

Submitted by

 Shubham Kumar(13000118046)

 Simran Mittal( 13000118045)

 Madhusree Bera( 13000118092)

 Bishyan Kar( 13000118109)

Submitted for the partial fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering



Techno Main Salt Lake,

EM 4/1, Salt Lake, Sector V, Kolkata – 700 091

**CERTIFICATE**

This is to certify that the project entitled “Harmonize Space” prepared by Shubham Kumar(13000118046), Simran Mittal(13000118045), Bishyan Kar(13000118109), Madhusree Bera(13000118092) of B.Tech (Computer Science & Engineering), Final Year, has been done according to the regulations of the Degree of Bachelor of Technology in Computer Science & Engineering. The candidates have fulfilled the requirements for the submission of the project report.

It is to be understood that, the undersigned does not necessarily endorse any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which it has been submitted.

----------------------------------------------- -----------------------------------------------

(Signature of the Internal Guide) (Signature of the HOD)

--------------------------------------------------------------------------------------------

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

--------------------------------------------------------------------------------------------

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to our mentor of the department of Computer Science and Engineering, whose role as project guide was invaluable for the project. We are extremely thankful for the keen interest he / she took in advising us, for the books and reference materials provided for the moral support extended to us.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skill that will always remain as our asset and to all non-teaching staffs for the gracious hospitality they offered us.

Place: Techno India, Salt Lake

Date:

Shubham Kumar

………………………………

Bishyan Kar

………………………………

Madhusree Bera

………………………………

Simran Mittal

………………………………

**Contents**

[Introduction](#_gjdgxs) **6**

[Abstract](#_30j0zll) 6

[Problem Domain](#_1fob9te) 6

[Related Studies](#_3znysh7) 6

[Glossary](#_2et92p0) 7

[Problem Definition](#_tyjcwt) **8**

[Scope](#_3dy6vkm) 8

[Exclusions](#_1t3h5sf) 8

[Assumptions](#_4d34og8) 8

[Project Planning](#_2s8eyo1) **8**

[Software Life Cycle Model](#_17dp8vu) 8

[Scheduling](#_3rdcrjn) 8

[Cost Analysis](#_26in1rg) 9

[Requirement Analysis](#_lnxbz9) **10**

[Requirement Matrix](#_35nkun2) 10

[Requirement Elaboration](#_1ksv4uv) 11

[Desktop Audio Transcription (DAT)](#_qt8jibszewu8) 11

[Mobile Audio Transcription (MAT)](#_yu5hb2a66l9) 12

[Meeting Scheduler (MS)](#_s0uqrb4rf4rk) 13

[Web Highlighter (WH)](#_vh9r8hi4hsnt) 13

[Sync Clipboard (SC)](#_lvb4k5sg86vi) 14

[Manage Files (MF)](#_aibdirr9feqm) 14

[Manage Websites (MW)](#_1ggqzx6i6w3h) 15

[User Profile(UP)](#_qd82ia8j8zow) 16

[4.2.9 Harmonize Search (HS)](#_45q178ycdxx4) 16

[Design](#_44sinio) **17**

[Technical Environment](#_2jxsxqh) 17

[Detailed Design](#_z337ya) 17

[Hierarchy of Harmonize Space](#_wzgu9ymea0kv) 18

[Hierarchy of modules for Desktop and Mobile App](#_e2k075t4mya3) 19

[Desktop Audio Transcription (DAT)](#_8r7xczn5wxh) 19

[Mobile Audio Transcription (MAT)](#_uhele0sdj5sc) 21

[Meeting Scheduler (MS)](#_1t4vua6fjgts) 21

[Website Highlighter (WH)](#_hy0byiduvnqk) 22

[Sync Clipboard (SC)](#_hxk6krk216xe) 25

[Manage File (MF)](#_1x3u443uaaav) 25

[Manage Website (MW)](#_c6fckpqp1l7x) 26

[User Profile (UP)](#_tjvg4q6tgtmt) 27

[5.2.11 Harmonize Search (HS)](#_eogsya2piybi) 28

[Implementation](#_3j2qqm3) **29**

[Implementation Details](#_1y810tw) 29

[System Installation Steps](#_4i7ojhp) 34

[System Usage Instructions](#_2xcytpi) 36

[Test Results and Analysis](#_1ci93xb) **36**

[Conclusion](#_3whwml4) **36**

[Project Benefits](#_2bn6wsx) 36

[Future Scope for improvements](#_qsh70q) 37

[**References / Bibliography**](#_t28qhpjyjozx) **38**

# Introduction

## Abstract

Harmonize Space is a utility software that provides various productivity and organizing tools for online meetings and website browsing. This project has 3 major requirements: Application Software, Mobile Application and Browser Extension. This project will be beneficial to anyone who attends online meetings, uses computers and/or mobile phones extensively and wants to save time by keeping things organized.

## Problem Domain

The domains of this project are as follows:

1. **Web application**
2. **Desktop Application**
3. **Android Application**
4. **Browser Extension**
5. **Cloud Computing**

## Related Studies

One of the features provided by our application is "live meeting transcription." The question is why would anyone want to transcribe a recording at all ?

One of the reasons is that transcription enables us to get a verbatim record of what is being said and as such is a more accurate way of representing the interview than taking notes. [1]

CLAYMAN and TEAS GILL, who work within the tradition of conversation analysis, remark

"Transcripts make features of the recording more transparent and accessible, enabling one to 'see' the vocal and non-vocal activities that unfold on the tape. A good transcript helps the analyst to get a purchase on the organization of the interaction, including its fleeting and momentary features. A transcript is not a substitute for the recording, but rather is an essential analytical tool to be used along with the recording" [2]

In recent times, video conferencing apps are becoming a daily need for not only those who are working but for online interviews too. [3] Following is a comparative study of features available in various similar meeting applications and “Harmonize Space”

*Table 1 Comparative Analysis*

| **Features** | **Web Highlighter** | **Transcription** | **Auto Scheduling meeting** | **Record Meeting** | **Sync clipboard** |
| --- | --- | --- | --- | --- | --- |
| LINER | ✔ | x | x | x | x |
| MS Teams | x | ✔ | x | ✔ | x |
| Google Meet | x | ✔ | x | x | x |
| Zoom | x | ✔ | x | ✔ | x |
| Calendly | x | x | ✔ | x | x |
| Proposed system (Harmonize Space) | ✔ | ✔ | ✔ | ✔ | ✔ |

## Glossary

| **Term** | **Meaning** |
| --- | --- |
| Application Software | a computer program designed to help people perform an activity. |
| Mobile Application | A mobile application, most commonly referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone |
| Browser Extension | A browser extension is a small software module for customizing a web browser. |
| Audio Transcription | Audio transcription is the process of converting speech in an audio file into written text. |
| Clipboard | The clipboard is a buffer that some operating systems provide for short-term storage and transfer within and between application programs. |
| Google | A Google Account is required for Gmail, Google Hangouts, Google Meet and Blogger. |
| Zoom | Zoom provides video telephony and online chat services through a cloud-based peer-to-peer software platform and is used for teleconferencing, telecommuting, distance education, and social relations. |

# Problem Definition

## Scope

A utility software for desktop, android phones and web browsers that will provide various tools to make browsing and organizing more effective. The proposed system will have the features of  website liner, audio transcription and organizing, synchronization of copying texts/ files across various devices like laptops, mobile, etc.

## Exclusions

* Recording meeting / calls  in Android OS after Android v10. [4]

## Assumptions

* Computers (laptop/ desktop) have Windows OS version 8 or above
* Mobile Phone with Android version 10 or above

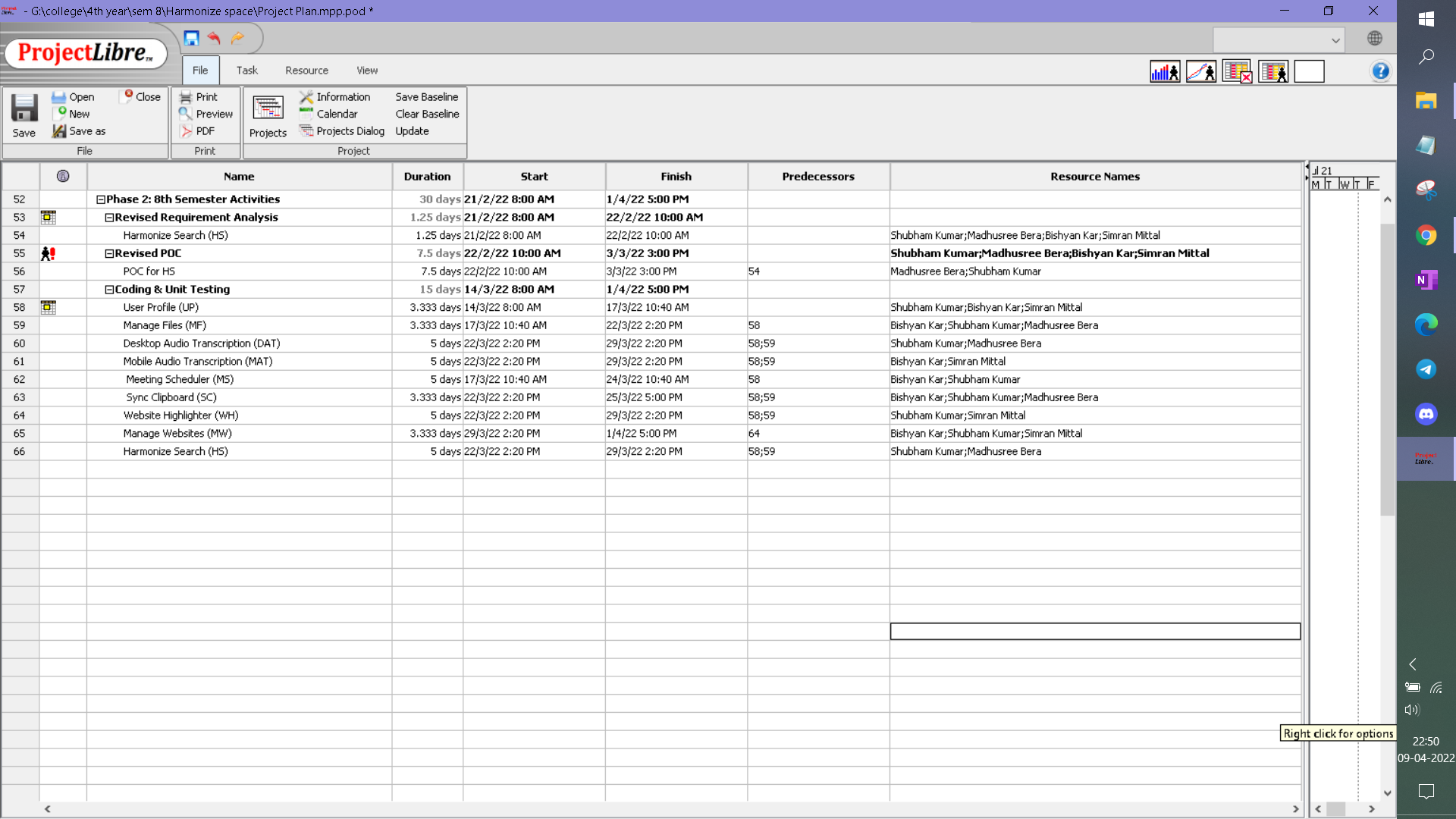
# Project Planning

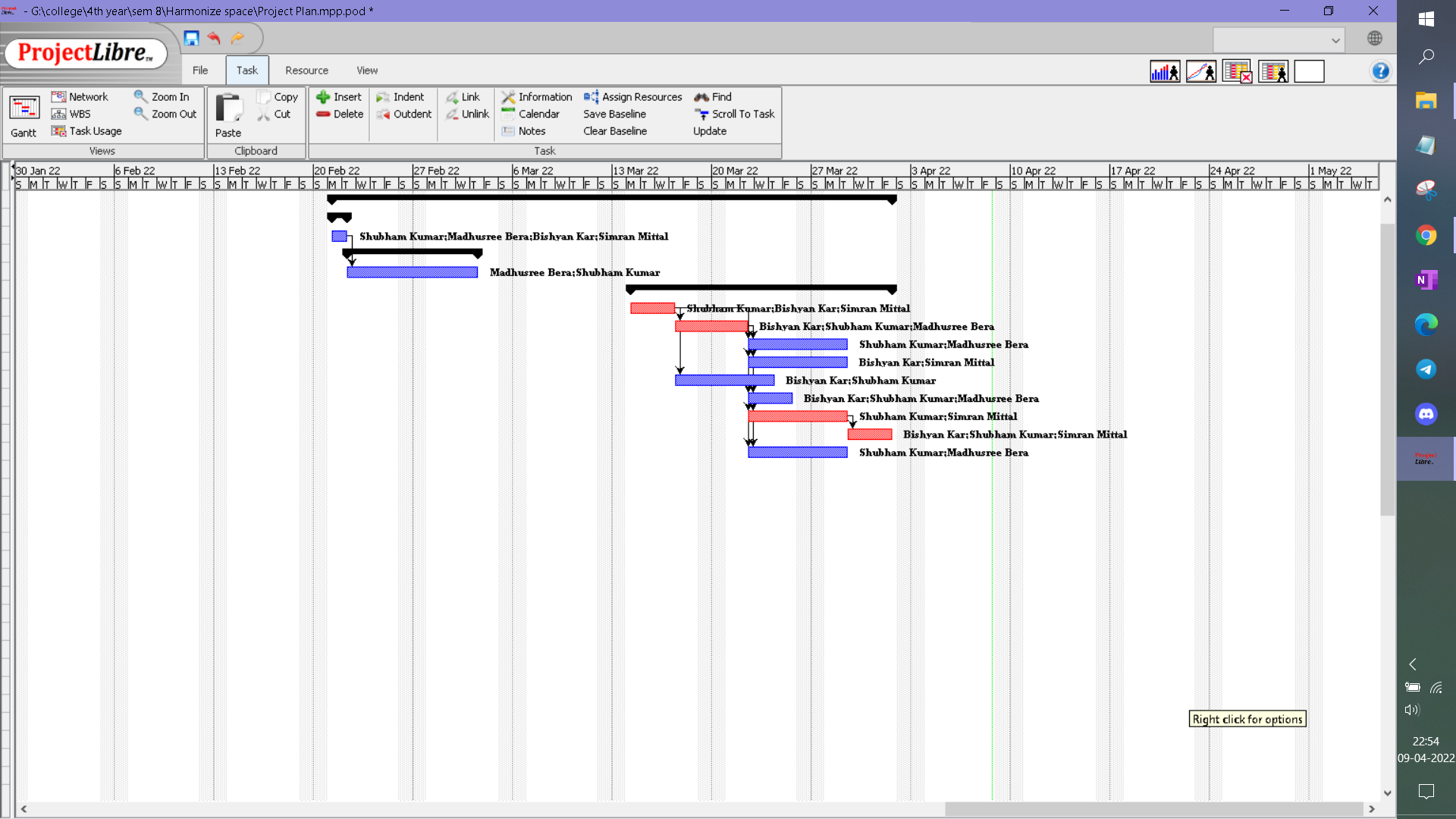
For **Project Planning** / schedule, please paste suitably from MS Project. Gantt charts should be shown for major phases with highlighted milestones.

## Software Life Cycle Model

We have chosen Iterative Waterfall Model as Software Development Life Cycle model

## Scheduling





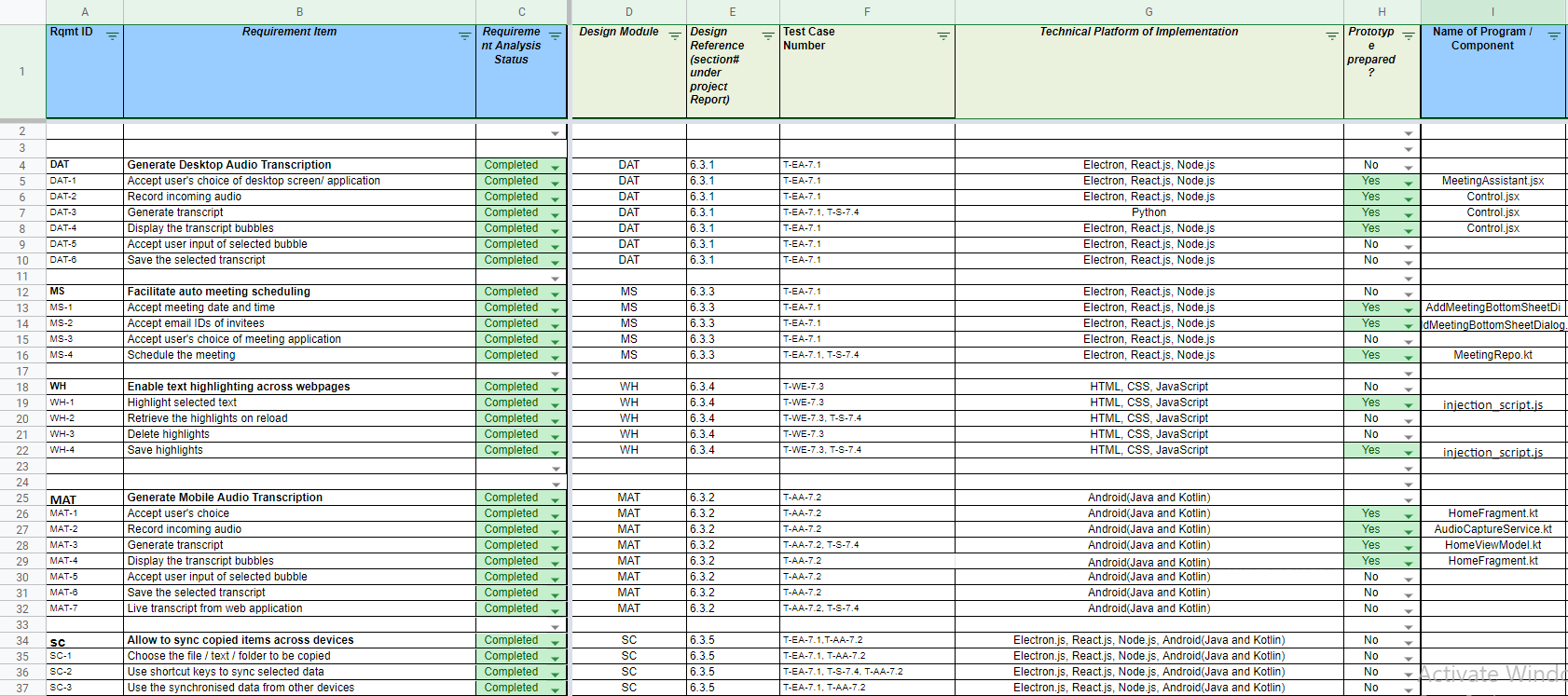
## Cost Analysis

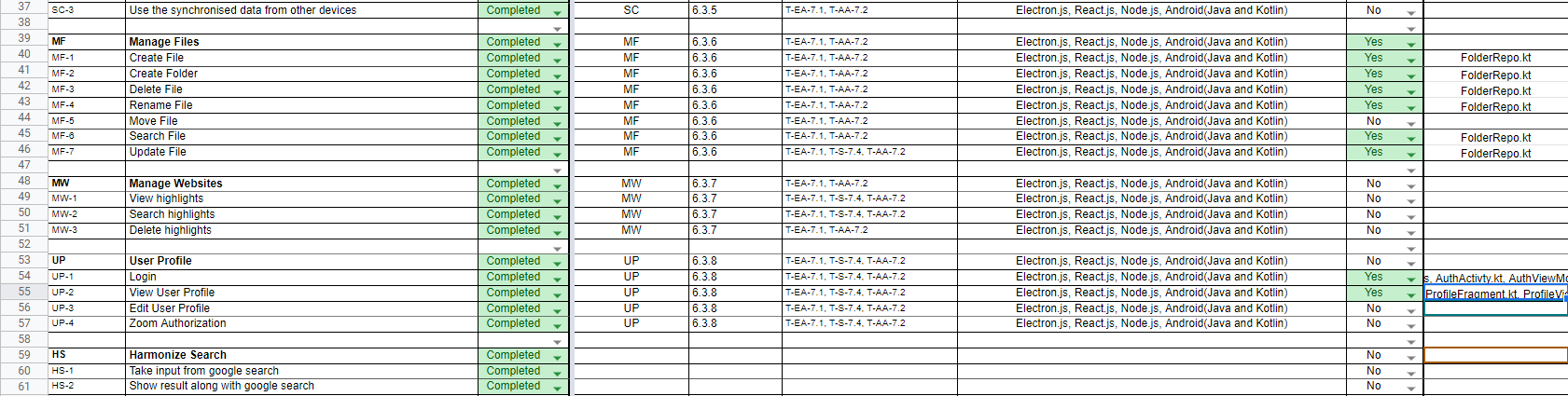
After analyzing our project requirements and outcome, we have come up with the list of expenses

* Premium API : Google Speech-to-Text API - Rs. 0.30 per 15 seconds
* Hosting: Amazon AWS Lambda and EC2 - Rs 1500 per month
* Domain provider - GoDaddy - Rs. 2227

# Requirement Analysis

## Requirement Matrix





## Requirement Elaboration

### Desktop Audio Transcription (DAT)

**4.2.1.1**

**DAT-1: Accept user’s choice of desktop screen/ application:** Here the user is given choices to select the required application for which he wants to record audio and generate transcript

**4.2.1.2**

**DAT-2: Record Incoming audio:** Audio of the selected application is recorded via API. The audio is recorded in packets and prepared to be sent to the desktop Applications

**4.2.1.3**

**DAT-3: Generate transcript:** The recorded audio packets are sent to the desktop application one by one in order. In the application, a request is sent to the server where an algorithm (Python script) is used to generate the transcripts. The transcripts of the audio packets are prepared to be sent to the desktop application which in turn sends the generated transcript to the bubble display window.

**4.2.1.4**

**DAT-4:** **Display the transcript bubbles on the Bubble display window:** The transcripts will appear in the bubble display window and this window will hover on the screen of the user. If the user wants he can resize / minimize the window as per requirement. This window will also contain Pause/ Resume , Stop , Save All, Save Selected buttons which will provide specific functionalities.

**4.2.1.5**

**DAT-5:** **Accept user input of selected bubble:** Once the user stops the recording, the user can save the transcripts and organize them into files and folders. The user can either save all the transcript bubbles or select a few as per the user’s needs. The user can select the bubbles by clicking on them.

**4.2.1.6**

**DAT-6: Save the selected transcript:**  After the bubbles are selected, the user can save the bubbles, give an appropriate name to the file and select the folder under which this file is to be saved.

### Mobile Audio Transcription (MAT)

**4.2.2.1**

**MAT-1: Accept user’s choice**: Here the user is given choices to start/stop a service to record and transcript the audio.

**4.2.2.2**

**MAT-2: Record Incoming audio**: The service should record the audio in packets and prepare to be sent to an API endpoint.

**4.2.2.3**

**MAT-3: Generate transcript**: The recorded audio packets should be sent to an api endpoint one by one in order and the app should store(temporarily) the transcripts, received as a response.

**4.2.2.4**

**MAT-4: Display the transcript bubbles**: App should display the received transcripts in the form of a chat bubble so as to allow the user to view it from anywhere and also take necessary actions.

**4.2.2.5**

**MAT-5: Accept user input of selected bubble**: App should allow the user to interact with the transcript bubble and select one/many items to be saved.

**4.2.2.6**

**MAT-6: Save the selected transcript**: App should allow the user to save a selected transcript(permanently) from the  bubble, as a note.

**4.2.2.7**

**MAT-7: Live transcript from web application**: Alternately the app should be able to show real time transcriptions generated from the web app. This becomes more useful in case of certain restrictions in android.

### Meeting Scheduler (MS)

**4.2.3.1**

**MS-1: Accept meeting date and time:** User should enter the date and time of the meeting he wants to schedule.

**4.2.3.2**

**MS-2: Accept email IDs of invitees:** User should enter the email IDs of the participants of the meeting

**4.2.3.3**

**MS-3: Accept user’s choice of meeting application:** A drop down menu will be available containing the list of meeting applications available. The user can choose the meeting application that he prefers.

**4.2.3.4**

**MS-4: Schedule the meeting:**  The application will schedule a meeting according to the user’s given choice

### Web Highlighter (WH)

**4.2.4.1**

**WH-1: Highlight selected text:**  Enable user to select any text on web pages and use the web highlighter to highlight the text

**4.2.4.2**

**WH-2: Retrieve the highlights on reload:** Once a user highlights any text, the highlighted text must be visible when the user visits the same web page next time.

**4.2.4.3**

**WH-3: Delete highlights:** The user can delete highlighting of any text on any web page.

**4.2.4.4**

**WH-4: Save highlights:** All the text highlighted by the user must persist even after the user closes his web browser. In other words, the highlighted text must be saved so that the data persists.

**4.2.4.5**

**WH-4: Search saved highlights:** The user must be able to search his saved highlights using the search functionality provided by Harmonize Space

### Sync Clipboard (SC)

**4.2.5.1**

**SC-1: Choose the file / text / folder to be copied:**  The user must choose the file / text / folder which he wants to sync across all his devices

**4.2.5.2**

**SC-2: Use shortcut keys to sync selected data:** The sync clipboard functionality can be used in 2 ways. The user can use keyboard shortcut Ctrl + M. For the mobile application, select the option to sync across devices.

**4.2.5.3**

**SC-3: Use the synchronized data from other devices:** Once the data is synced across other devices where the user is using Harmonize Space, the user will receive the message “data synced successfully”. In case there is failure due to connectivity issues or any other reason, the user will be notified about data synchronization failure. Once the data is synced successfully, it can be used from the other device.

### Manage Files (MF)

**4.2.6.1**

**MF-1: Create file:** User can create and store files of any type (doc, image, text, audio, video)

**4.2.6.2**

**MF-2: Create folder:** User can create folder as required to organize their files

**4.2.6.3**

**MF-3: Delete file:**  User can delete a file when needed

**4.2.6.4**

**MF-4: Rename file:** User can a rename file as and when required

**4.2.6.5**

**MF-5: Move file:** User can move a file from one folder to another as per requirement

**4.2.6.6**

**MF-6: Search file:** User can search files created by him by using file name or any other search keyword

**4.2.6.7**

**MF-7: Sort files:** User can view the files in sorted order (by name / date created/ last modified). By default the system will show the files according to the “last modified” file.

**4.2.6.8**

**MF-8: Update file:** The user can update existing files and save the changes

### Manage Websites (MW)

**4.2.7.1**

**MW-1: View highlights:** The user can view all the highlighted texts in each website arranged in files. The files are sorted according to “last modified”

**4.2.7.2**

**MW-2: Search highlights:** The user can search the saved highlights using search keywords/ website name

**4.2.7.3**

**MW-3 Delete highlights:** The user can delete highlights from the file of the particular website

### User Profile(UP)

**4.2.8.1**

**UP-1: Login:** To use services provided by Harmonize Space, one needs to login to the application using a Google account.

**4.2.8.2**

**UP-2: View User Profile:** Here the user can view his profile, his saved information which are:

name

* Google login ID
* Zoom login ID
* drive space utilized by Harmonize Space
* data statistics
* other connected devices

**4.2.8.3**

**UP-3: Edit User Profile:** Here the user can add or remove devices.

**4.2.8.4**

**UP-4: Zoom Authentication:** Asking user to give access to Zoom account so that he can schedule Zoom meetings using Harmonize Space.

### 4.2.9 Harmonize Search (HS)

**4.2.9.1**

**HS-1: Take input from google search**: Harmonize search is a web extension that will take input when the user types search keyword in search box of web browser

**4.2.9.2**

**HS-2: Show result along with google search:** The extension will show the search result in the web browser after searching the user’s files saved in Harmonize Space across all devices

# Design

## Technical Environment

**Hardware:**

Min 4GB RAM

Min i3 processor

Min free space 150 MB

**OS:**

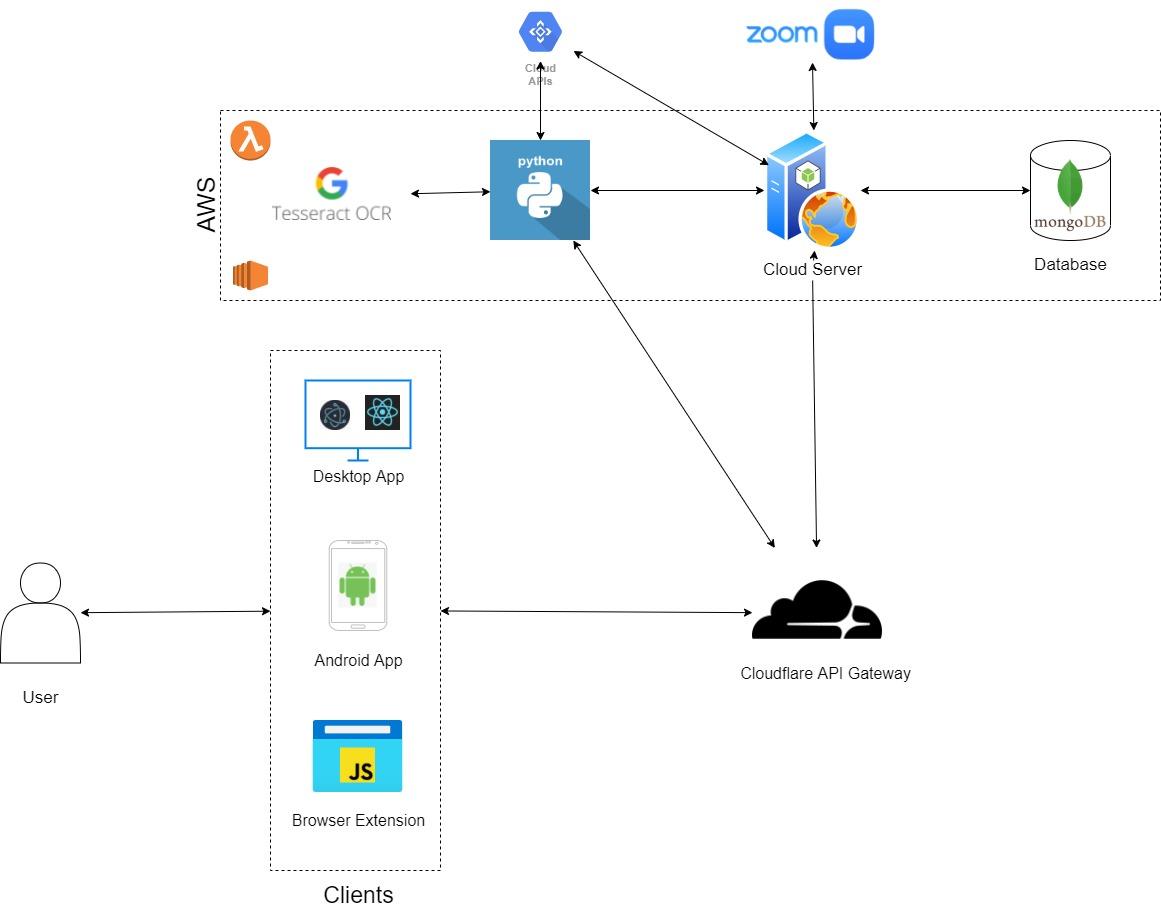
Windows, Android

**Software:**

Clipboard, Web Browser

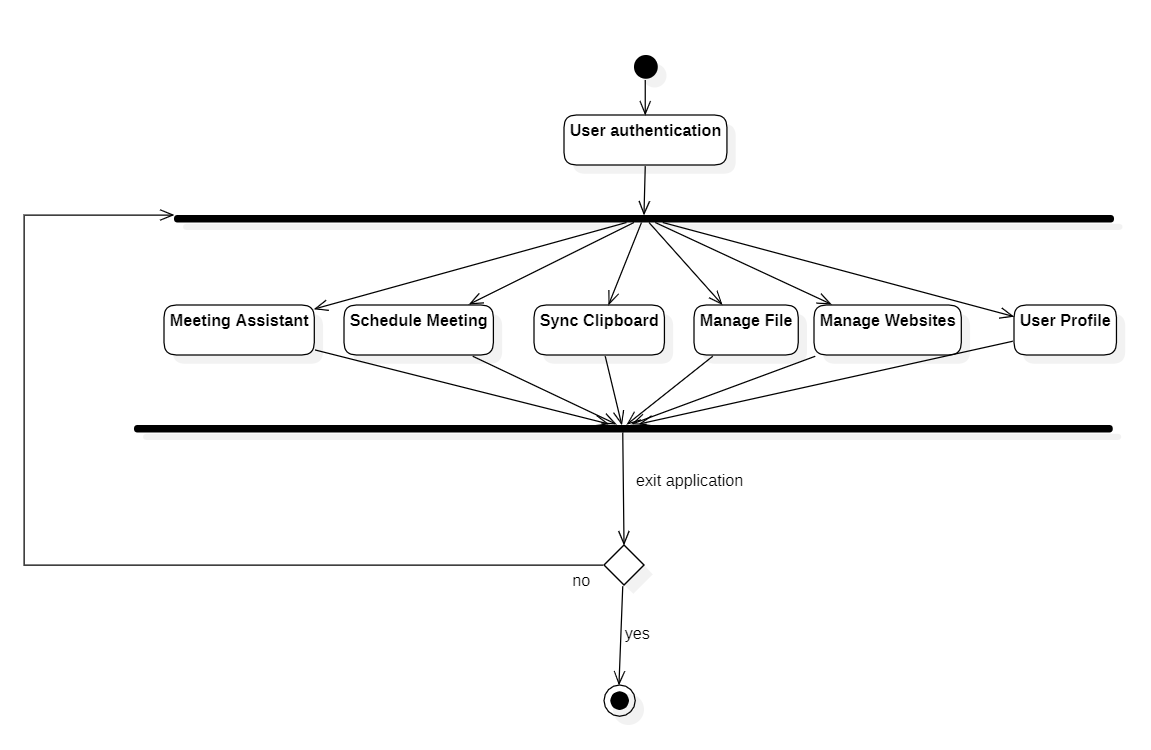
## Detailed Design

### Hierarchy of Harmonize Space



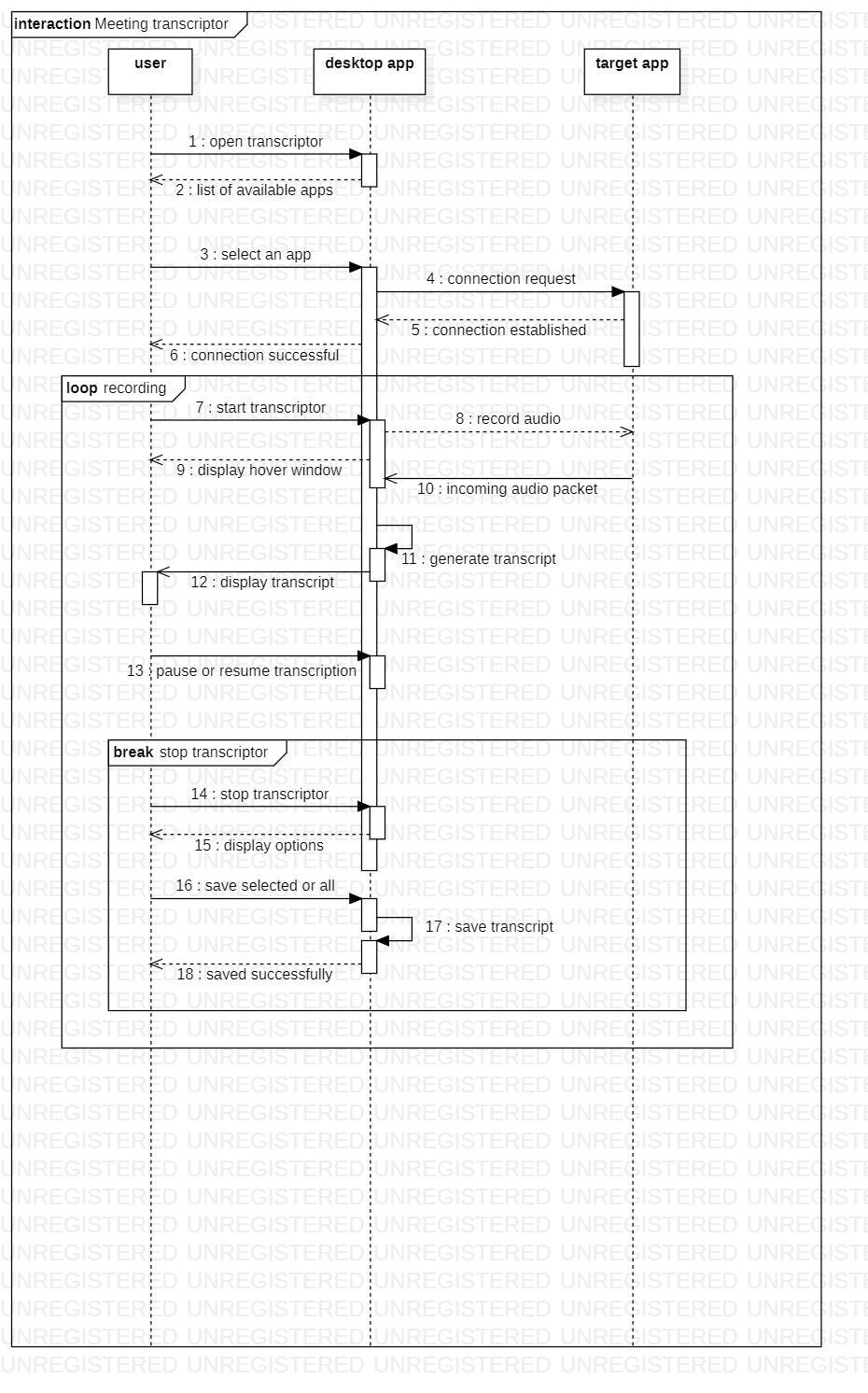
**Fig 3: Architecture Diagram**

### Hierarchy of modules for Desktop and Mobile App



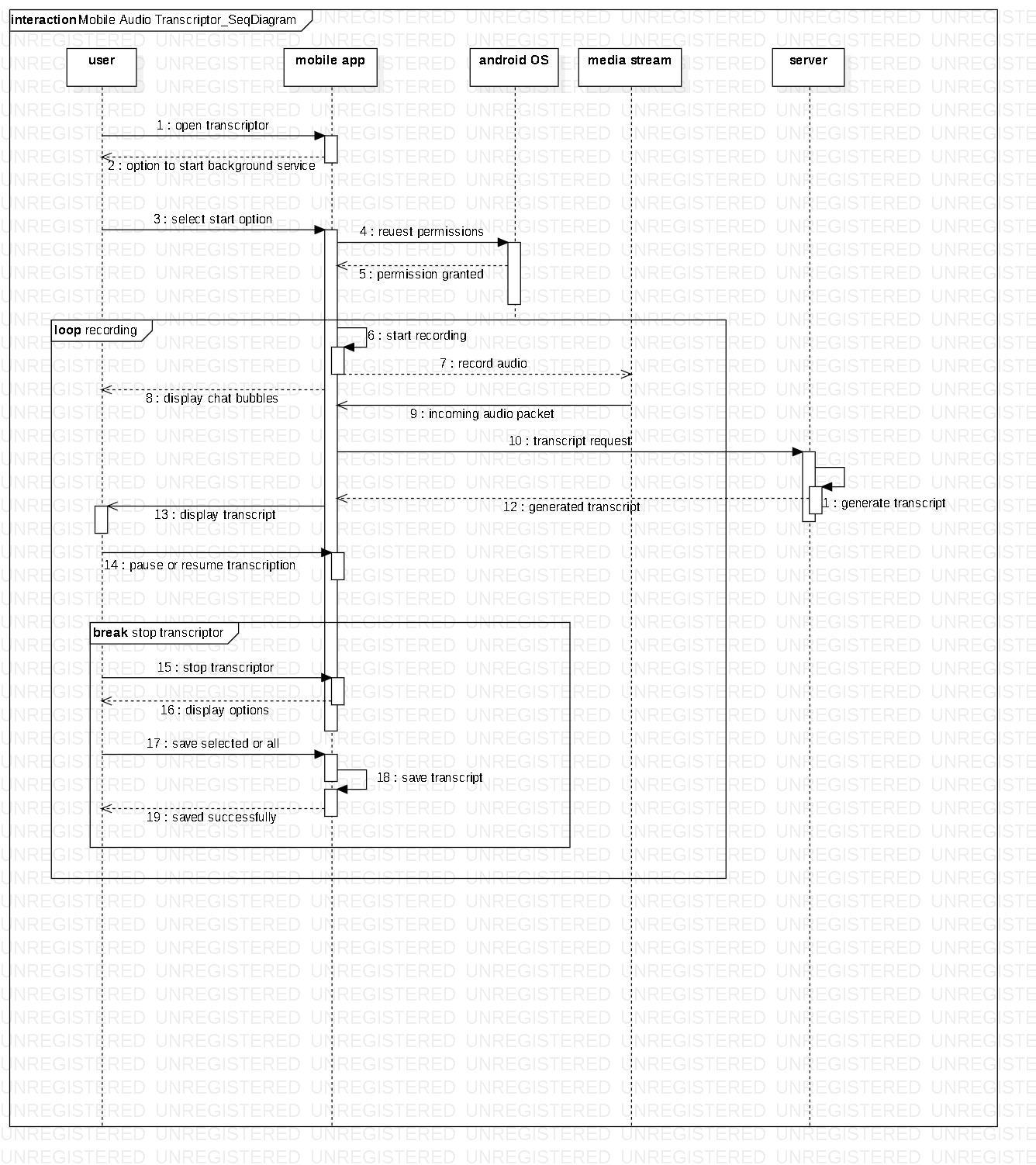
**Fig 4: Hierarchy of Modules**

### Desktop Audio Transcription (DAT)



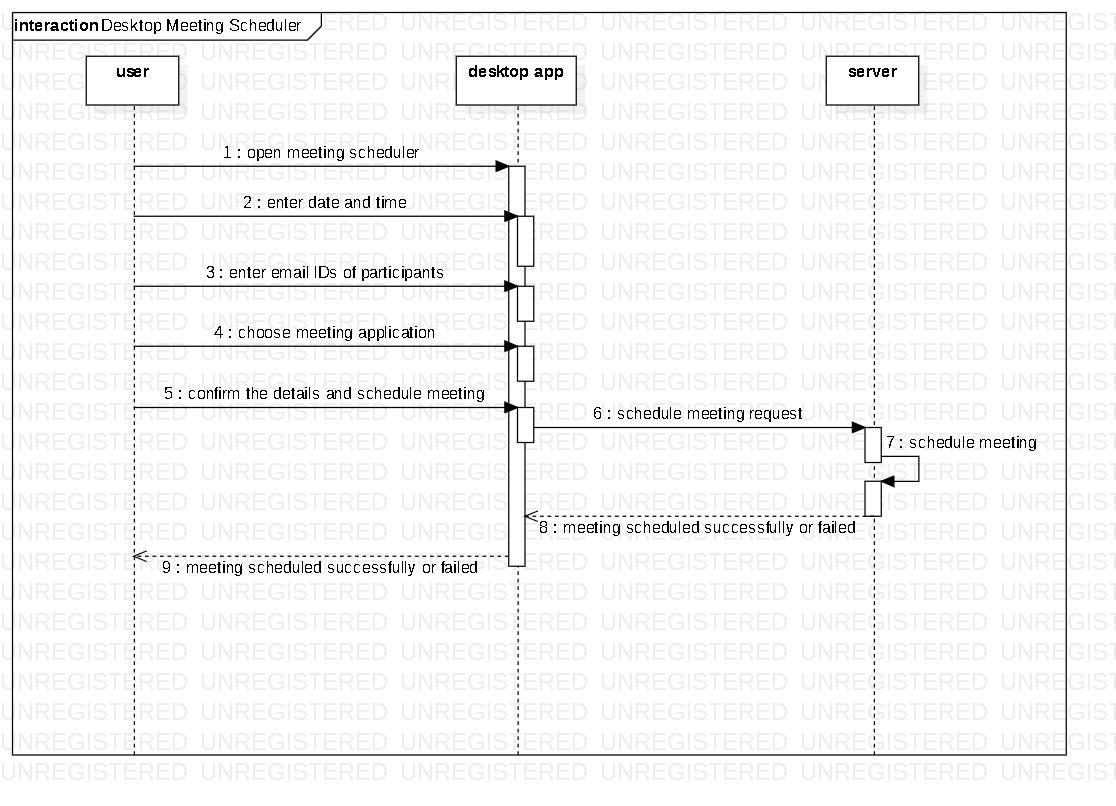
**Fig 5: Desktop Audio Transcription : Sequence Diagram**

### Mobile Audio Transcription (MAT)



**Fig 6: Mobile Audio Transcription: Sequence Diagram**

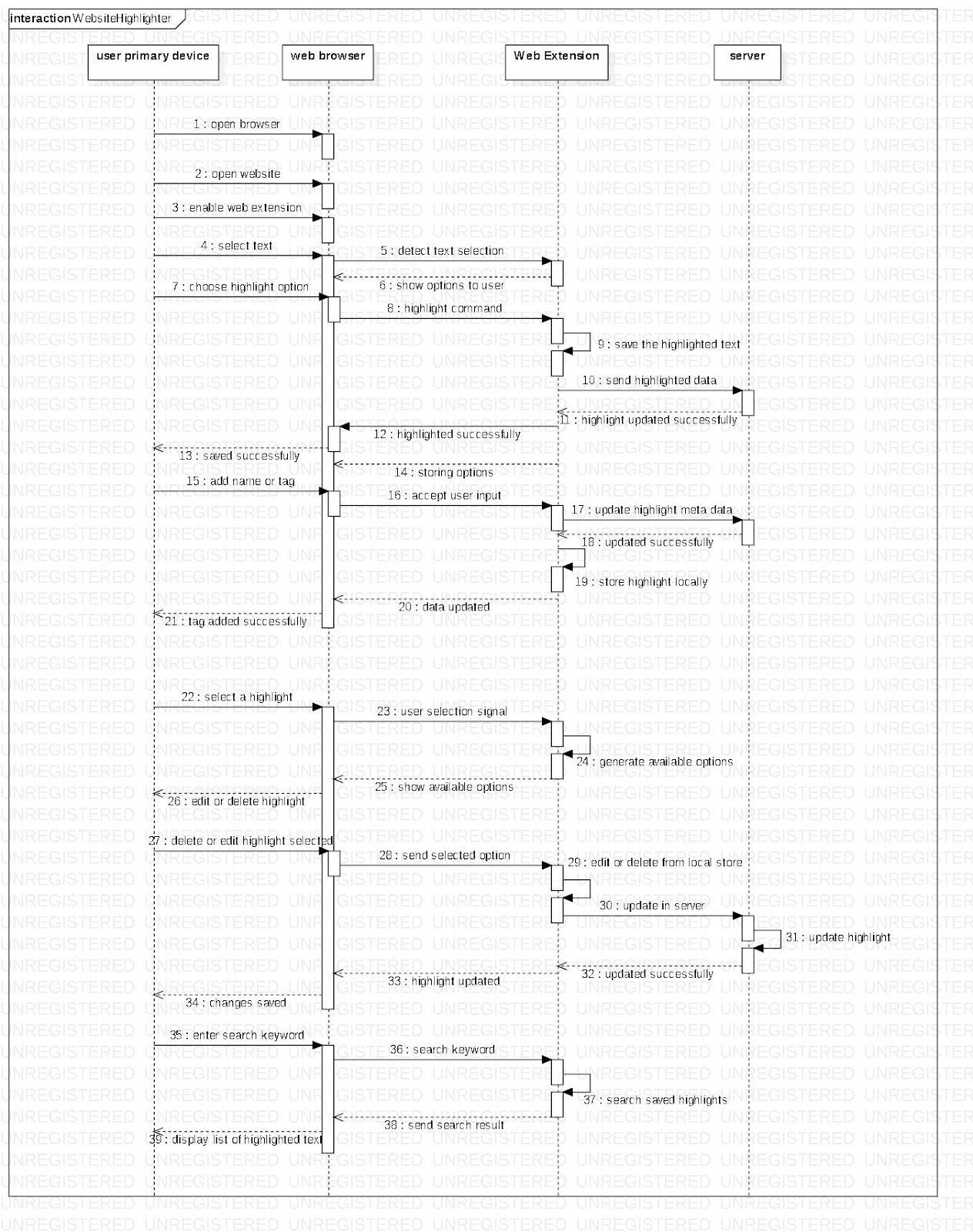
### Meeting Scheduler (MS)



**Fig 7: Meeting Scheduler: Sequence Diagram**

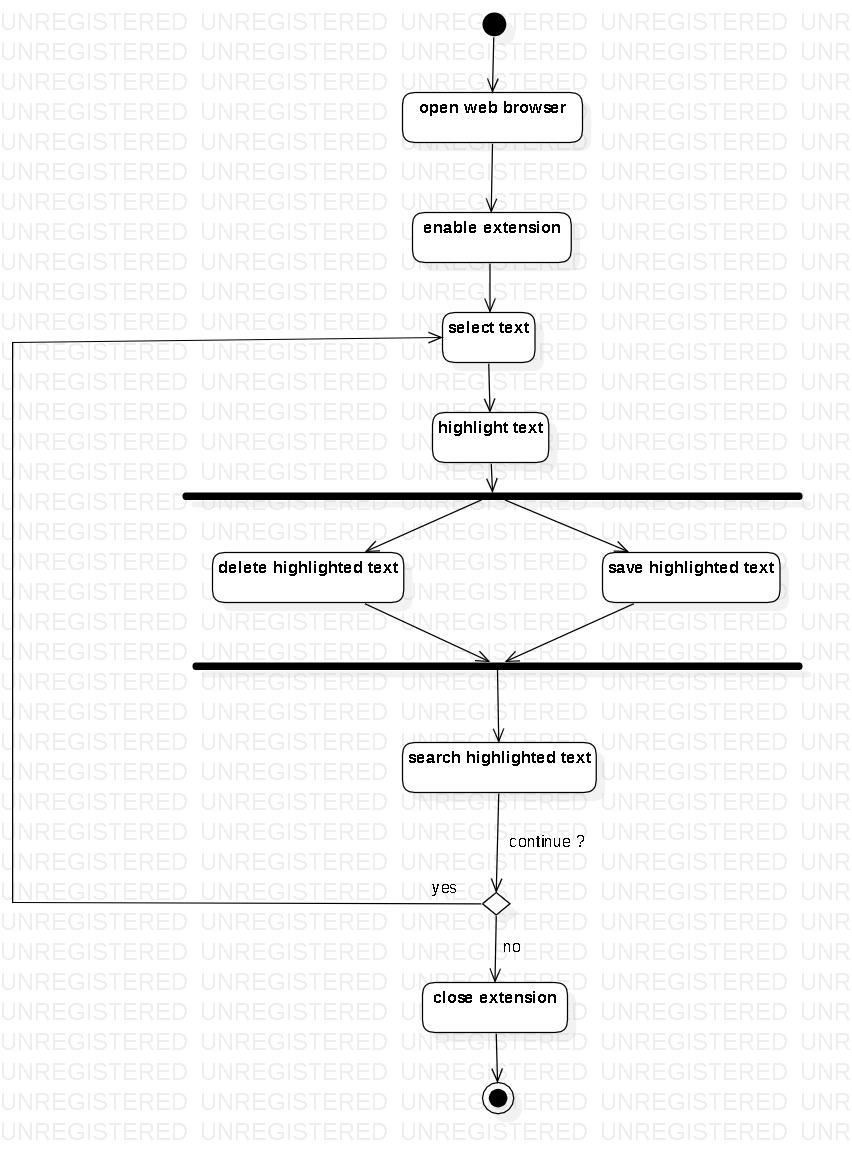
### Website Highlighter (WH)

5.2.6.1: Sequence Diagram



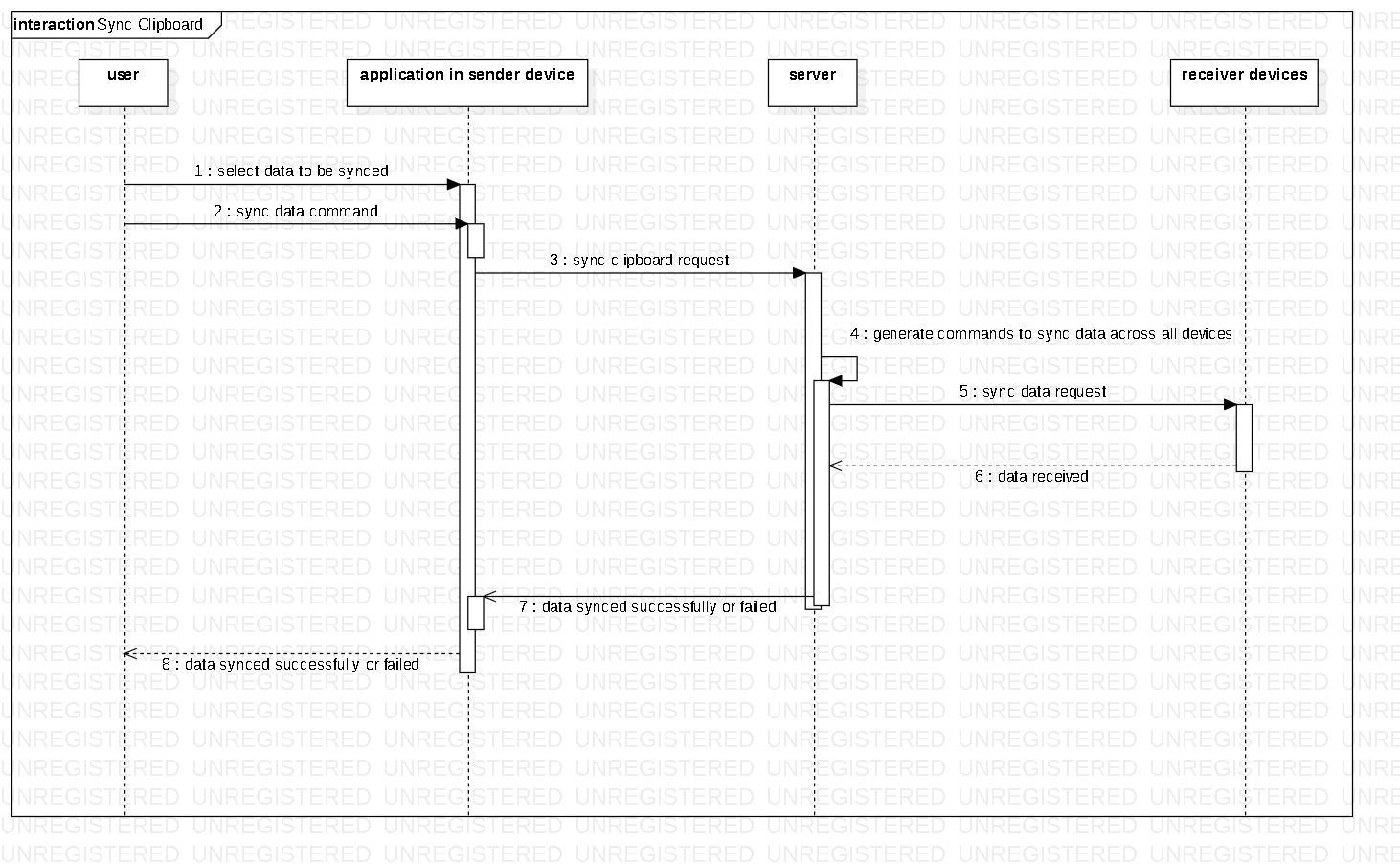
**Fig 9: Website Highlighter: Sequence Diagram**

5.2.6.2: Activity diagram of user

****

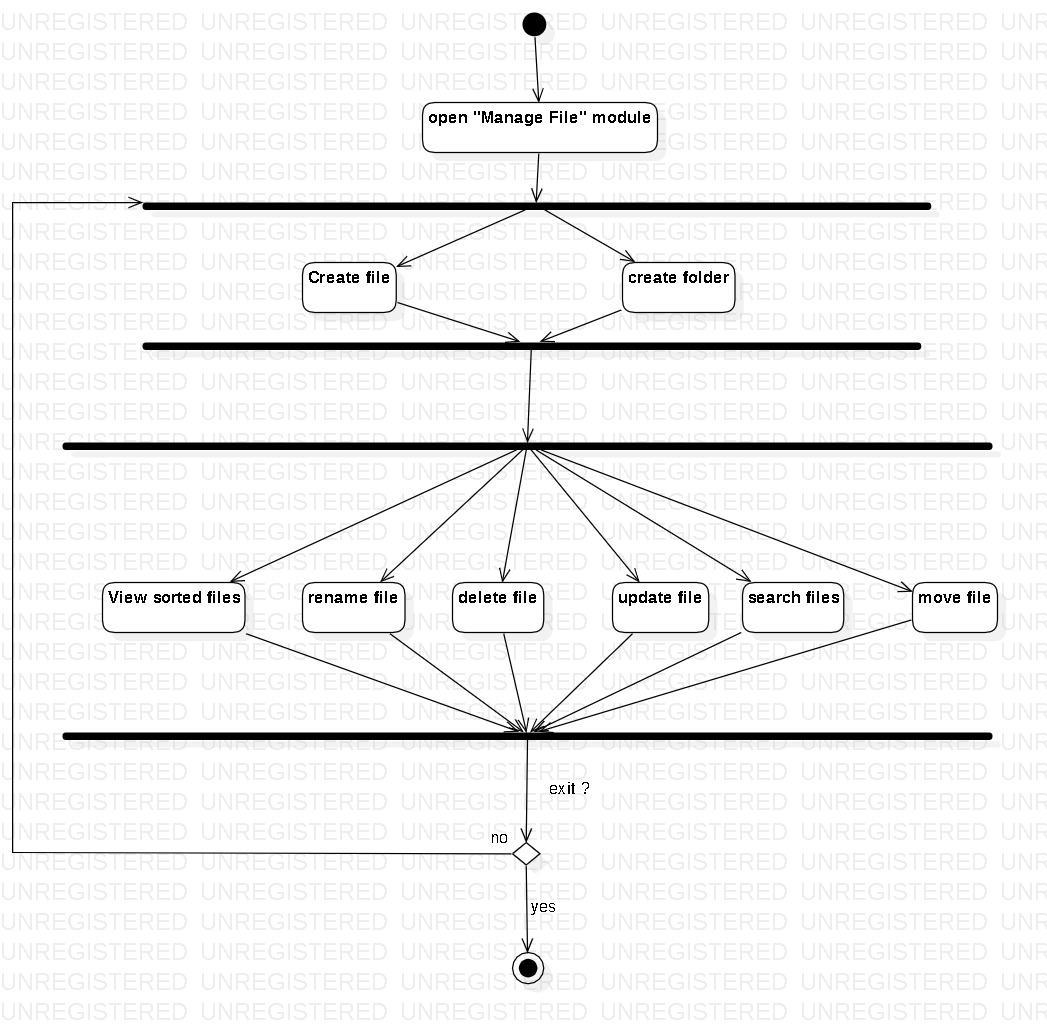
**Fig 10: Website Highlighter: User Activity Diagram**

### Sync Clipboard (SC)

****

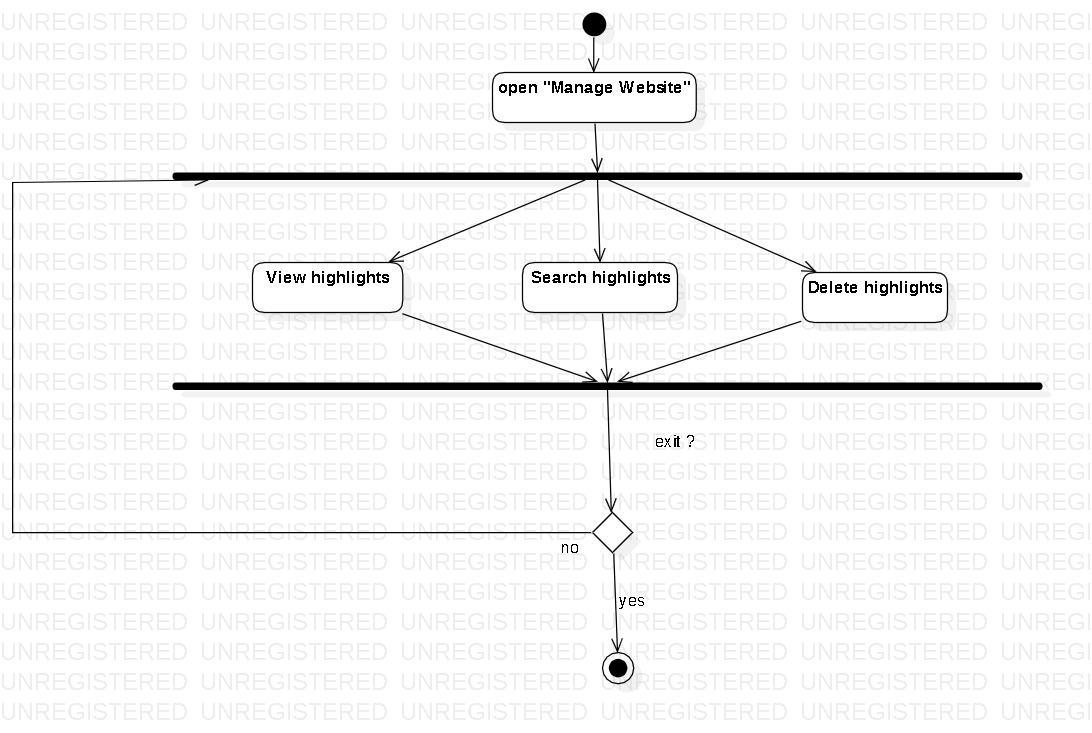
**Fig 11: Sync Clipboard: Sequence Diagram**

### Manage File (MF)



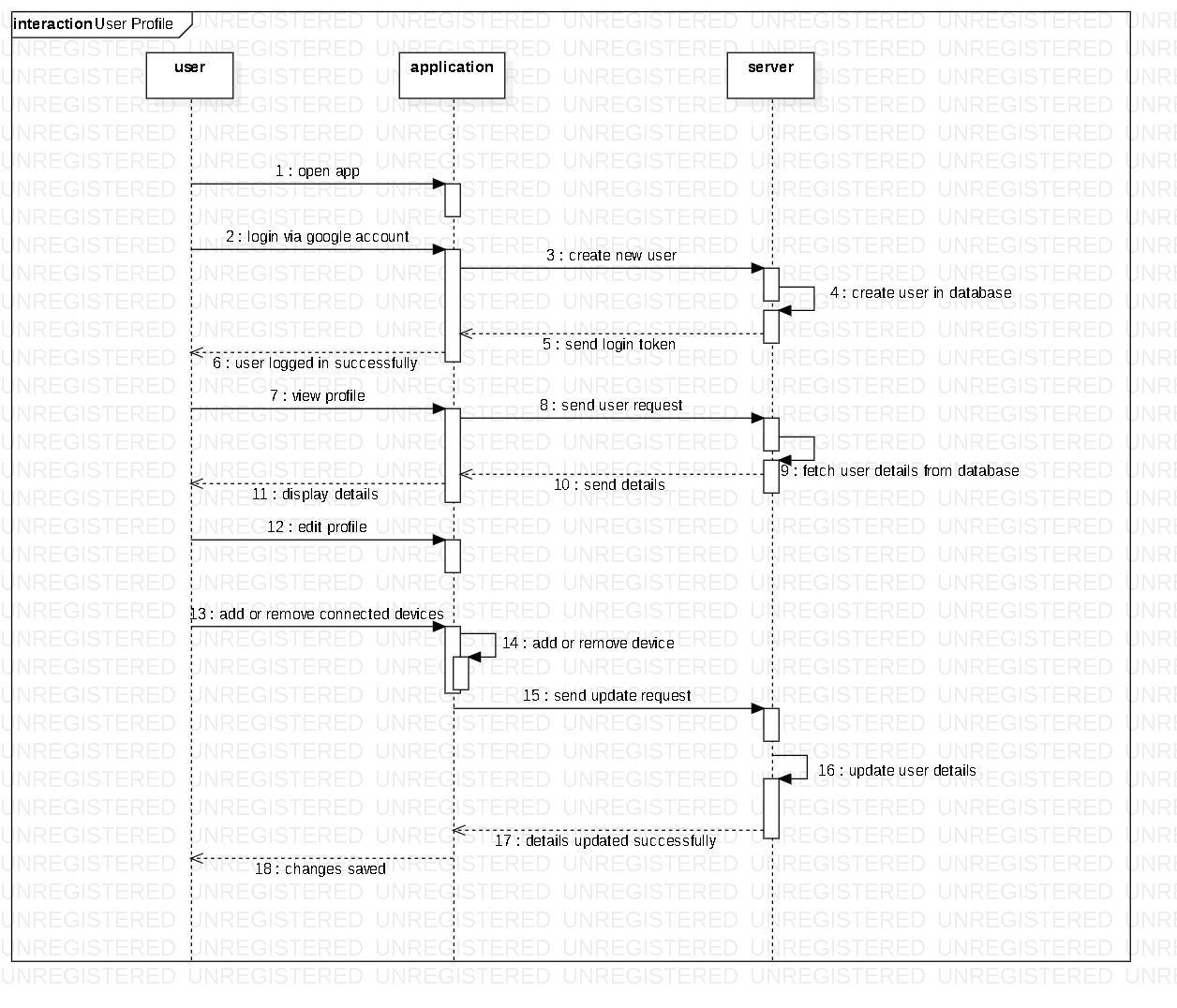
**Fig 12: Manage File: Activity Diagram**

### Manage Website (MW)

****

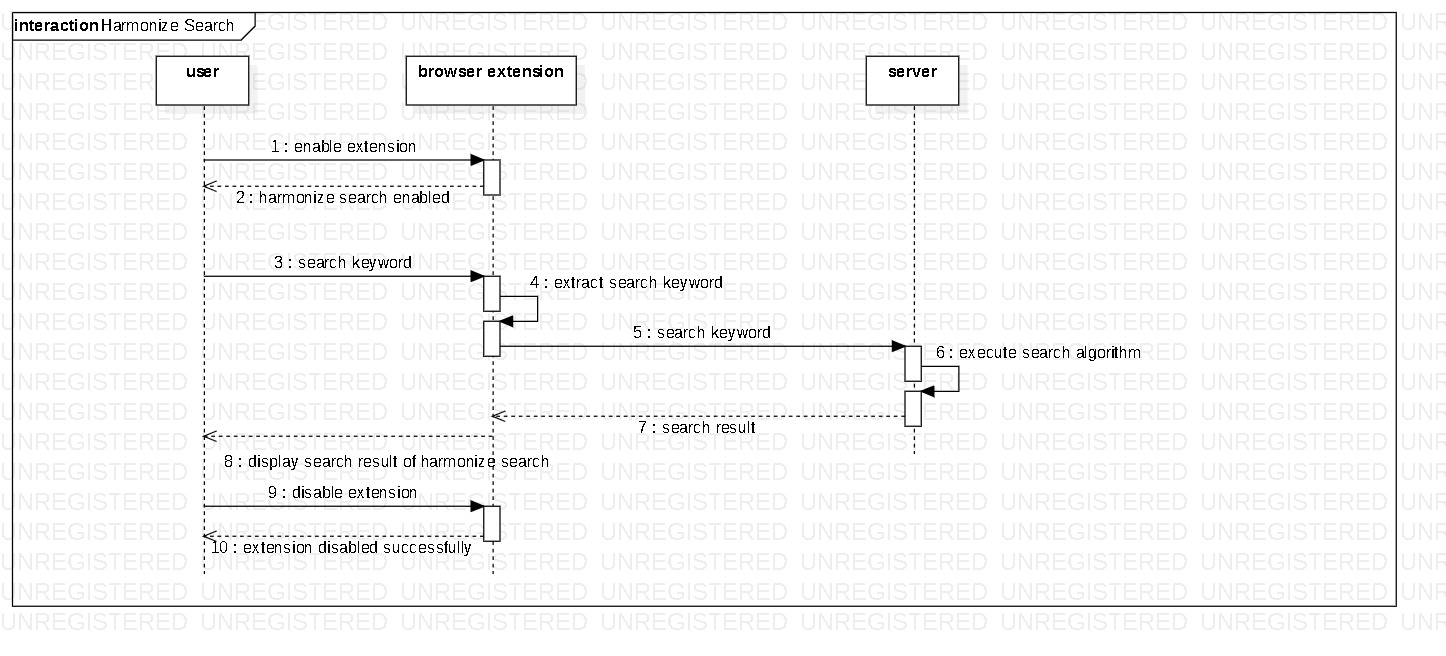
**Fig 13: Manage Website: Activity Diagram**

### User Profile (UP)

****

**Fig 14: User Profile: Sequence Diagram**

### 5.2.11 Harmonize Search (HS)



**Fig 15: Harmonize Search: Sequence Diagram**

# Implementation

## Implementation Details

Create separate sections for separate modules as in Requirement Matrix

6.1.1 Desktop Audio Transcription

* Accept user's choice of desktop screen/ application

We are using electron’s desktopCapturer module to fetch all users' active screens and windows. Then populating them on the user's screen using react component and taking user's choice by onclick event.

<https://bitbucket.org/mate-ai/harmonize-desktop/src/master/src/components/Meeting/Transcriptor/MeetingAssistant.jsx>

* Record incoming audio

We are using `navigator.mediaDevices.getUserMedia` to get the stream of the user's selected window. On successfully receiving the stream we are using `MediaRecorder` to record the stream in chunks. We are also using `ts-ebml` for blob corrections in recorded chunks.

<https://bitbucket.org/mate-ai/harmonize-desktop/src/master/src/components/Meeting/Transcriptor/Controls/Control.jsx>

* Generate transcript

We are converting the above recorded into files using `File` an inbuilt type in javascript. Then sending this file to our transcriptor api to generate text. Our transcriptor then uses `movie.py` with `SpeechRecognition` to get text from audio.

<https://bitbucket.org/mate-ai/harmonize-desktop/src/master/src/components/Meeting/Transcriptor/Controls/Control.jsx>

* Display the transcript bubbles

We are using react `useState` to store text of individual chunks as array. Then rendering these texts as react components as jsx.

<https://bitbucket.org/mate-ai/harmonize-desktop/src/master/src/components/Meeting/Transcriptor/Controls/Control.jsx>

6.1.2 Mobile Audio Transcription

* Requesting permissions -
  + We are using Android’s ContextCompat class to check and request permissions for required File operation, audio recording, etc.
  + Once we receive the user response in the callback of the specific function, we check if it’s granted or not and proceed accordingly.

<https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/fragments/home/HomeFragment.kt>

* Start recording audio -
  + Used MediaProjectionManager to get user permission and initialize it.
  + Start the AudioCaptureService
  + Using this object, inside AudioCaptureService, we created AudioPlaybackCaptureConfiguration and AudioFormat objects.
  + Using this AudioFormat object we create an AudioRecord object which is used to record audio
  + <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/service/audio/AudioCaptureService.kt>
* Break audio into chunks -
  + We create a new file for every chunk and keep writing to it from the audioFormat object in a separate background thread upto a certain limit and then prepare for the the next file in a similar described above
* <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/service/audio/AudioCaptureService.kt>
* Send request to generate transcripts -
  + These chunks are represented as different temporary files which are then sent to the server for generating transcripts. This is done inside HomeViewModel using ApolloClient for Android.
* <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/viewmodels/HomeViewModel.kt>
* Display returned transcripts -
  + We listen for the Api response with the help for LiveData and Observers and display them accordingly inside the HomeFragment
* <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/fragments/home/HomeFragment.kt>

6.1.3 Meeting Scheduler

* Firstly we are accepting the date and time for scheduling meetings inside AddBottomSheetDialog.kt
  + <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/dialogs/AddMeetingBottomSheetDialog.kt>
* Then, accepting the email Ids of all the members joining the meet using a custom Fragment and defining UIs for them.
  + <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/dialogs/AddMeetingBottomSheetDialog.kt>
* Finally scheduling the meeting for them inside MeetingRepo.kt by making api calls to server using ApolloClient for Android
  + <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/repos/MeetingRepo.kt>

6.1.4 Website Highlighter

* Highlight selected text

We are using window’s `getSelection()` to get user’s highlighted text in a website.Then we are checking if the whole selection is in the same container or not. If yes then we activate highlighting by warping the highlighted text into a highlight container and also storing it into local storage.

<https://bitbucket.org/mate-ai/harmonize-highlighter/src/master/injection_script.js>

* Save highlights

Currently we are storing the data in localstorage of the extension using `chrome.storage`. Later on this will be shifted to the server.

<https://bitbucket.org/mate-ai/harmonize-highlighter/src/master/injection_script.js>

6.1.5 Manage Files

* Create File
  + Accepting File name from the user:
    - <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/dialogs/FileFolderAddOptionsDialog.kt>
  + API calls using Apollo Clients:
    - <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/repos/FolderRepo.kt>
* Create Folder
  + Accepting Folder name from the user:
    - <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/dialogs/FileFolderAddOptionsDialog.kt>
  + API calls using Apollo Clients:
    - <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/repos/FolderRepo.kt>
* Delete File
  + Confirmation delete dialogue for user confirmation:
    - <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/dialogs/DeleteConfirmationDialog.kt>
  + API calls using Apollo Clients:
    - <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/repos/FolderRepo.kt>
* Rename File
  + Asking user to input file name for updation:
    - <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/dialogs/FileFolderEditOptionsDialog.kt>
  + API calls using Apollo Clients:
    - <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/repos/FolderRepo.kt>
* Search File
  + Taking Search input from the user:
    - <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/fragments/folder/FileSearchFragment.kt>
  + API calls using Apollo Clients:
    - <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/repos/FolderRepo.kt>
* Update File
  + Asking user to input file name for updation:
    - <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/dialogs/FileFolderEditOptionsDialog.kt>
  + API calls using Apollo Clients:
    - <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/repos/FolderRepo.kt>

6.1.6 Manage Website

* Desktop
* Mobile

6.1.7 User Profile

* Desktop
  + Login

When the user opens the app and there is no user token then `harmonize` is registered as a protocol using `Protocol-registry` and the user is redirected to our auth portal. There user completes its authentication using `Google Sign-up` then on successful login user’s token to stored in the app.

<https://bitbucket.org/mate-ai/harmonize-desktop/src/master/src/main.js>

* + View User Profile

We are calling our profile api with the above token to get the user profile.

Based on the token our server extracts the user’s data from the database and responds accordingly.

Then finally we are populating the user's profile using react component in jsx.

<https://bitbucket.org/mate-ai/harmonize-desktop/src/master/src/components/Profile/Profile.jsx>

* Mobile
  + Login
    - We get the token from the redirect url in this activity and make a call to our AuthViewModel function.
    - After getting the server response stores the token locally for faster access in future.
      * <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/activities/auth/AuthActivity.kt>
    - We make an api call here to our server to get a customized server token, which is used as an authentication parameter hereafter for all the server endpoints to ensure security.
      * <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/activities/auth/AuthViewModel.kt>

* + View User Profile
    - We are able to see the user name, email-Id and the profile image.
      * <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/ui/fragments/notifications/ProfileFragment.kt>
    - We make a GET API call to fetch the user details.
      * <https://bitbucket.org/mate-ai/harmonize-android/src/dev-transcript/app/src/main/java/com/example/harmonizespace/viewmodels/ProfileViewModel.kt>

6.1.8 Harmonize Search

* Take input from google search
* Show result along with google search

6.1.9 Sync Clipboard

* Desktop
* Mobile

## System Installation Steps

System Installation steps with detailed instruction of compilation, execution and other necessary setups

**Harmonize Desktop Frontend**

In the project directory, run:

yarn start

Runs the app in the development mode. Open http://localhost:3000 to view it in the browser.

The page will reload if you make edits. You will also see any lint errors in the console.

yarn test

Launches the test runner in the interactive watch mode. See the section about running tests for more information.

yarn build

Builds the app for production to the build folder. It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes. Your app is ready to be deployed!

**Harmonize Server**

To get started, install the required node modules:

npm install

Run

Then issue the following command to run the server:

npm start

Or use the following command to run the server in development mode:

npm run dev

**Harmonize Android**

1. Clone this project

2. Open this project in Android Studio

3. Android studio will give all the suggestions (Adding adk, doing gradle setup)

4. Once all the setup and gradle build is done. Go to Run option in Studio and press run as .

5. If you find the error "Failed to install the following Android SDK packages as some licenses have not been accepted." update the sdk tools of the sdk library which is present at the top right corner of the screen.

**Harmonize Highlighter**

1. Clone the repository from Bitbucket
2. navigate to chrome://extensions
3. Enable developers mode.
4. Click the load unpacked option and select your folder where you cloned this repository.
5. Save the changes, in no time you will be seeing the extension enabled.
6. Click on the extension icon, enjoy highlighting.

## System Usage Instructions

System usage instructions with screen layouts and detailed instructions

# Test Results and Analysis

Provide test plan, test data and test scripts (as applicable). Test Plan should be provided in tabular format with Sl, Test Case, Expected Results, Observed Results and Status, where each Test Case should be represented with distinct id, prefixed with “T-<module>-“, where module represents the short code of the respective design module. Test Case numbers should be matching as stated in the Requirement Matrix.

Appropriate definition of ‘Performance Metrics’ (in terms of which performance is evaluated, as for example Classification Accuracy, Recall, Precision, Mean Squared Error, Sensitivity, Specificity,..etc.) should be included if required.

Depending on your specific project, test results can be represented as a table of data and corresponding pie chart / bar chart (if required).

Analysis of test results should be discussed in terms of a few bullet points.

# Conclusion

## Project Benefits

* **Meeting -** Our project will enable users to generate live transcripts, take notes, etc. thus allowing them to better manage their meetings and stay organized.
* **Sharing -** Users will be able to share contents across their devices with much ease as compared to the existing crude process of leveraging social media.
* **Browsing -** Enables users to highlight important texts in the websites they visit, which allows them to better navigate to the specific section of a particular article/blog.

These are just a few benefits we have cited, the main aim is to improve productivity and create a better ecosystem for being organized.

## Future Scope for improvements

* Schedule meeting using voice commands
* Feature to organize online meeting

# References / Bibliography

1. J. C. Evers, “From the Past into the Future. How Technological Developments   Change Our Ways of Data Collection, Transcription and Analysis”, FQS, vol. 12, no. 1, Nov. 2010.
2. S. E. Clayman, V. T. Gills, "Conversation Analysis", in Melissa Hardy & Alan Bryman (Eds.), *Handbook of data analysis*, London: Sage, 2004, pp-589-606
3. R. Singh, S. Awasthi, “Updated comparative analysis on video conferencing platforms-zoom, Google meet, Microsoft Teams, WebEx Teams and GoToMeetings”, *EasyChair Preprint*, *4026*, 2020, 1-9.
4. *MediaRecorder.AudioSource | Android Developers*, Android. [Online]. Available: https://developer.android.com/reference/android/media/MediaRecorder.AudioSource