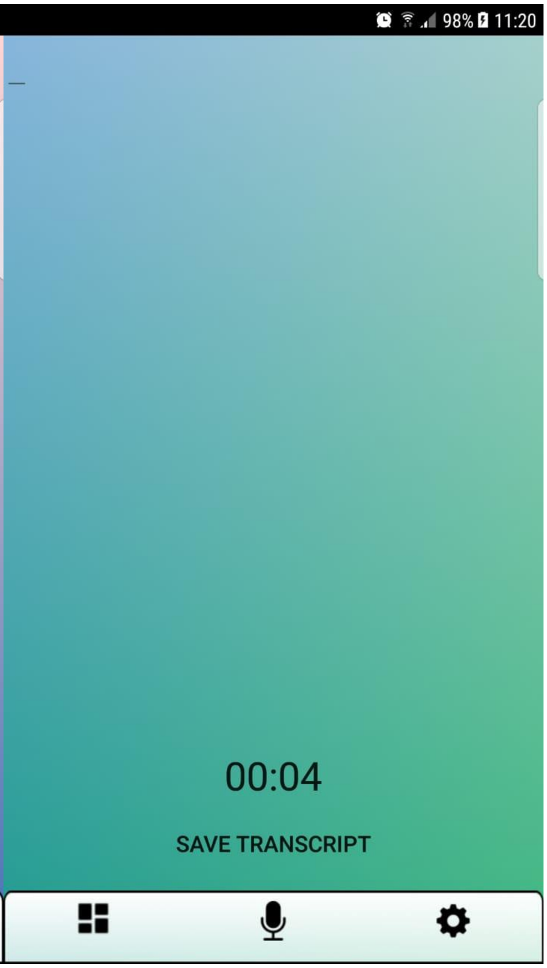


Harry Potter - Chamber of Secrets.pdf
Harry Potter - Goble
Harry Potter - Prisoner of Azkaban.pdf
Harry Potter - Sorcere
Harry Potter - The Deathly Hallows.pdf

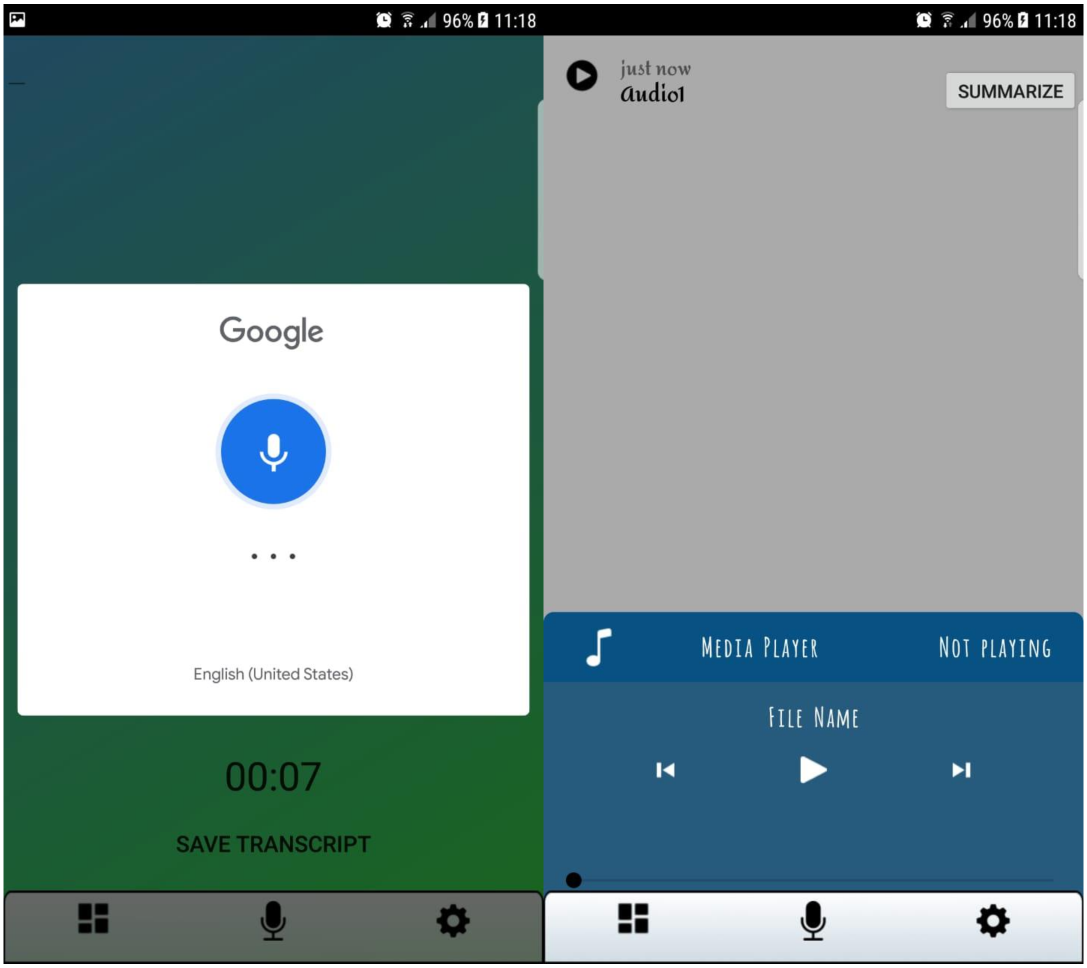

b9d0b212cdff7930...
1 / 294
Harry Potter
AND THE CHAMBER OF SECRETS
J. K. ROWLING
Downloads
Modified
b9d0b212cdff...
863 kB Nov 25
.com.google.C...
1.08 MB Nov 25
b9d0b212cdff...
863 kB Nov 25
e1ed6e0cd5e8...
1.08 MB Nov 25

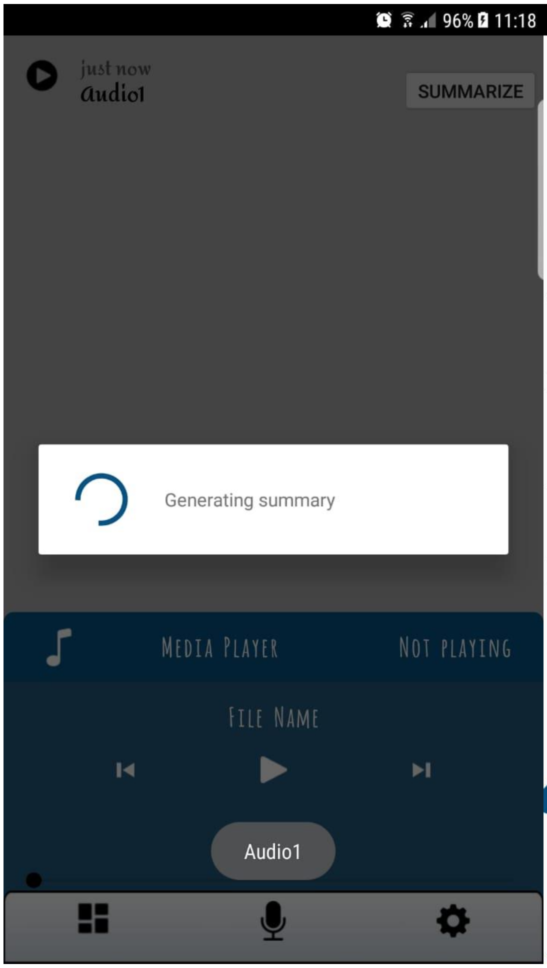


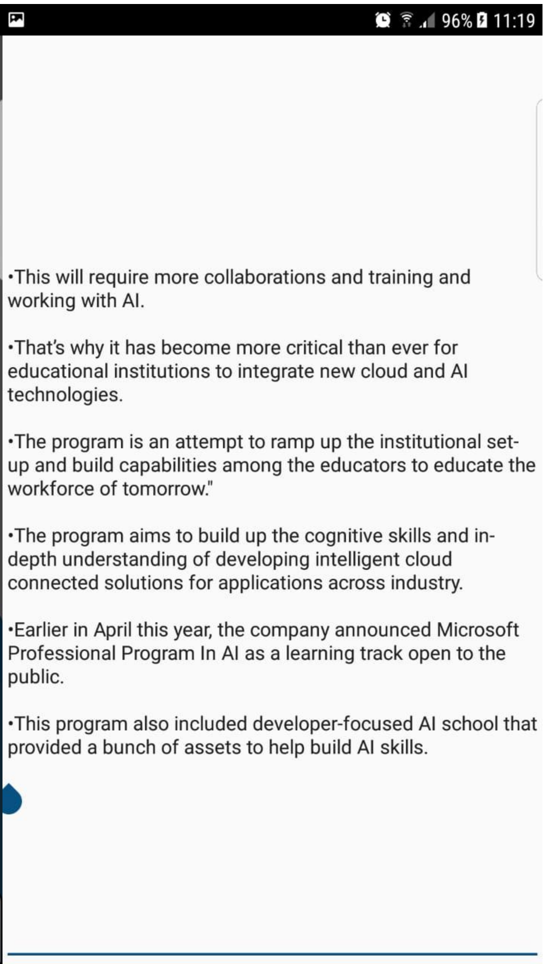
87% 22:16
Press the button to start
00:00



98% 11:20
00:04
SAVE TRANSCRIPT


96% 11:18
Google
English (United States)
00:07
SAVE TRANSCRIPT
96% 11:18
just now
audio1
SUMMARIZE
Media Player
Not playing
File Name





•This will require more collaborations and training and working with AI.

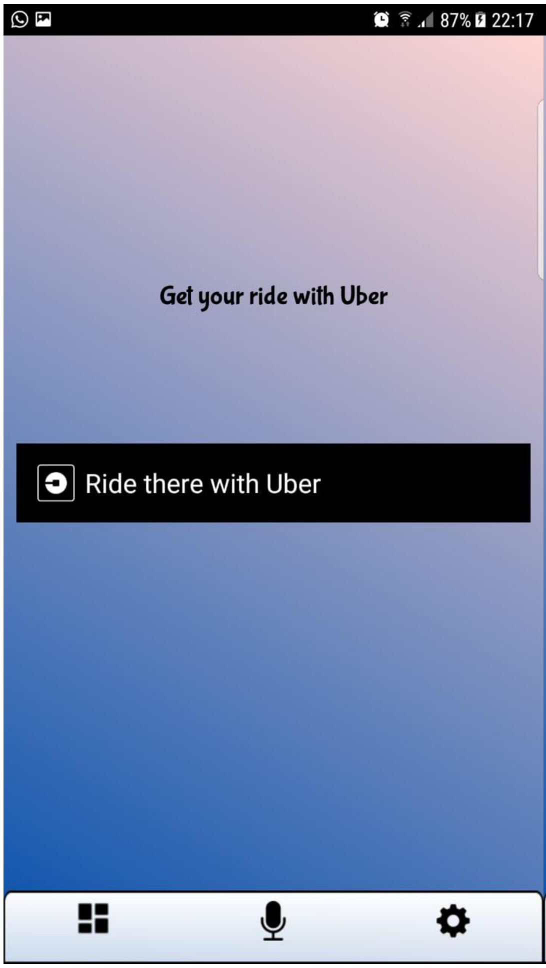
•That's why it has become more critical than ever for educational institutions to integrate new cloud and AI technologies.

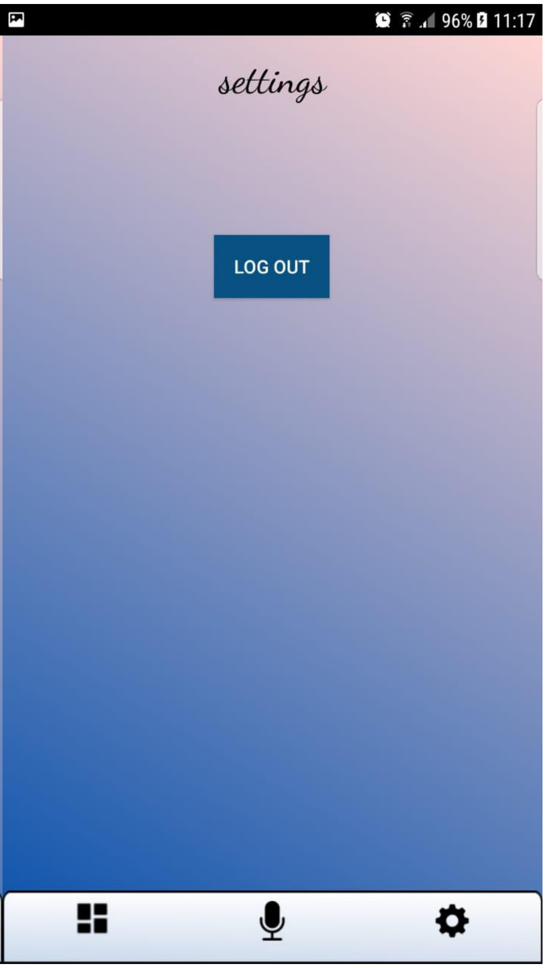
•The program is an attempt to ramp up the institutional set-up and build capabilities among the educators to educate the workforce of tomorrow."

•The program aims to build up the cognitive skills and in-depth understanding of developing intelligent cloud connected solutions for applications across industry.

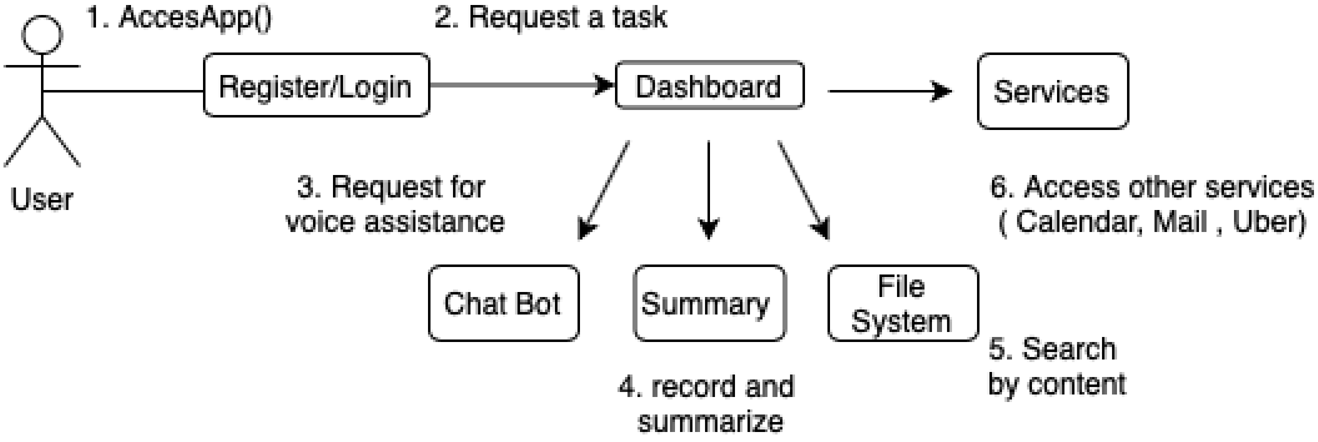
•Earlier in April this year, the company announced Microsoft Professional Program In AI as a learning track open to the public.

•This program also included developer-focused AI school that provided a bunch of assets to help build AI skills.





Collaboration diagram



5. Low-level Design

Pseudo codes

- Log In

```
Create firebase instance.
Connect to firebase authentication.
If (connection successful) {
    Get user credentials.
    Validate user credentials with firebase authentication.
    If (user not validated) then display “user not found” error.
    Else redirect to the dashboard.
}
Else display “Could not connect” error.
```

- Sign Up

```
Create firebase instance.
Connect to firebase authentication.
If (connection successful) {
    Get user credentials.
    Create new user object.
    Store new user object in firebase authenticator.
    Redirect to dashboard.
}
Else {
    Display “Could not connect” error.
    Redirect to login page.
}
```

- File System

```
get keyword
searchList = search(keyword)
clusterArray = cluster(searchList) uses K-means
Visualize(clusterArray) displays using charts.js
```

Cluster()

utilizes a clustering algorithm called k-means to store the results in a cluster. K-means is an optimal clustering algorithm that has been utilized in multiple implementations of clustering. "It is a point-based clustering method that starts with the cluster enters initially planned at arbitrary positions and proceeds by moving at each step of the cluster enters in order to minimize the clustering error. The main disadvantage of the method lies in its sensitivity to initial positions of the cluster enters" (Likas, Vlassis and J. Verbeek, 2003).

Visualize()

Takes the cluster formed using the cluster function and then takes the input from the cluster and transforms them into a visual cluster in the form of an interactive sunburst chart using an open-source python library called charts.js

- Note Maker

Android Studio:

- Microphone Button

```
Check microphone access permission.
    Redirected to new screen
    Recorder started in background
    If (permission has been granted) {
    Connect to Google speech API
    If (connection to API successful) {
    Get user speech input
    Convert speech input to text and store in string variable in realtime.
    Display text in user edit box to summarize .
        }
        Else display “Could not initiate speech to text converter” error.
    Else {
        Ask for permission.
        If (permission has been granted) {
            Re-call microphone button method.
        }
        Else display “Permission not granted” error.
    }
```

- Save Transcript Button

```
Recorder stopped in background
Recording shows up on recycler view in new fragment to be replayed by user
Connect to Firebase Storage
Transcript saved to Firebase Storage media
```

Python:

```
Connect to Firebase Storage
Get Transcript from Firebase Storage media for the audio that is selected by
checking
If (Audio name == Transcript name that exists in firebase) {
    Select that Transcript
    Open a new file
    Save the Transcript content in new file
    Read file then summarize
    Save Transcript in firebase
```

- Summarize function()

```
Clean the text(Remove stop words, remove punctuations)
Lemmaize , tokenize
```

```
Apply TFIDF algorithm to perform summary
```

Android Studio:

```
Retrieve Summarized Transcript file from firebase
Save the Transcript content in new file in device internal storage
Display Summarized Transcript in editable new screen.
```

- Voice Assistant

Python:

```
Get data from user query or training data.
If (data from user query) {
    Wait for user query.
    Get user query and store it in string variable called query.
}
Else {
    Store training data as query.
}
Remove whitespaces from query.
Perform lemmatization of words from query.
Remove stop words from query.
Perform bag of words operation from query.
Get number of elements depending on the words in the query.
Convert n-Dimensional array to 2-dimensional array.
Optimize data loss during conversion as much as possible.
Perform supervised learning using regression techniques to predict pattern.
Perform backpropagation to predict intent based on supervised learning.
Get intent.
Based on intent choose appropriate response using pre-built mapping policy in RASA.
If (intent was successfully predicted) return appropriate response or action.
Else return “Sorry, did not understand.”
```

NGROK:

```
Host the chat bot in local host.
Use NGROK to host from locally to publicly and establish communication between the python bot and android studio.
```

Android Studio:

```
Microphone Button
    Check microphone access permission.
    If (permission has been granted) {
```

```
    Connect to Google speech API
    If (connection to API successful) {
        Get user speech input
        Convert speech input to text and store in string variable.
        Display text in user edit box of the chat bot’s layout.
    }
    Else display “Could not initiate speech to text converter” error.
Else {
    Ask for permission.
    If (permission has been granted) {
        Re-call microphone button method.
    }
    Else display “Permission not granted” error.
}
```

Send Button:

```
Get text from the user edit box of the chat bot’s layout.
Check if text is empty.
If (text is empty) display “Sorry, did not understand.”
Else {
    Connect to bot’s public network address.
    If (connection successful) {
        Send user query.
        Get bot response.
        Convert bot response to text format.
        Connect to Google speech API.
        If (connection successful) {
            Get bot text response.
            Convert bot response to speech.
            Play the speech.
        }
        Get reference to text box in chat bot’s layout.
        Display bot response in the text box.
    }
    Else display “Could not connect to the bot” error.
}
```

- Contacts

```
Get contact access permission.
If (Permission granted) open contacts application.
Else display “Permission for contacts not granted” error.
```

- Uber

```
Check if Uber application is installed.
If (application is installed) {
    Get user request.
```

```
        Request to book a taxi or cancel the ride.
        If (book a taxi) open book a taxi page on Uber application.
        Else if (cancel a taxi) open cancel taxi page on Uber application.
        Else open homepage on Uber application.
}
Else {
        Redirect to Uber’s page on Google play store.
}
```

- Email

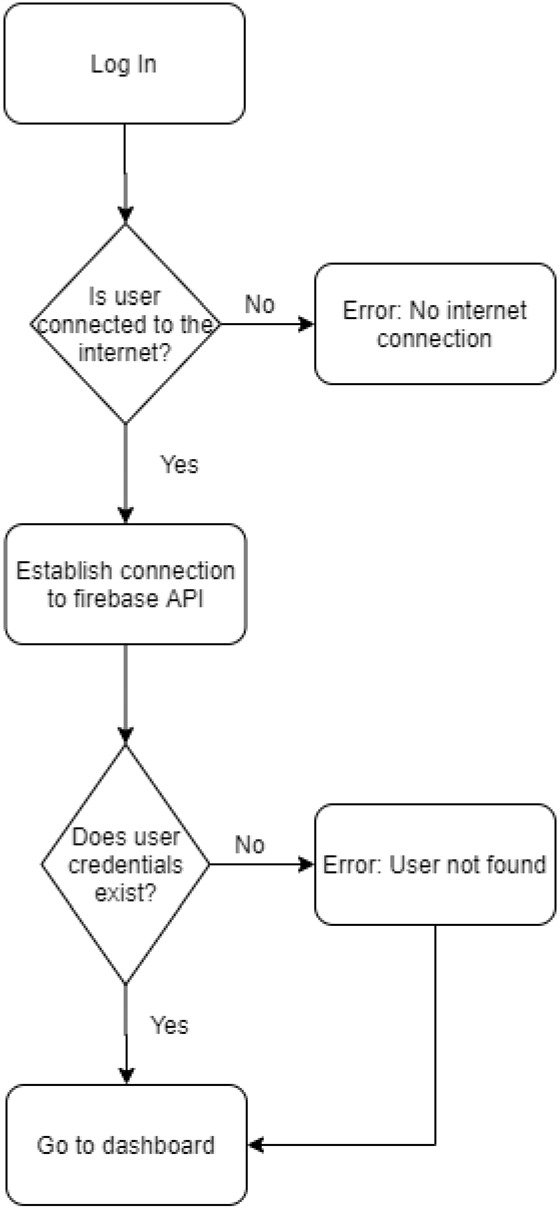
```
Get email access permission.
If (Permission granted) open email application.
Else display “Permission for email not granted” error.
```

- Calendar

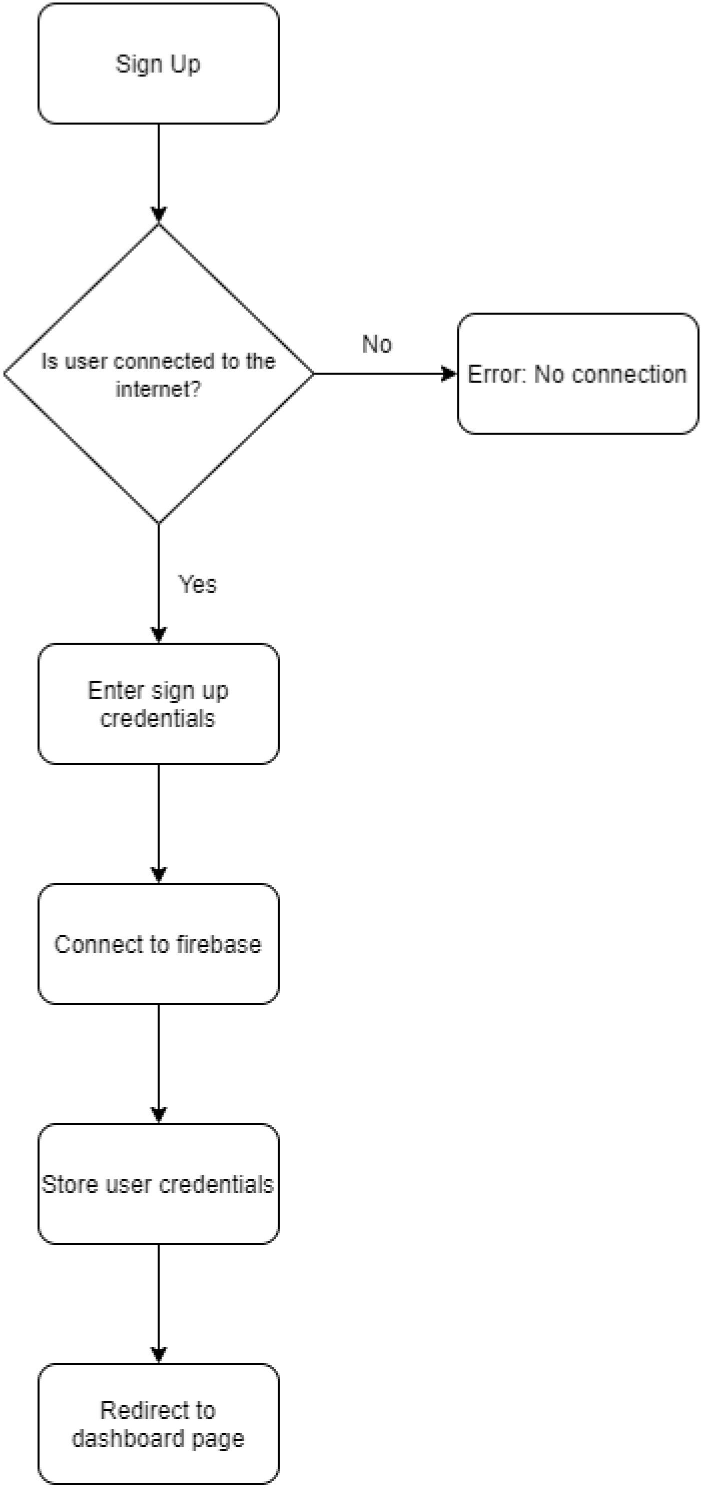
```
Get calendar access permission.
If (Permission granted) open calendar application.
Else display “Permission for calendar not granted” error.
```

Activity Diagrams

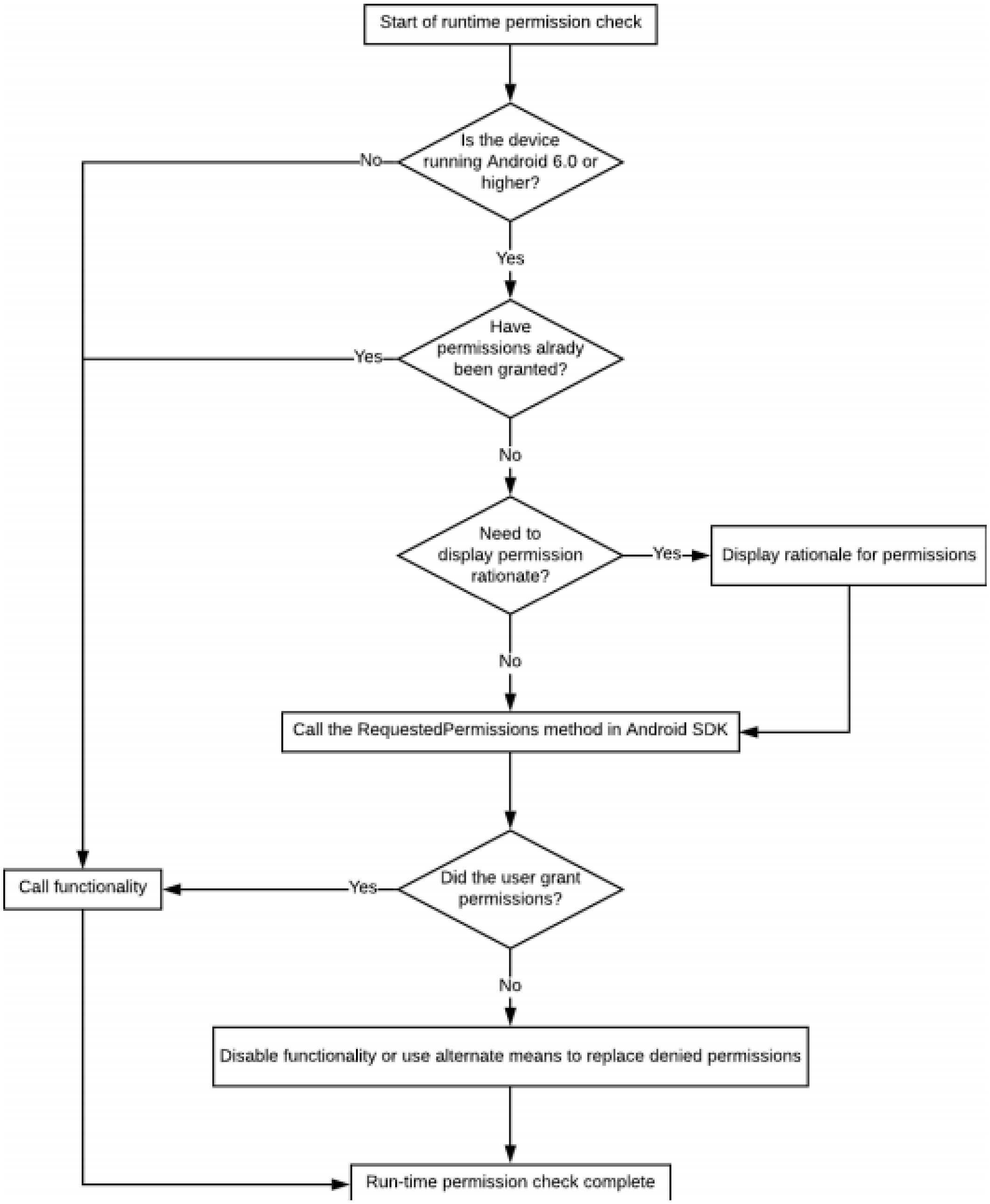
- Log In


Log In
Is user connected to the internet?
No
Error: No internet connection
Yes
Establish connection to firebase API
Does user credentials exist?
No
Error: User not found
Yes
Go to dashboard

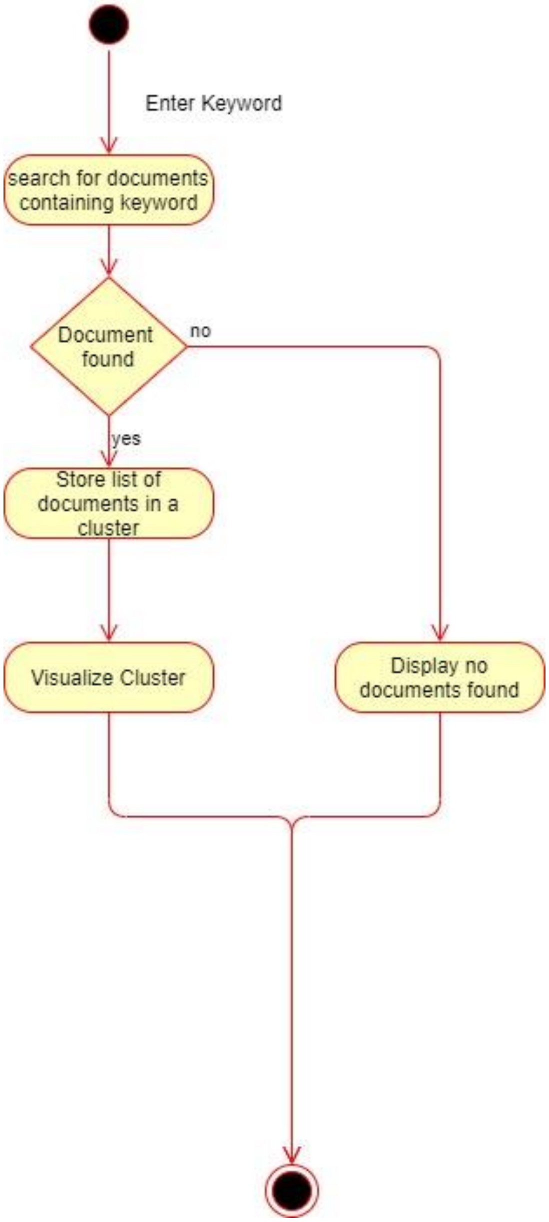
- Sign Up


Sign Up
Is user connected to the internet?
No
Error: No connection
Yes
Enter sign up credentials
Connect to firebase
Store user credentials
Redirect to dashboard page

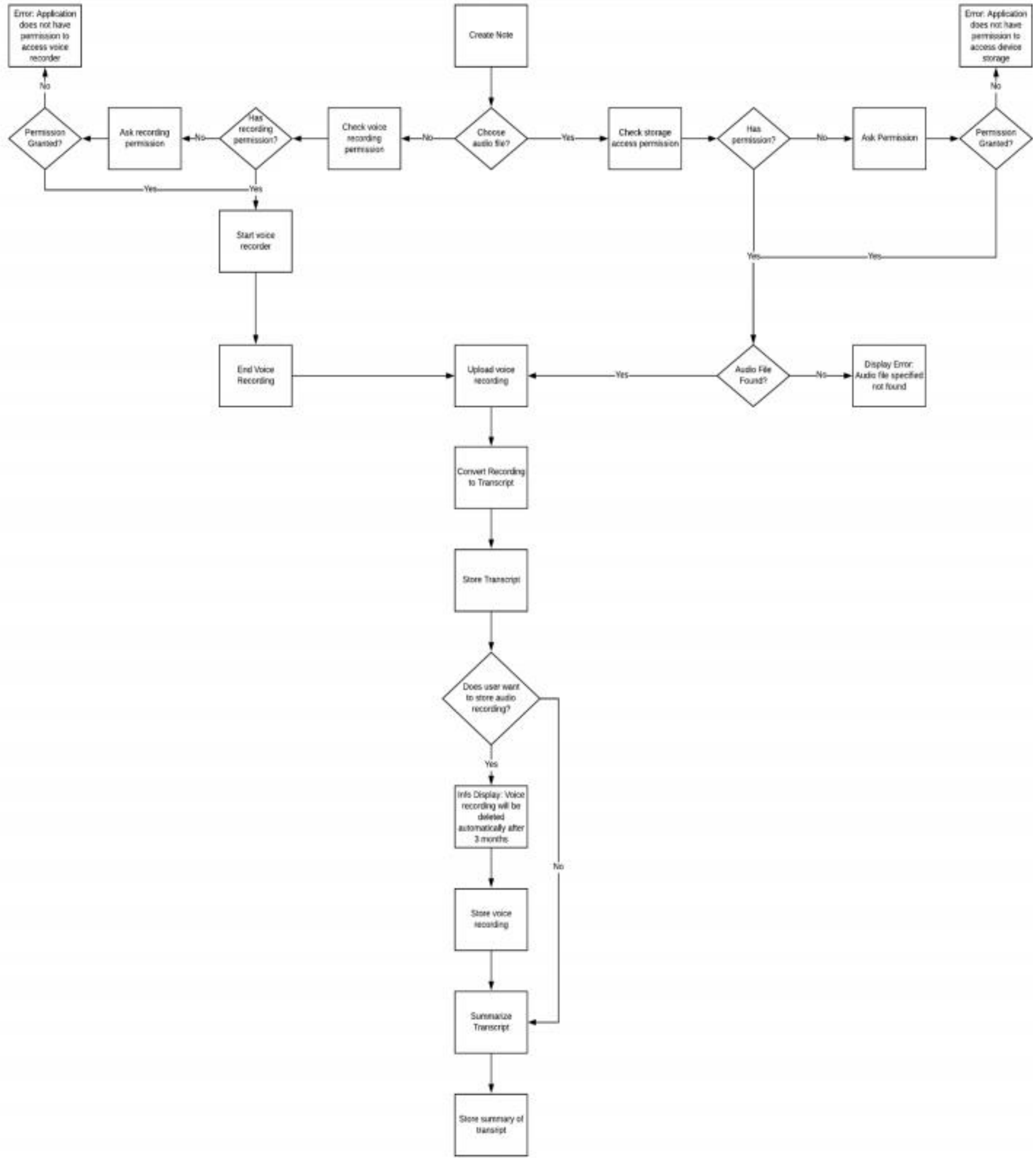
- Permission


Start of runtime permission check
Is the device running Android 6.0 or higher?
No
Yes
Have permissions alrady been granted?
Yes
No
Need to display permission rationale?
Yes
Display rationale for permissions
No
Call the RequestedPermissions method in Android SDK
Did the user grant permissions?
Yes
Call functionality
No
Disable functionality or use alternate means to replace denied permissions
Run-time permission check complete

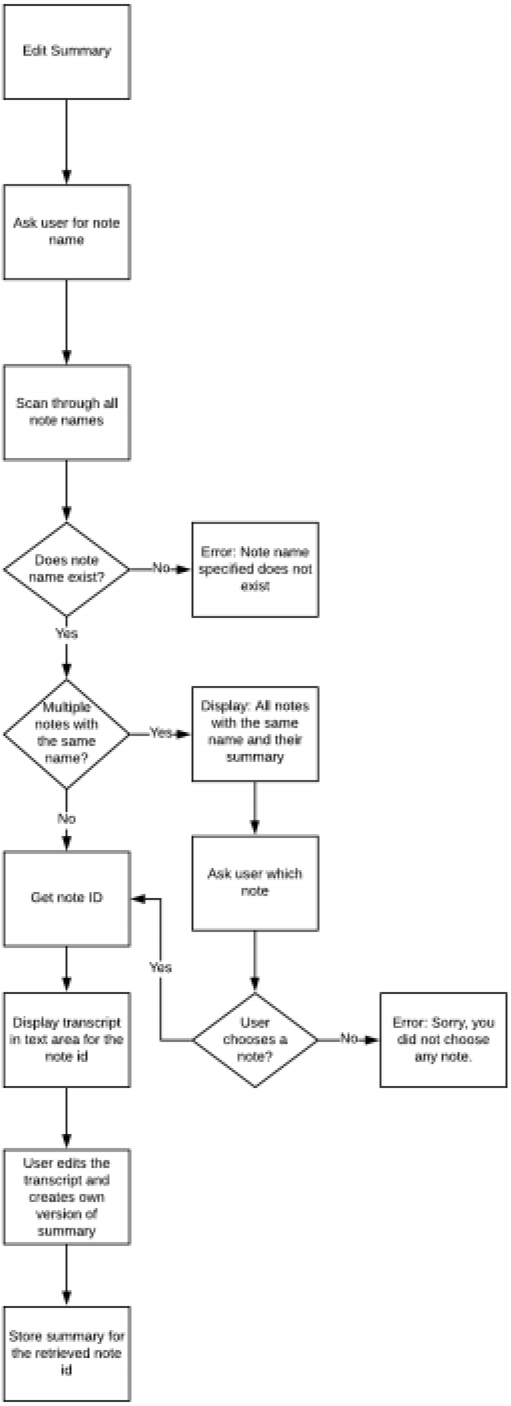
- File System


Enter Keyword
search for documents containing keyword
Document found
no
yes
Store list of documents in a cluster
Visualize Cluster
Display no documents found

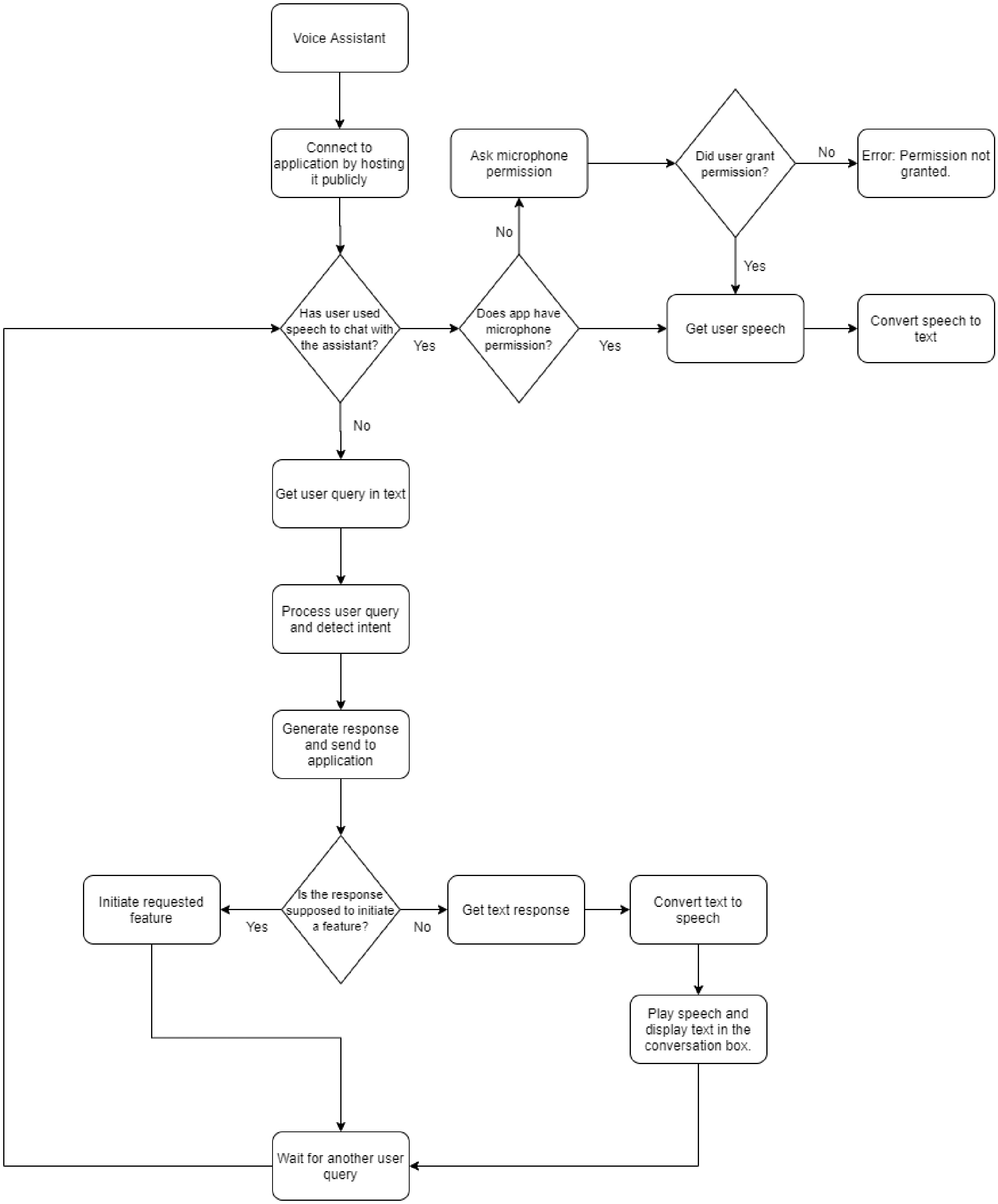
- Summary: Note Taking


Create Note
Choose audio file?
No
Check voice recording permission
Has recording permission?
No
Ask recording permission
Permission Granted?
No
Error: Application does not have permission to access voice recorder
Yes
Yes
Start voice recorder
End Voice Recording
Yes
Check storage access permission
Has permission?
No
Ask Permission
Permission Granted?
No
Error: Application does not have permission to access device storage
Yes
Yes
Audio File Found?
No
Display Error: Audio file specified not found
Yes
Upload voice recording
Convert Recording to Transcript
Store Transcript
Does user want to store audio recording?
Yes
Info Display: Voice recording will be deleted automatically after 3 months
No
Store voice recording
Summarize Transcript
Store summary of transcript

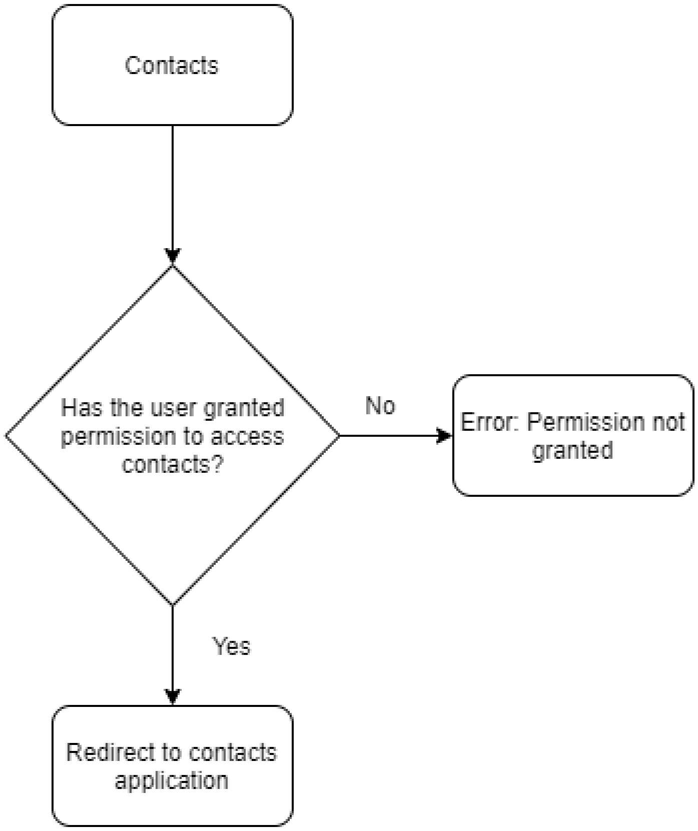
- Edit Summary


Edit Summary
Ask user for note name
Scan through all note names
Does note name exist?
No
Error: Note name specified does not exist
Yes
Multiple notes with the same name?
Yes
Display: All notes with the same name and their summary
Ask user which note
No
Get note ID
Yes
User chooses a note?
No
Error: Sorry, you did not choose any note.
Display transcript in text area for the note id
User edits the transcript and creates own version of summary
Store summary for the retrieved note id

- Voice Assistant

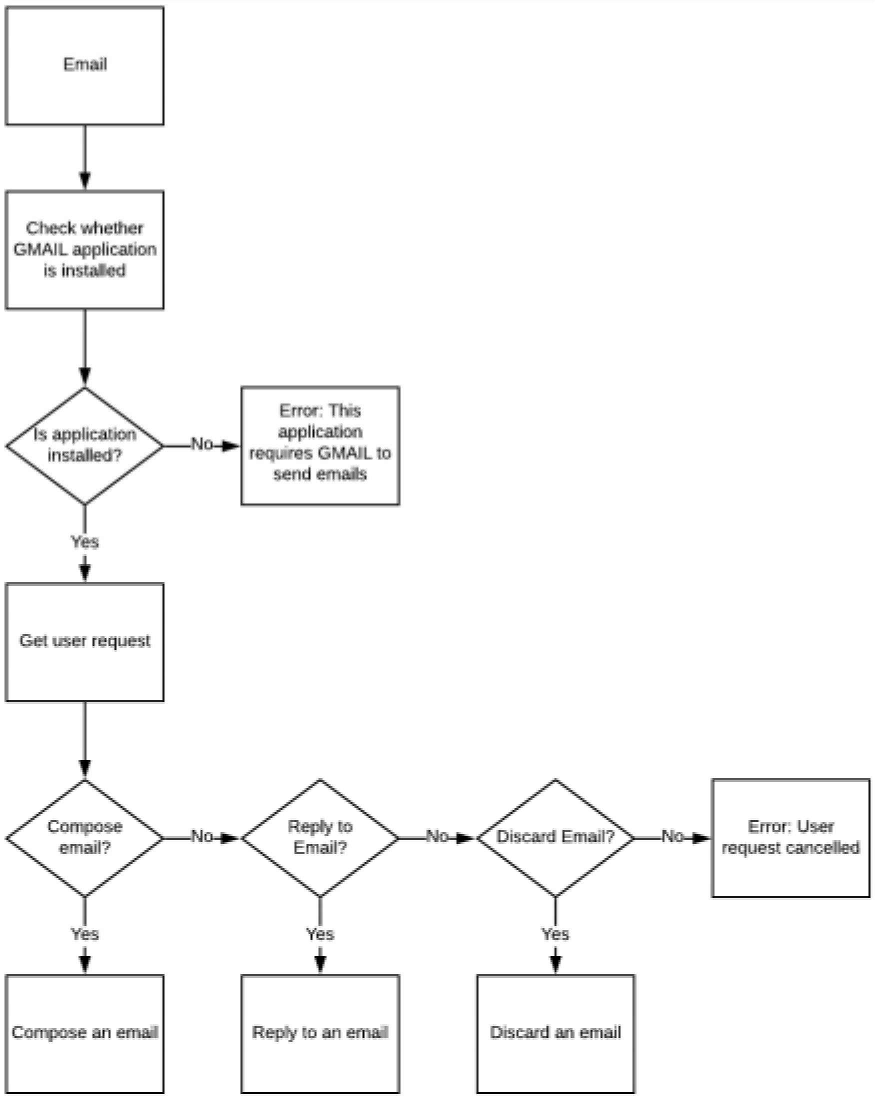

Voice Assistant
Connect to application by hosting it publicly
Ask microphone permission
Did user grant permission?
No
Error: Permission not granted.
No
Yes
Has user used speech to chat with the assistant?
Yes
Does app have microphone permission?
Yes
Get user speech
Convert speech to text
No
Get user query in text
Process user query and detect intent
Generate response and send to application
Initiate requested feature
Yes
Is the response supposed to initiate a feature?
No
Get text response
Convert text to speech
Play speech and display text in the conversation box.
Wait for another user query

- Contacts

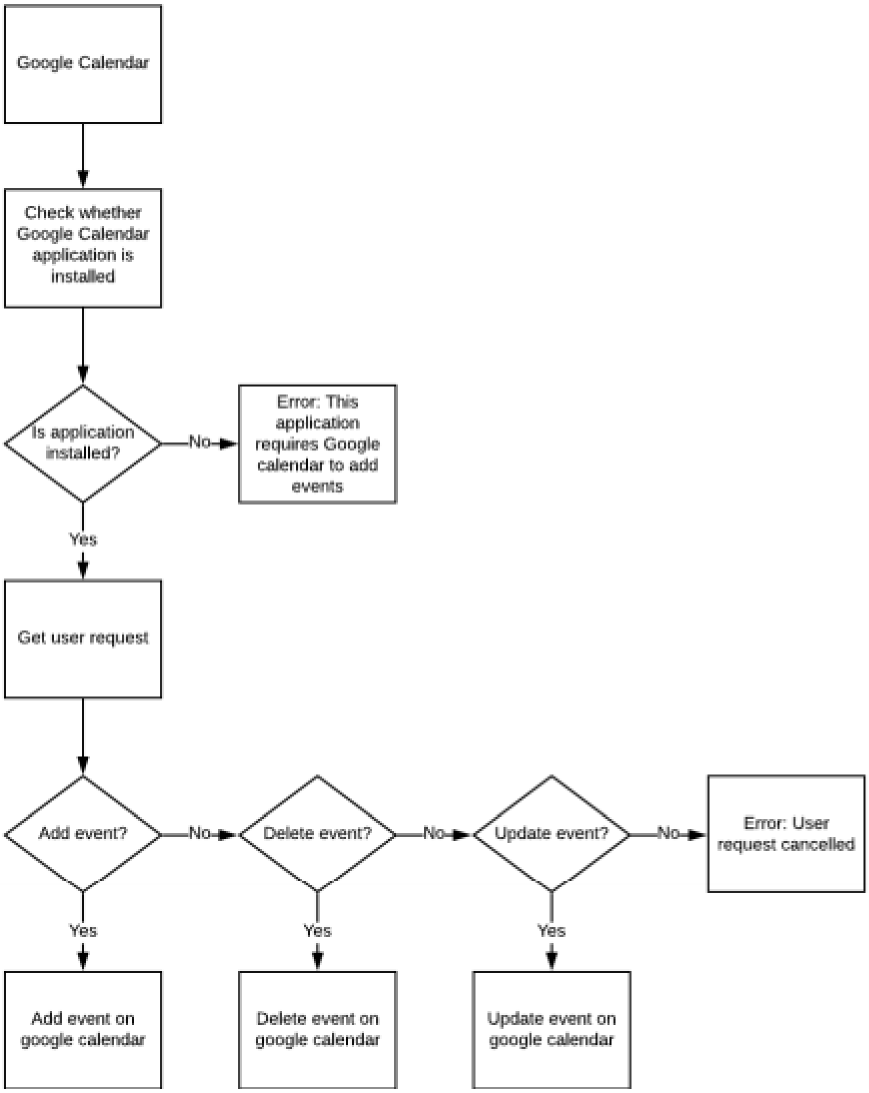

Contacts
Has the user granted permission to access contacts?
No
Error: Permission not granted
Yes
Redirect to contacts application

- Uber

- Email


Email
Check whether GMAIL application is installed
Is application installed?
No
Error: This application requires GMAIL to send emails
Yes
Get user request
Compose email?
No
Reply to Email?
No
Discard Email?
No
Error: User request cancelled
Yes
Yes
Yes
Compose an email
Reply to an email
Discard an email

- Calendar



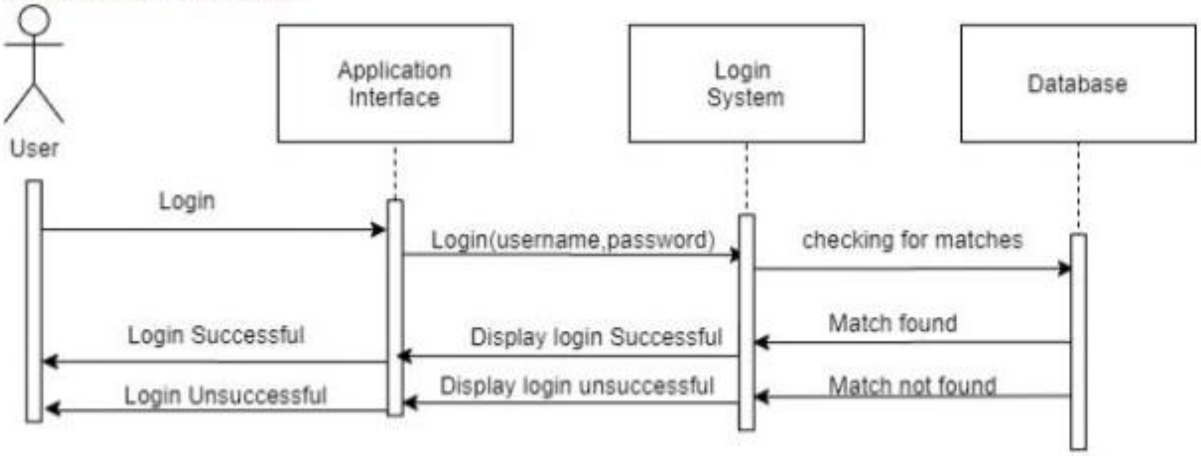
Google Calendar
Check whether Google Calendar application is installed
Is application installed?
No
Error: This application requires Google calendar to add events
Yes
Get user request
Add event?
No
Delete event?
No
Update event?
No
Error: User request cancelled
Yes
Yes
Yes
Add event on google calendar
Delete event on google calendar
Update event on google calendar

Sequence diagrams

- Log In

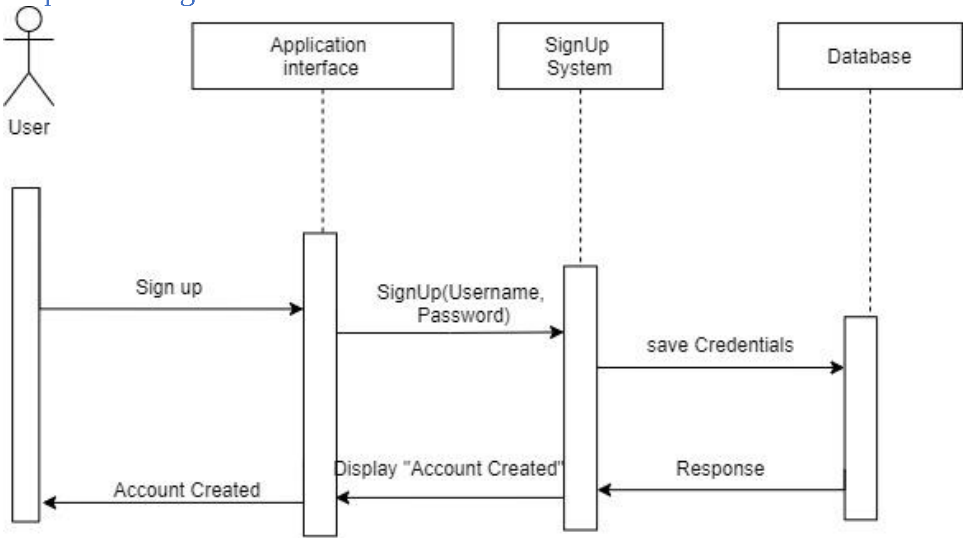
LOGIN :

Sequence Diagrams



- Sign Up

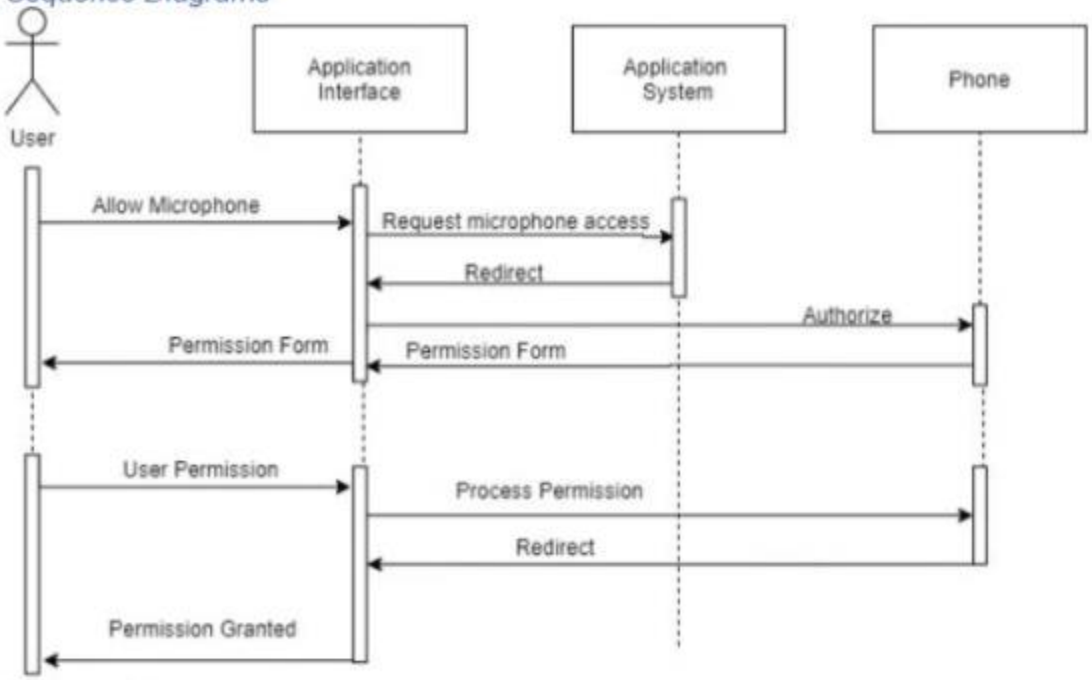
Sequence Diagram



- Permission

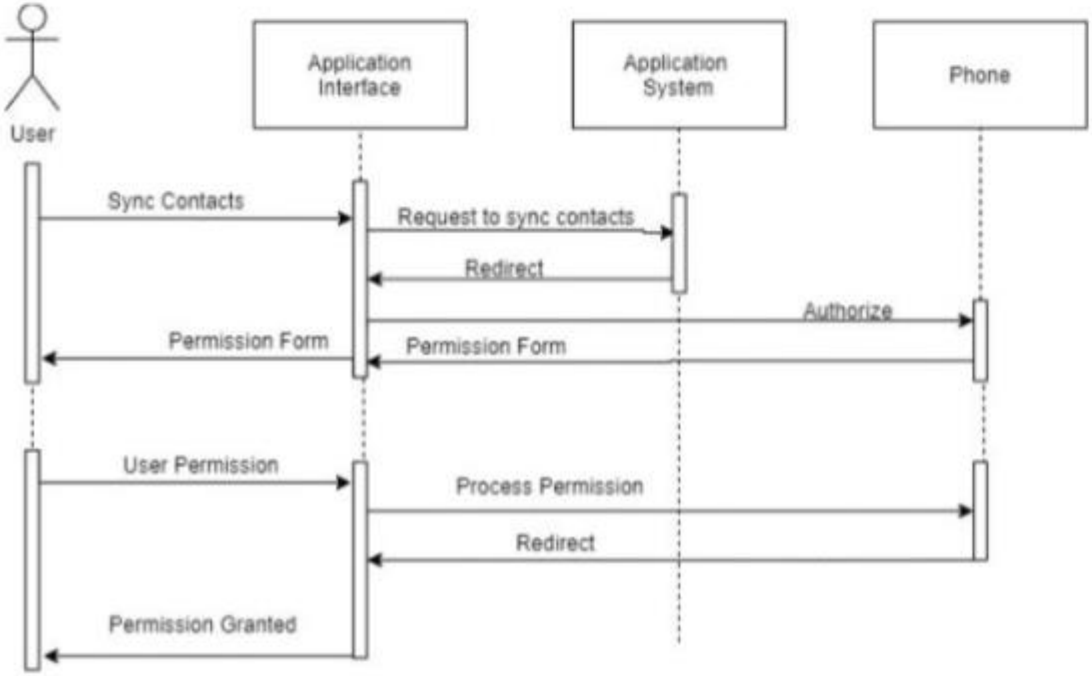
Access microphone

Sequence Diagrams



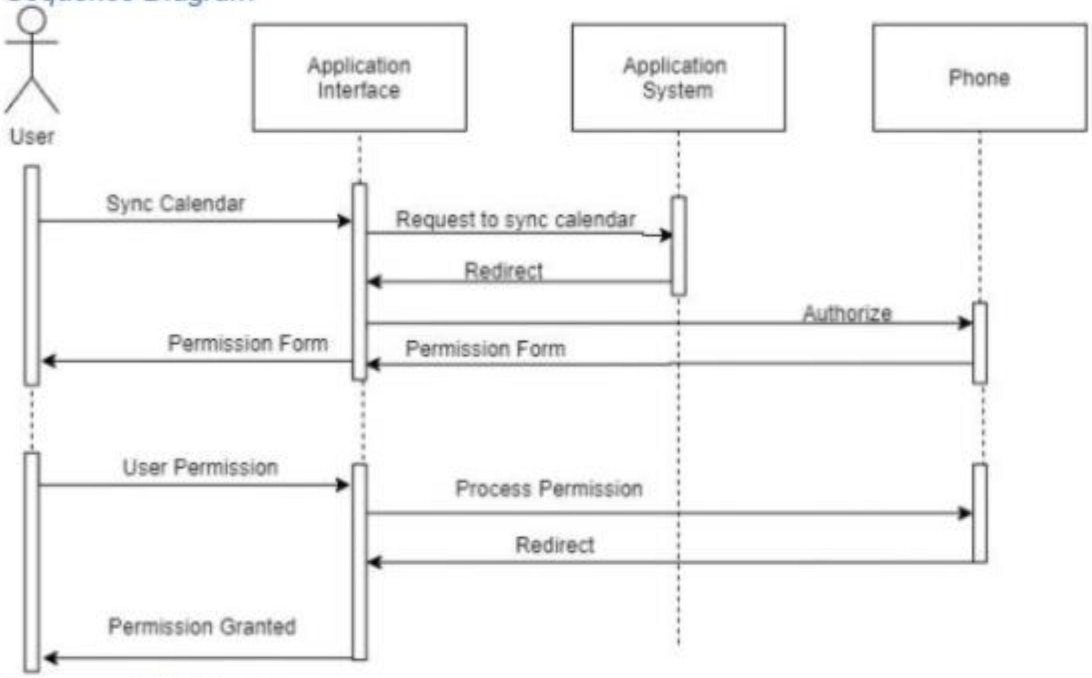
Sync contacts

Sequence Diagram



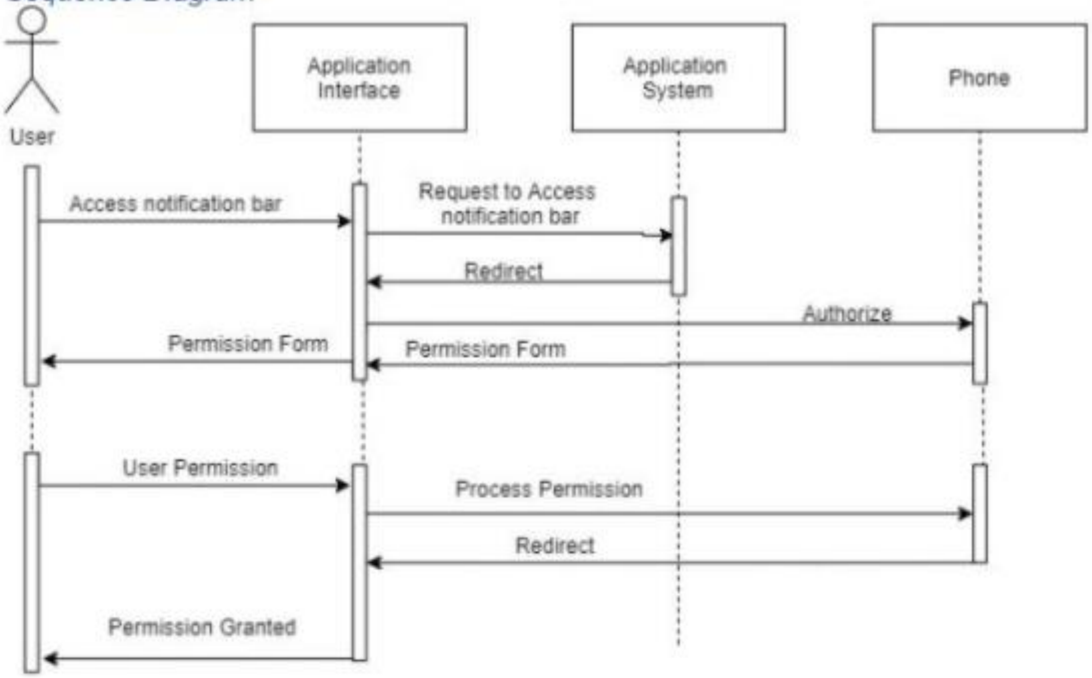
Sync calendar

Sequence Diagram



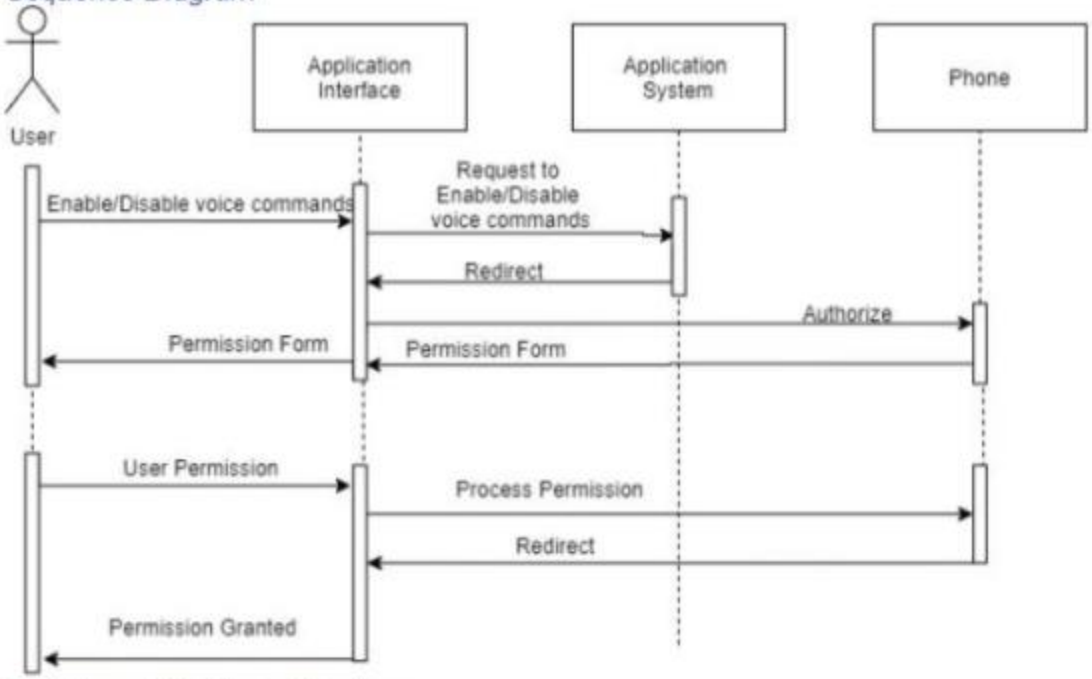
Access notification bar

Sequence Diagram



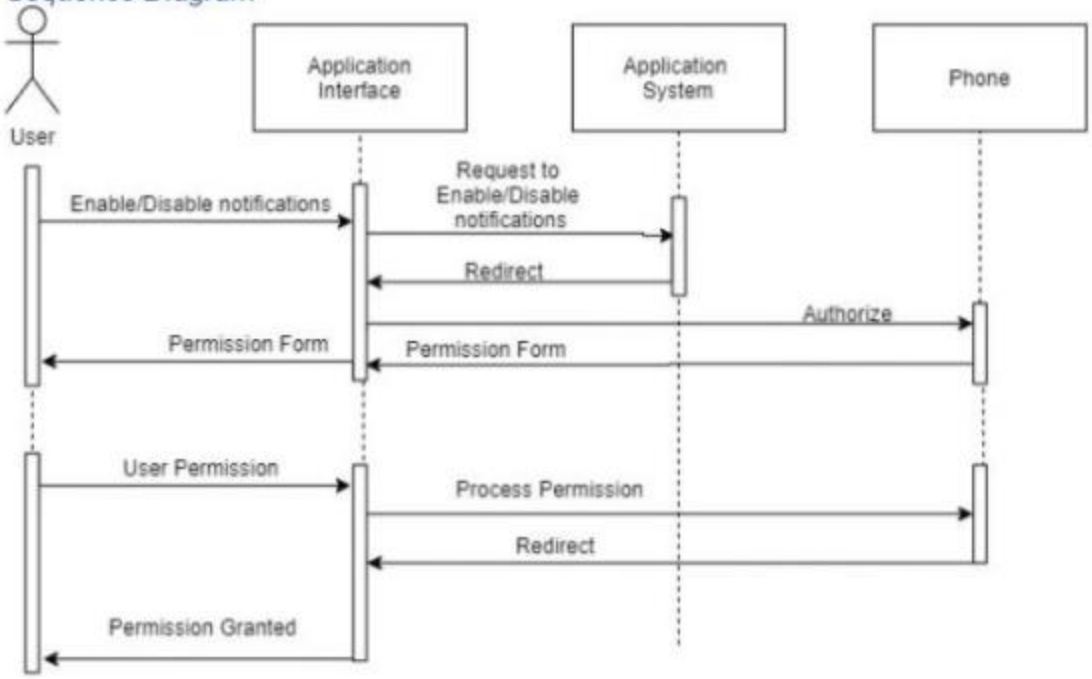
Enable and disable voice commands

Sequence Diagram



Enable and disable notifications

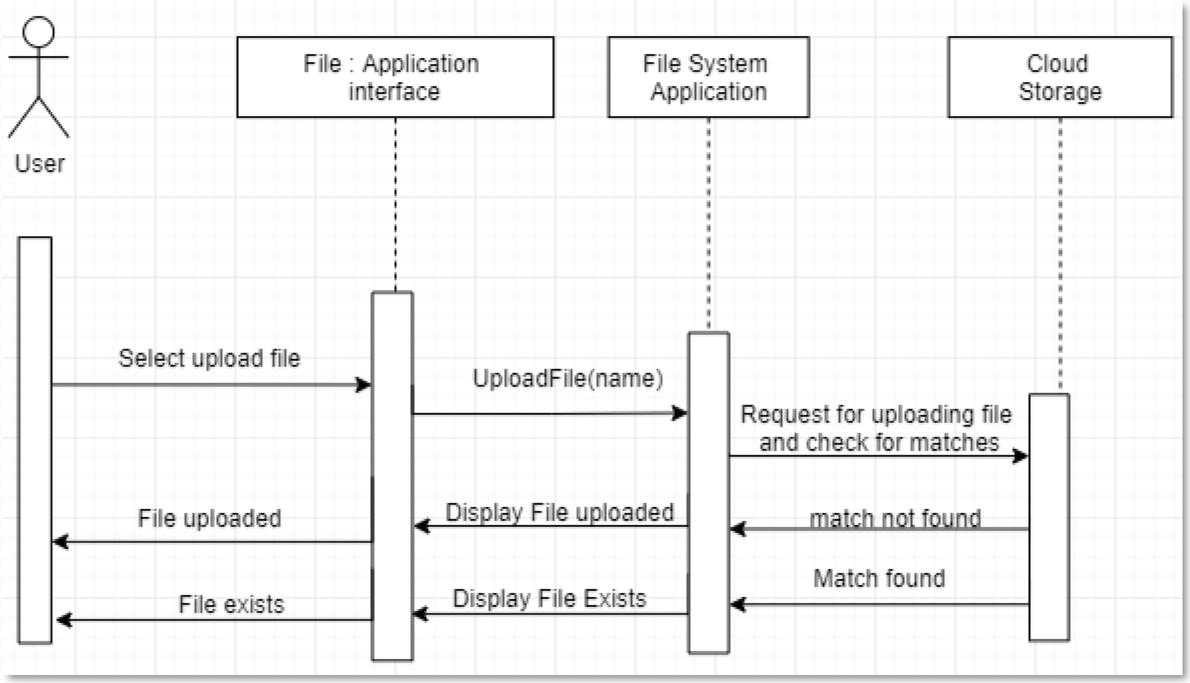
Sequence Diagram



- File System

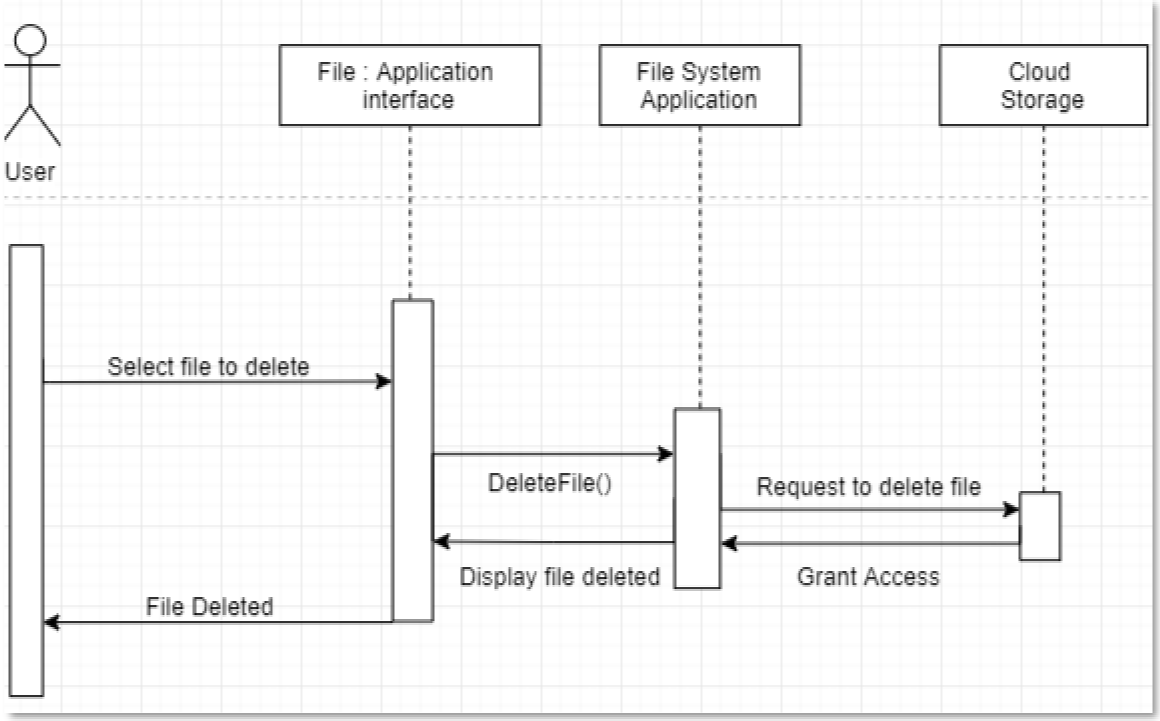
Upload file

Sequence Diagram



Delete file

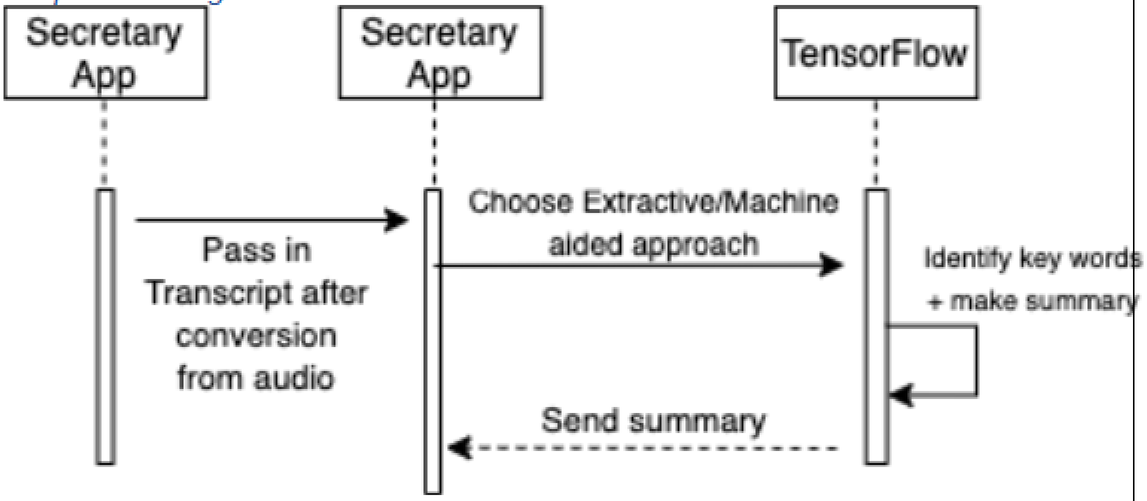
Sequence Diagram



- Note Maker

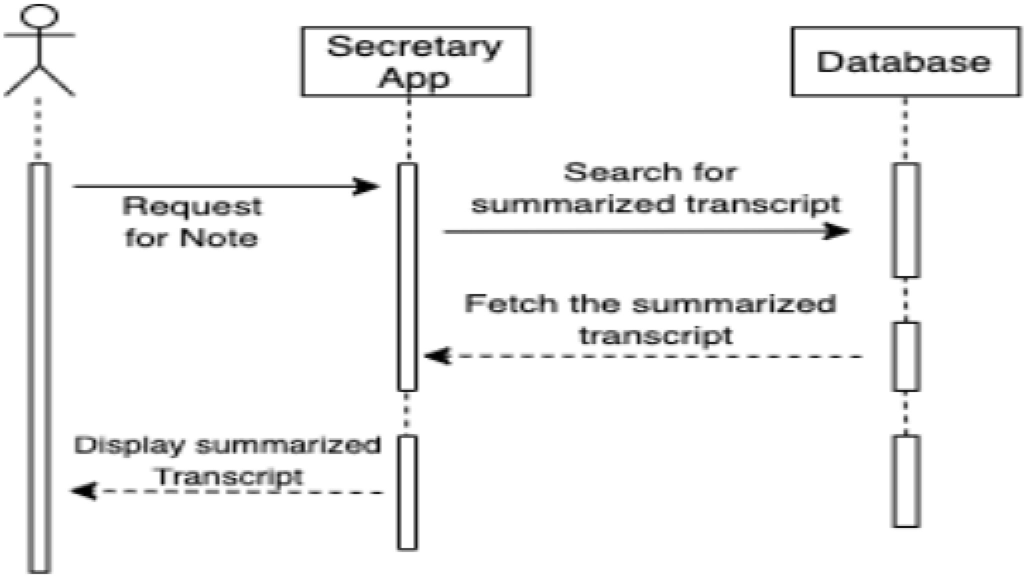
Text Summarizer

Sequence Diagram


Secretary App
Secretary App
TensorFlow
Pass in Transcript after conversion from audio
Choose Extractive/Machine aided approach
Identify key words + make summary
Send summary

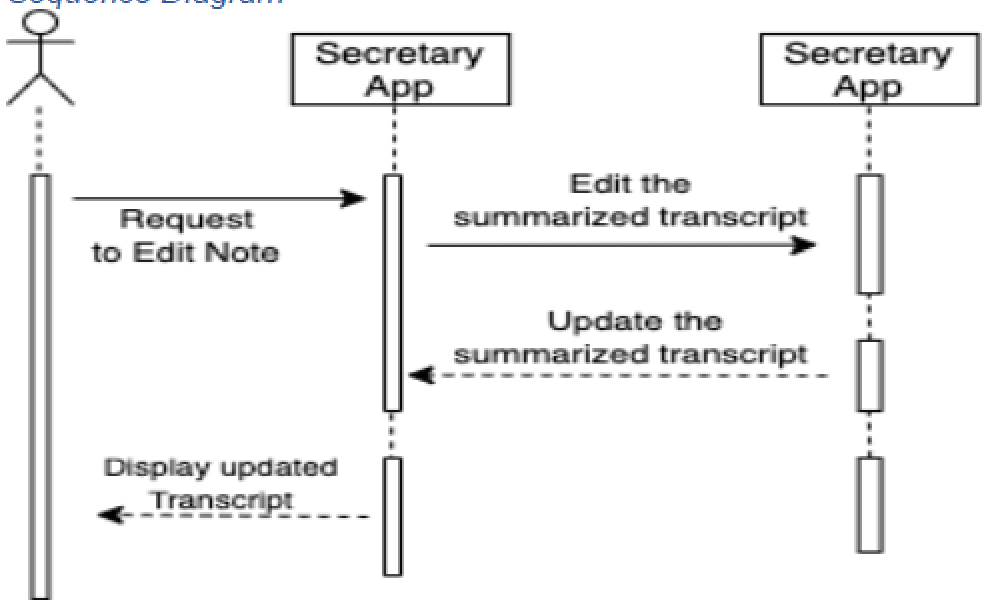
Display summarized text

Sequence Diagram


Secretary App
Database
Request for Note
Search for summarized transcript
Fetch the summarized transcript
Display summarized Transcript

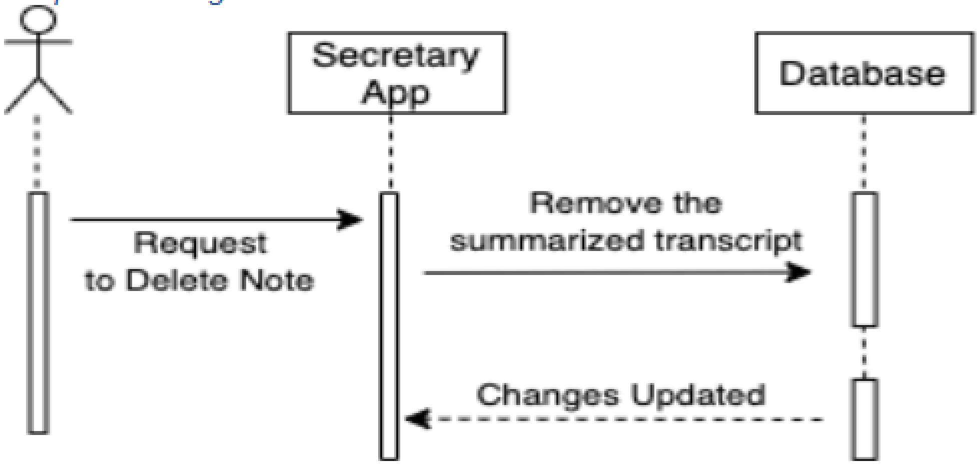
Edit Note

Sequence Diagram


Secretary App
Secretary App
Request to Edit Note
Edit the summarized transcript
Update the summarized transcript
Display updated Transcript

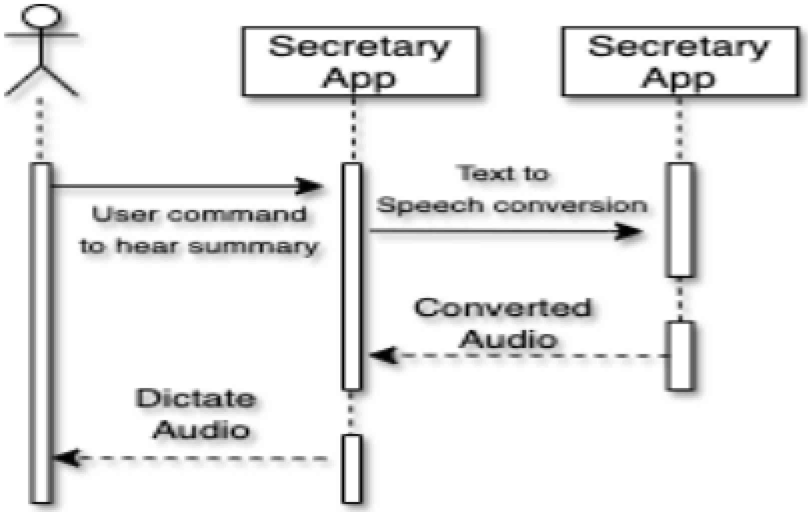
Delete Note

Sequence Diagram

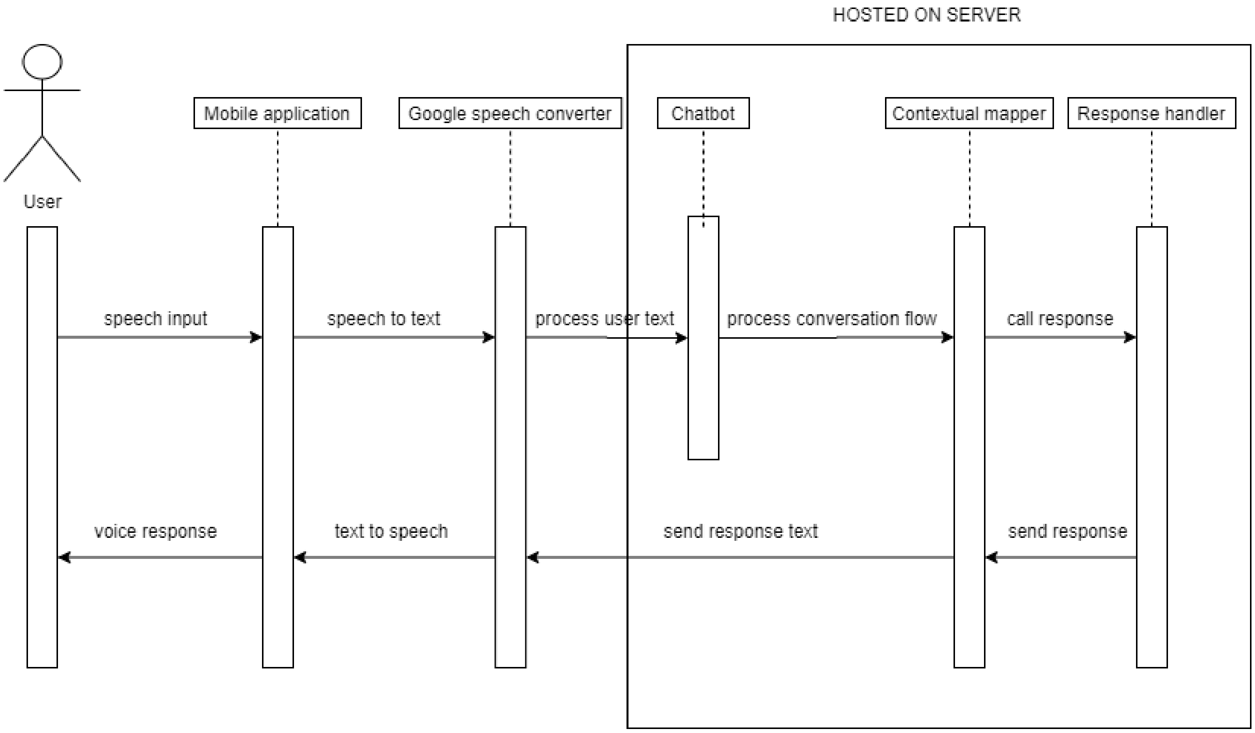


Text to Speech converter

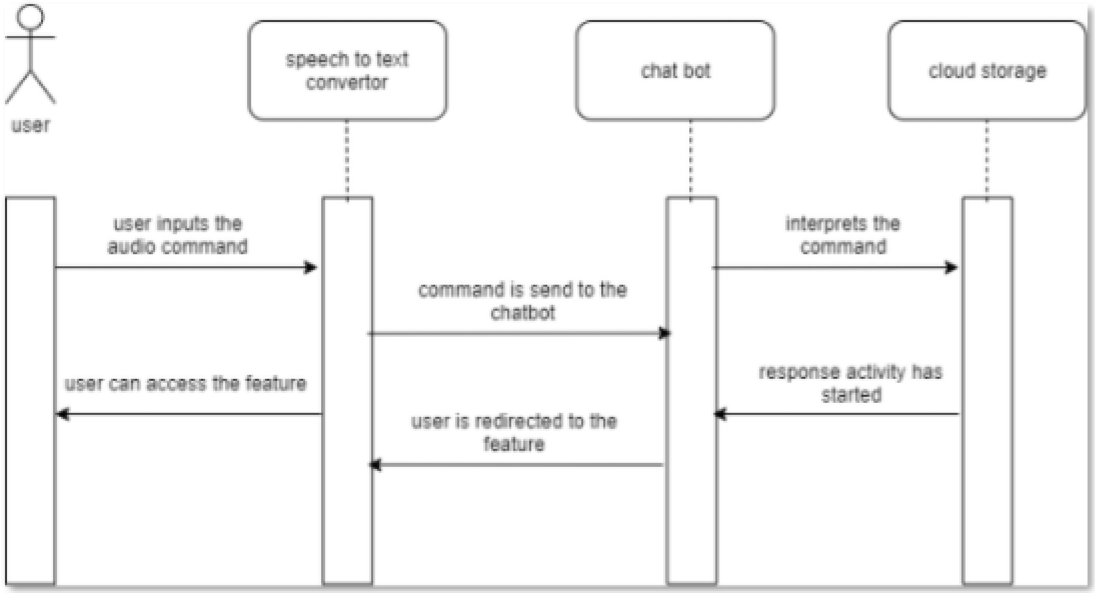
Sequence Diagram



- Chatbot

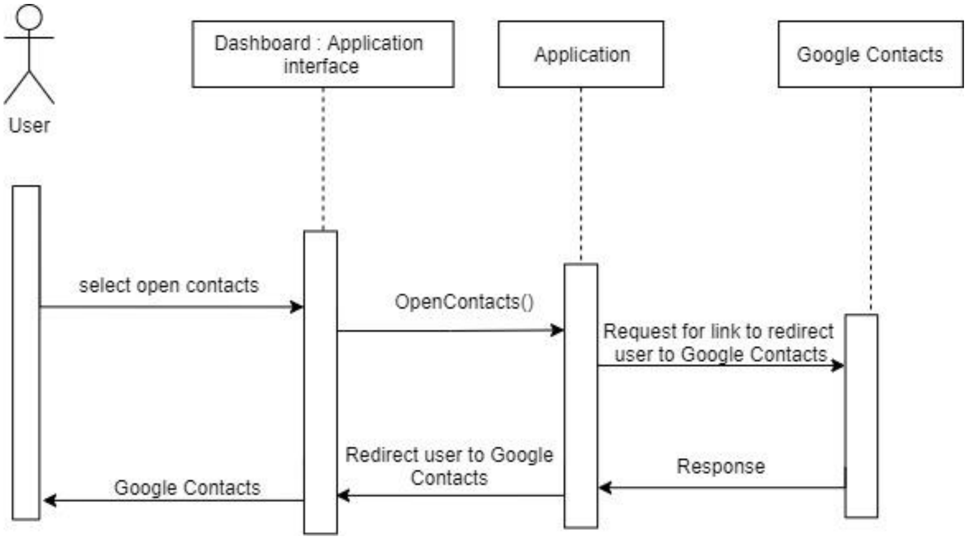


Redirect to appropriate feature


user
speech to text convertor
chat bot
cloud storage
user inputs the audio command
command is send to the chatbot
interprets the command
user can access the feature
response activity has started
user is redirected to the feature

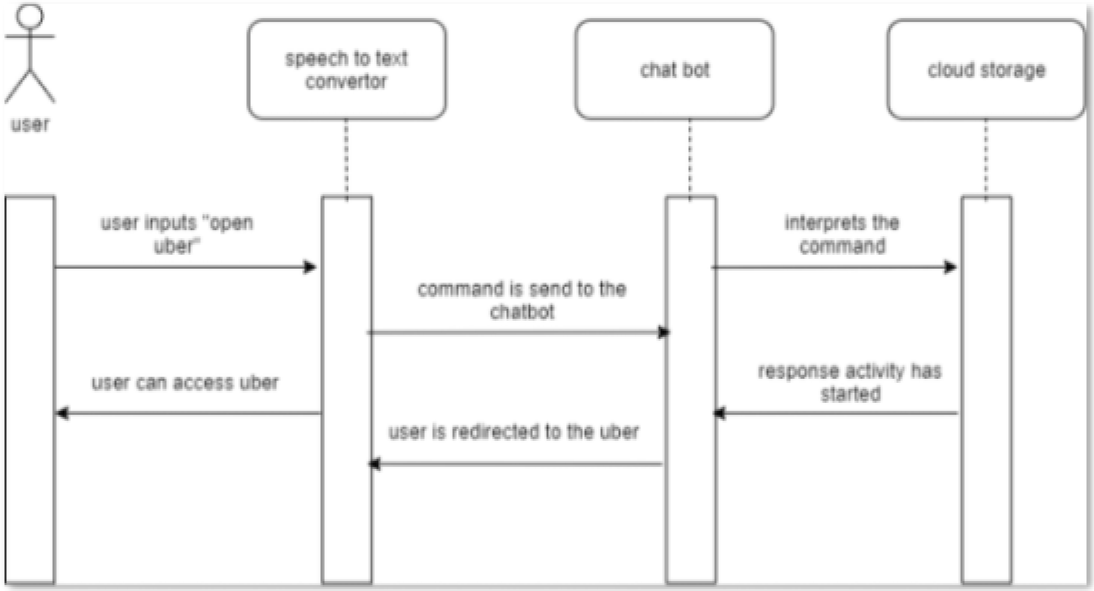
- Contacts

Redirect to Contacts


User
Dashboard : Application interface
Application
Google Contacts
select open contacts
OpenContacts()
Request for link to redirect user to Google Contacts
Response
Redirect user to Google Contacts
Google Contacts

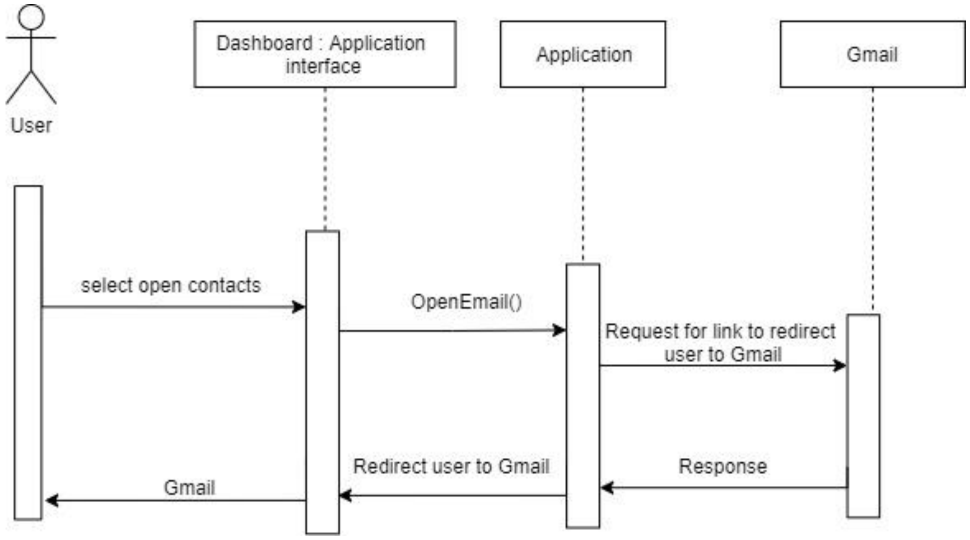
- Uber

Redirect to Uber application
Sequence Diagram

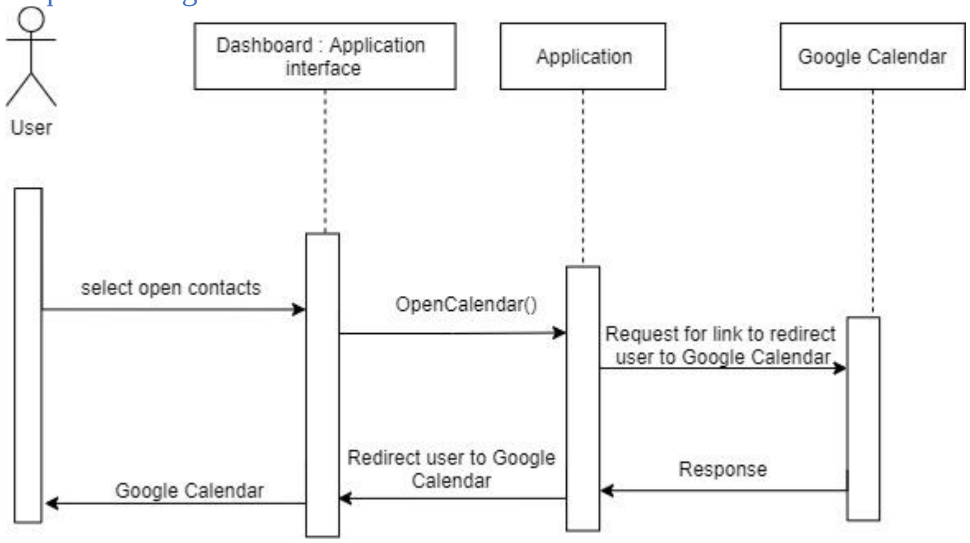


- Email

Redirect to Email
Sequence Diagram



- Calendar
  Redirect to Calendar
  Sequence Diagram


User
Dashboard : Application interface
Application
Google Calendar
select open contacts
OpenCalendar()
Request for link to redirect user to Google Calendar
Response
Redirect user to Google Calendar
Google Calendar

Implementation details

The voice assistant spans over python and android studio. The first task was to handle the speech aspect of the conversation by converting the speech to text of the user's response and paste it into the edit box where the user can change their query accordingly and then converting bot's response in text format to speech using Google's text to speech API in Android Studio Java. The next step was to create the bot using natural language processing techniques to create the training data and later, regression and backpropagation techniques were used to train the model in Python. To host the bot publicly and connect it to Android Studio NGROK services were used. (Gandhi, 2020)

The file system was first made using a node.js server which allows the user to upload files which is then connected to a python flask server which not only reads the documents but also clusters them. The node.js server then renders the cluster into an interactive visual chart that the user can utilize to access a particular document.

The note maker was built mostly on android studio to record the audio and transcribe the speech to text. Then we make use of python to summarize the text, which uses NLTK library and is connected to firebase (cloud storage). Then android studio retrieves the summarized files and displays them for the user on the application.

6. Testing

Testing goals

The goal is to find all errors and loopholes present in the application and to assesses other quality characteristics such as portability, maintainability and usability

Test plan scope

The A.I Office Secretary, Sarah (a mobile application) will be tested by evaluating the overall performance of the APP and by evaluating each system features individually. The app will be tested in normal environments. An android mobile device is needed to carry out the testing. The systems connection with the servers and databases will also be tested during this process.

Test forms and test results

- Testing Environment
  - 1.1. Language Support: English (US)
  - 1.2. Permission
  - 1.3. Voice Assistant – Python 3.7.9
  - 1.4. File System – Python 3.8.3, Node.js v12.18.4
  - 1.5. Integration – Android Studio
- Login

Tested Requirement	Log In (Req. 1.1)
Quick Description	Verifying if the user logs in valid username and password.
Prerequisite	Should be registered as a user
Input	Email ID and password
Expected Output	Access enabled to the user.
Observed Output	Used is enabled access
Verdict	PASS
Comments	The test case returns true which implies that the user was successfully able to log in to the account when the valid details are entered.

Tested Requirement	Log In (Req. 1.2)

Quick Description	This test is to check if system gives access when the wrong credentials are entered.
Prerequisite	Database has a stored username and password.
Input	User enters wrong credentials
Expected Output	Access denied to the user
Observed Output	Used is denied access
Verdict	PASS
Comments	The test cases return false which shows that the user is denied access when wrong credentials were used

Tested Requirement	**Log In (Req. 1.3)**
Quick Description	This test is done to check is the user is granted access when blank values are entered.
Prerequisite	Database has a stored username and password.
Input	User enters blank values
Expected Output	Access denied to the user.
Observed Output	User is denied access
Verdict	PASS
Comments	The test cases return false which shows that the user is denied access when blank entries were submitted

Tested Requirement	**Log In (Req. 1.4)**
Quick Description	This test is to check if a valid user enters a valid username but an incorrect password
Prerequisite	This test is done to check is the user is granted access when the wrong password is entered.
Input	Database has a stored username and password.
Expected Output	User enters an incorrect password
Observed Output	Access denied to the user.
Verdict	User is denied access
Comments	PASS
	The test cases return false which shows that the user is denied access when a wrong password was inputted

- Sign Up

Tested Requirement	Sign Up (Req. 2.1)
Quick Description	User enters all the required values
Prerequisite	none
Input	Enters values for the input filed
Expected Output	User is registered successfully
Observed Output	User is registered with the app
Verdict	PASS
Comments	The test case returns true which shows that the user was able to successfully register with the app.

Tested Requirement	Sign Up (Req. 2.2)
Quick Description	User enters blank values
Prerequisite	none
Input	Enters blank values
Expected Output	User is denied registration
Observed Output	User is not registered with the app
Verdict	PASS
Comments	The test case returns false which shows that the user was unable to successfully register with the app.

Tested Requirement	Sign Up (Req. 2.3)
Quick Description	User enters some required values
Prerequisite	none
Input	Enters values for the input filed
Expected Output	User is unable register successfully
Observed Output	User is not registered with the app
Verdict	PASS
Comments	The test case returns true which shows that the user was unable to successfully register with the app.

Tested Requirement	Sign Up (Req. 2.4)
Quick Description	User enters wrong data in the field
Prerequisite	none
Input	Enters wrong values for the input filed
Expected Output	User is not registered successfully
Observed Output	User is not registered with the app

Verdict	PASS
Comments	The test case returns true which shows that the user was unable to successfully register with the app.

- Permission

Tested Requirement	Permission (Req. 3.1)
Quick Description	Permission requests send to the user
Prerequisite	none
Input	User can either accept or deny the request
Expected Output	User can view the permission dialog message
Observed Output	The permission dialog pops up on user's screen
Verdict	PASS
Comments	The test case returns true which shows that the user was able to see the permission request

Tested Requirement	Permission (Req. 3.2)
Quick Description	User denies permission
Prerequisite	none
Input	User can deny the permission
Expected Output	User is unable to the use the feature as the permission is denied
Observed Output	User cannot use system features
Verdict	PASS
Comments	The test case returns true which shows that the user was unable to use the function as the permission request

Tested Requirement	Log In (Req. 3.3)
Quick Description	User accepts permission
Prerequisite	none
Input	User grants permission
Expected Output	User can use the function successfully
Observed Output	User can use system features
Verdict	PASS
Comments	The test case returns true which shows that the user was able to use the functions

- Voice Assistant

1.6. Greetings

Tested Requirement	**Voice Assistant (Req. 6.1)**
Quick Description	Understand that the user is greeting the bot and reply with a greeting.
Prerequisite	Greetings keywords in user's text input.
Input	User greeting text input query
Expected Output	Understand greeting text input and reply with a greeting.
Observed Output	Replies with a greeting.
Verdict	PASS
Comments	Uses NLP and NLU to understand the user input.

1.7. Assistance

Tested Requirement	**Voice Assistant (Req. 6.2)**
Quick Description	Understand that the user is asking for how the bot can be of assistance and reply with all the features.
Prerequisite	Assistance keywords in user's text input.
Input	User assistance text input query
Expected Output	Reply will all the features of the bot.
Observed Output	Replies with all the features that the bot can assist the user with.
Verdict	PASS
Comments	Uses NLP and NLU to understand the user input

1.8. File Search

Tested Requirement	**Voice Assistant (Req. 6.3)**
Quick Description	Understand that the user is trying to search for a file.
Prerequisite	• File searching keywords in user's text input. • File searching feature to be implemented and integrated on Android Studio.
Input	User file searching text input query
Expected Output	Initiate the file searching feature.
Observed Output	Initiates the file searching feature.
Verdict	PASS

Comments	Uses NLP and NLU to understand the user input

1.9. Recorder

Tested Requirement	**Voice Assistant (Req. 6.4)**
Quick Description	Understand that the user is trying to ask the weather.
Prerequisite	Weather keywords in user’s text input.
Input	Weather input query.
Expected Output	Reply what the weather is like today.
Observed Output	Replies how the weather is today.
Verdict	PASS
Comments	Uses NLP and NLU to understand the user input

1.10. Summarize

Tested Requirement	**Voice Assistant (Req. 6.5)**
Quick Description	Understand that the user is trying to summarize.
Prerequisite	• Summarizing keywords in user’s text input. • Recording permissions on the phone must be allowed. • Storage permissions.
Input	User file recording text input query
Expected Output	Initiate the summarizer.
Observed Output	Initiates the summarizer
Verdict	PASS
Comments	Uses NLP and NLU to understand the user input

1.11. Uber

Tested Requirement	**Voice Assistant (Req. 6.6)**
Quick Description	Understand that the user is trying to book a taxi.
Prerequisite	• Taxi keywords in user’s text input. • Uber must be pre-installed on the phone.
Input	User taxi query.
Expected Output	Initiate the Uber application.
Observed Output	Initiates the Uber application.
Verdict	PASS

Comments	Uses NLP and NLU to understand the user input

1.12. Calendar

Tested Requirement	**Voice Assistant (Req. 6.7)**
Quick Description	Understand that the user is trying to view the calendar
Prerequisite	• Calendar keywords in user query. • Must have calendar permissions to initialize the calendar
Input	User calendar query.
Expected Output	Initialize calendar.
Observed Output	Initializes the calendar.
Verdict	PASS
Comments	Uses NLP and NLU to understand the user input

1.13. Google

Tested Requirement	**Voice Assistant (Req. 6.8)**
Quick Description	Understand that the user is trying to search on google.
Prerequisite	Google search keywords in user’s text input.
Input	User searches google query.
Expected Output	Search on google.
Observed Output	Searches on google depending on user query.
Verdict	PASS
Comments	Uses NLP and NLU to understand the user input

- Contacts

Tested Requirement	**Contact (Req. 7.1)**
Quick Description	User can access contacts through the app when the app is present on the phone
Prerequisite	none
Input	Clicks on the contact button
Expected Output	User is redirected to the contact page
Observed Output	User is navigated to the contact app
Verdict	PASS

Comments	The test case returns true as the user is redirected the contact page and they can view and edit their contacts

Tested Requirement	Contact (Req. 7.2)
Quick Description	User cannot access contacts through the app when the app is not present on the phone
Prerequisite	none
Input	Clicks on the contact button
Expected Output	User is redirected to play store to download the application and install it
Observed Output	User is asked to download the app
Verdict	PASS
Comments	The test case returns true as the user is redirected the contact app on playstore and they can install it.

- File System

1.14. File Upload

Tested Requirement	File upload (Req. 8.1)
Quick Description	This test case is to check if the system can successfully upload a document (.PDF, .DOCX, .TXT).
Prerequisite	none
Input	A word documents.
Expected output	File uploaded.
Observed output	File uploaded
Verdict	Pass
Comments	none

1.15. File Download

Tested Requirement	File download (Req. 8.2)
Quick Description	This test case is to check if the system can successfully download a document (.PDF, .DOCX, .TXT).
Prerequisite	Document must exist on the file server.
Input	press on the name of the document of choice.
Expected output	File downloaded.
Observed output	File downloaded.
Verdict	Pass
Comments	none

1.16. Delete File

Tested Requirement	Delete file (Req. 8.3)
Quick Description	This test case is to check if the system can successfully delete a document (.PDF, .DOCX, .TXT).
Prerequisite	Document must exist on the file server.
Input	A word documents.
Expected output	File deleted.
Observed output	File deleted
Verdict	Pass
Comments	none

1.17. Search “Hagrid”

Tested Requirement	Search “hagrid” (Req. 8.4)
Quick Description	This test case is to check if the system can successfully retrieve documents containing the word “hagrid”.
Prerequisite	none
Input	Type the word “hagrid” on the search results.
Expected output	Retrieves a visual cluster all documents containing “hagrid”.
Observed output	Retrieves a visual cluster all documents containing “hagrid”.
Verdict	Pass
Comments	none

1.18. Search for documents containing the word “Android”

Tested Requirement	Search “android” (Req. 8.5)
Quick Description	This test case is to check if the system can successfully retrieve documents containing the word “android”.
Prerequisite	none
Input	Type the word “android” on the search results.
Expected output	Retrieves a visual cluster all documents containing “android”.
Observed output	Retrieves a visual cluster all documents containing “android”
Verdict	Pass
Comments	none

1.19. Search for documents containing the word “aladdin”

Tested Requirement	Search “aladdin” (Req. 8.6)
Quick Description	This test case is to check if the system can successfully retrieve documents containing the word “aladdin”.
Prerequisite	none
Input	Type the word “aladdin” on the search bar.
Expected output	Retrieves a visual cluster all documents containing “aladdin”.
Observed output	Retrieves a visual cluster all documents containing “aladdin”.
Verdict	Pass
Comments	none

1.20. Search for documents containing the word “carlisle”

Tested Requirement	Search “carlisle” (Req. 8.7)
Quick Description	This test case is to check if the system can successfully retrieve documents containing the word “carlisle”.
Prerequisite	none
Input	Type the word “carlisle” on the search bar.
Expected output	Retrieves a visual cluster all documents containing “carlisle”.
Observed output	Retrieves a visual cluster all documents containing “carlisle”.
Verdict	Pass
Comments	none

1.21. Search for documents containing the word “Hermione”

Tested Requirement	Search “Hermione” (Req. 8.9)
Quick Description	This test case is to check if the system can successfully retrieve documents containing the word “Hermione”.
Prerequisite	none
Input	Type the word “Hermione” on the search bar.
Expected output	Retrieves a visual cluster all documents containing “Hermione”.
Observed output	Does not retrieve a visual cluster all documents containing “Hermione”.
Verdict	fail

Comments	The application fails to retrieve the cluster that matches the searched word simply since the application cross-checks documents containing the words which have the same TF-IDF Score because every word from the document is tokenized and converted to lower case to avoid redundancy. So since there are no capital letters in any of the tokens, the program is unable to match the searched word with any document.

- Summary

1.22. Record Audio

Tested Requirement	**Summary note taking (Req. 9.1)**
Quick Description	The user records the audio of meeting / discussion through our recorder.
Prerequisite	• Recording keywords in user's text input. • Recording permissions on the phone must be allowed.
Input	User file recording text input query
Expected Output	Recorded audio
Observed Output	Recorded audio
Verdict	Pass
Comments	Uses NLP and NLU to understand the user input

1.23. Realtime speech to text

Tested Requirement	**Summary note taking (Req. 9.1)**
Quick Description	Converts the user's audio to text in Realtime as the user speaks using google API
Prerequisite	Can't use the mic at the same time for recorder
Input	Voice input from user
Expected Output	The speech input will be streamed to a server, on the server voice will be converted to text and finally text will be sent back to our app. Transcribed text from the voice input
Observed Output	Realtime speech to text conversion
Verdict	Pass

Comments	Difficult to achieve this with the recorder working at the same time.

1.1. Speech saved on firebase

Tested Requirement	**Summary note taking (Req. 9.1)**
Quick Description	The recorder speech saved on firebase when user finishes their meeting as clicked on the stop button
Prerequisite	The user already starts the recording
Input	Voice input from user
Expected Output	Audio file gets saved automatically with timestamp on firebase
Observed Output	Not required so discarded
Verdict	Pass
Comments	Old files were getting replaced and couldn’t save the correct audio format of the file.

1.2. Transcript saved on firebase

Tested Requirement	**Summary note taking (Req. 9.1)**
Quick Description	The audio transcript can be editable and gets saved in firebase.
Prerequisite	User already needs to have the audio transcript.
Input	User clicks on save transcript button
Expected Output	Saved to firebase with timestamp in correct file format
Observed Output	Transcript saved on firebase
Verdict	Pass
Comments	Old files were getting replaced and couldn’t save the correct text format of the file. Initially file contained junk values.

1.3. Empty Transcript

Tested Requirement	**Summary note taking (Req. 9.1)**
Quick Description	Checking to see whether text file gets saved if empty
Prerequisite	none
Input	User speech to text
Expected Output	Error validation saying no text
Observed Output	
Verdict	
Comments	

1.4. Edit Transcript

Tested Requirement	**Summary note taking (Req. 9.1)**

Quick Description	Checking to see whether user can edit text file before saving
Prerequisite	Needs to be some text to edit
Input	User text correction/ customization
Expected Output	Text is editable
Observed Output	Transcript is editable
Verdict	Pass
Comments	User can correct the transcript before summarizing if some words are not captured accurately by speech to text converter.

1.5. Recording retrieved from firebase to play

Tested Requirement	**Summary note taking (Req. 9.1)**
Quick Description	Audio recordings played in Recycler view from firebase retrieval
Prerequisite	Needs to have prior recording saved in order to retrieve it
Input	User clicks on stop recording button
Expected Output	Retrieve files from firebase directly so that our recycler view can be dynamically linked with firebase. E.g.: if file deleted there, then deleted in the app view too.
Observed Output	Used alternative to retrieve files from device internal storage to play recordings on recycler view.
Verdict	Fail
Comments	Required to use a FirebaseRecyclerAdapter since I had more than one modal class from fragments.

1.6. Summarizer module in python

Tested Requirement	**Summary note taking (Req. 9.1)**
Quick Description	Text from recording gets summarized into bullet points
Prerequisite	Recording needs to be converted to text
Input	Recorded audio transcript
Expected Output	Summarized text into bullet points.
Observed Output	Summarizes the transcript
Verdict	Pass
Comments	None

- Uber

Tested Requirement	**Uber (Req. 10.1)**

Quick Description	User can access Uber through the app when the app is present on the phone
Prerequisite	none
Input	Clicks on the RideWithUber button
Expected Output	User is redirected to Uber
Observed Output	Returns true
Verdict	pass
Comments	The test case returns true as the user is redirected Uber and they can fix rides

Tested Requirement	Uber (Req. 10.2)
Quick Description	User cannot access contacts through the app when the app is not present on the phone
Prerequisite	none
Input	Clicks on the RideWithUber button
Expected Output	User is redirected to play store to download the application and install it
Observed Output	Returns true
Verdict	pass
Comments	The test case returns true as the user is redirected the Uber app on playstore, and they can install it

- Calendar

Tested Requirement	Calendar (Req. 11.1)
Quick Description	User can access Calendar through the app when the app is present on the phone
Prerequisite	none
Input	Clicks on the Calendar button
Expected Output	User is redirected to Calendar
Observed Output	Returns true
Verdict	pass
Comments	The test case returns true as the user is redirected Calendar and they can create events, view events, edit events and set reminders.

Tested Requirement	Calendar (Req. 11.2)
Quick Description	User cannot access Calendar through the app when the app is not present on the phone
Prerequisite	none
Input	Clicks on the Calendar button

Expected Output	User is redirected to play store to download the application and install it
Observed Output	Returns true
Verdict	pass
Comments	The test case returns true as the user is redirected the Calendar app on play store and they can install it

- Email

Tested Requirement	**Email (Req. 12.1)**
Quick Description	User can access Email through the app when the app is present on the phone
Prerequisite	none
Input	Clicks on the Mail button
Expected Output	User is redirected to Gmail
Observed Output	User is navigated to the GMAL app
Verdict	PASS
Comments	The test case returns true as the user is redirected Gmail and they can send and view emails, categorize and delete emails

Tested Requirement	**Email (Req. 12.2)**
Quick Description	User cannot access Email through the app when the app is not installed on the phone
Prerequisite	none
Input	Clicks on the Mail button
Expected Output	User is redirected to play store to download the application and install it
Observed Output	User is asked to download the app
Verdict	PASS
Comments	The test case returns true as the user is redirected the Gmail app on playstore, and they can install it

7. Conclusion and future work

Conclusion

In conclusion, Sarah has been developed after thorough research and experimentation. The results shown above are based on accurate dataset. It has been tested with new different users and conditions. The application has various features and services that aid in increasing productivity while doing tasks and responsibilities. The app was build following all usability guidelines and principles. The user data will be gathered and analyzed to further enhance and optimize the product. The application does not intend to replace the existing A.I voice assistants but to enhance the existing solutions and tailor it for the office environment. With this application now users have the most important tools such as a file system with cluster visualization, chatbot designed for workplaces, note maker to summarize meetings and various services like Uber for rides, Calendar, Contacts and Emails for an office environment, all in one platform reducing cost, saving time while increasing quality and productivity.

Strengths and weaknesses

Strengths:

- Our application utilizes a voice assistant to access many of its features which are inaccessible by other voice assistants which not only brings added value to our product but also makes our product easier to use and navigate across.
- Our application’s file system utilizes a cluster-based approach which enables the user to search for documents based on its content unlike other file managing application which only permits the user to search based on the name of the file itself.
- Our application’s note taking feature enables the user to record the meeting and have the application provide summarized bullet points of the meeting back to the user. Unlike other note taking applications in which the user notes down what he/she believes is important. As a result, we can see that our application in a nutshell takes notes for the user instead of the user having to do it manually.

Weakness:

- Since we have implemented a web-service architecture, setting up each of the components of the application can be a rather long and a tedious task.
- Our application’s voice assistant is limited in terms of mainstream functionalities that other voice-assistants such as Google, Siri and Alexa possess.
- The file system of our application may not always be accurate in terms of search results simply because the application cross-checks documents containing the words which have the same TF-IDF Score as the word searched instead of utilizing a string-matching technique.

Future improvements

- Add a meeting feature which allows the user to manage meetings in a much easier fashion and prevent them from overlapping.

- Add more mainstream functionalities for the chatbot such as being able to ask it questions such as “who is Elon Musk?”.
- Add a feature which sends notifications to the user saying that he/she is late for said meetings.
- Add a feature where the application can access the user’s contacts and book a meeting with said contact using the chatbot and meetings feature.
- Add a feature which sends notifications to the user saying that he/she has received when a request for a meeting has either been denied or confirmed.
- Modularity in microservices.

8. References

- 99firms.com. 2019, “How Many Email Users Are There?” , 99firms.com, viewed on 13th february 2020, <https://99firms.com/blog/how-many-email-users-are-there/#gref>
- AEO 2019, “DISADVANTAGES OF GOOGLE ASSISTANT” Answer Engine Optimisation, viewed on 14th February 2020, https://www.answer-engine-optimisation.com/aeo-blog/disadvantages-of-google-assistant
- Bera, A. 2019, “Siri Statistics 2019 - Is Siri Better Than Google?” , SafeAtLast.co., Viewed on 13 February 2020, < https://safeatlast.co/blog/siri-statistics/#gref>.
- Bhatia S. & Bharambe A. 2009, “Architecture for Mobile Based Intelligent Personal Assistant”, Research Gate, viewed on 9th May 2020, <https://www.researchgate.net/publication/291074826_Architecture_for_Mobile_Based_Intelligent_Personal_Assistant>.
- Delac G., Budiselic I., Zuzak I. 2011, “A FINITE-STATE MACHINE APPROACH FOR MODELING AND ANALYZING RESTFUL SYSTEMS”, Journal of Web Engineering, vol. 10, no. 4, pp. 353-390.
- Jansen M. 2019, “How to use Samsung Bixby: Everything you need to know”, Digital Trends, Viewed on 14th February 2020, <https://www.digitaltrends.com/mobile/how-to-use-bixby/>
- Likas, A., Vlassis, N. and J. Verbeek, J. 2003. “The global k-means clustering algorithm”, Pattern Recognition, vol. 36, no.2, pp.451–461.
- Meet Sally 2017, “AI Personal Assistant vs Secretary: 5 Key Differences” viewed on 14th February 2020, https://medium.com/@meetsally/ai-personal-assistant-vs-secretary-5-key-differences-569d65b8f570
- Nanalyze 2017, “How AI Personal Assistants will fill secretarial duties ”, viewed on 14th February 2020, <https://www.nanalyze.com/2017/03/ai-personal-assistants-secretary-duties/>
- O’Boyle B. & Tillman M. 2019, “What is Google Assistant and what can it do?”, Pocket-lint, Viewed on 14th February 2020, https://www.pocket-lint.com/apps/news/google/137722-what-is-google-assistant-how-does-it-work-and-which-devices-offer-it

- O'Boyle B. 2019, "What is Siri and how does Siri work?", Pocket-lint, Viewed on 14th February 2020, https://www.pocket-lint.com/apps/news/apple/112346-what-is-siri-apple-s-personal-voice-assistant-explained
- Ortinaou, D., Dunn, C., Schonning, N. and Opgenorth, T., 2018. Xamarin Android Application Fundamentals, Microsoft, viewed on 9th May 2020, https://docs.microsoft.com/en-us/xamarin/android/app-fundamentals/permissions-images/02-permissions-workflow.png.
- Ortinaou, D., Dunn, C., Schonning, N. and Opgenorth, T., 2018. Xamarin Android Application Fundamentals. [online] Microsoft. Available at: <https://docs.microsoft.com/en-us/xamarin/android/app-fundamentals/permissions-images/02-permissions-workflow.png> [Accessed 9 May 2020].
- Ratnesh 2018, "Amazon Alexa – Pros, Cons and Competitors", SeerOmega, viewed on 14th February 2020, <https://seeromega.com/amazon-alexa-pros-cons-competitors/>
- Rodriguez J. 2017, "Hello Bixby: Strengths and Weaknesses of Samsung New Digital Assistant Strategy", Medium, viewed on 14th February 2020, https://medium.com/@jrodthoughts/hello-bixby-strengths-and-weaknesses-of-samsung-new-digital-assistant-strategy-84855043cb3
- Statista. 2020 "Smartphone users worldwide 2020" , Statista, viewed on 13th february 2020, https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/
- Statista. 2020, "Number of e-mail users worldwide 2023" , Statista, viewed on 13th february 2020, https://www.statista.com/statistics/255080/number-of-e-mail-users-worldwide/
- Wetzel K. & Rawes E. 2020, "What exactly is Alexa? Where does she come from? How does she work?", Digital Trends, viewed on 14th February 2020, <https://www.digitaltrends.com/home/what-is-amazons-alexa-and-what-can-it-do/>
- Zujic B. 2020, "The Advantages and Disadvantages of Siri", Technobezz, viewed on 14th February 2020, <https://www.technobezz.com/advantages-and-disadvantages-of-siri/>
- Gandhi, D. '*Rasa Chatbot + Android Covid App Tutorial: Part 1*.' Medium. viewed on 2 December 2020 <https://medium.com/@dishant_gandhi/rasa-chatbot-android-covid-app-tutorial-part-1-1010f667c06c>
- Battoo, S. (2019). *Uploading Files to MongoDB with GridFS and Multer Using NodeJS*. [online] DEV Community. Available at: https://dev.to/shubhambattoo/uploading-files-to-mongodb-with-gridfs-and-multer-using-nodejs-5aed.
- Davydova, O. (2019). *Text Preprocessing in Python: Steps, Tools, and Examples*. [online] Data Monsters. Available at: https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908 [Accessed 16 Oct. 2018].
- Open Source Automation. (2019). *How to read Word documents with Python*. [online] Available at: http://theautomatic.net/2019/10/14/how-to-read-word-documents-with-python/ [Accessed 2 Dec. 2020].

- Pathak, A. (n.d.). *Text Data Clustering*. [online] kaggle.com. Available at: https://www.kaggle.com/akshatpathak/text-data-clustering [Accessed 2 Dec. 2020].
- Qaiser, R. (2020). *How to Extract Words From PDFs With Python*. [online] Medium. Available at: https://medium.com/better-programming/how-to-convert-pdfs-into-searchable-key-words-with-python-85aab86c544f [Accessed 2 Dec. 2020].
- (Building a text summarizer in Python using NLTK and scikit-learn class TfidfVectorizer, 2020) https://medium.com/saturdays-ai/building-a-text-summarizer-in-python-using-nltk-and-scikit-learn-class-tfidfvectorizer-2207c4235548