

Audio Freak – Technical Documentation

Group B
February 7, 2023



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Contents

1	Introduction	4
2	Technical Overview	6
2.1	Getting started(Tutorial)	6
2.2	Technical & Architecture Overview	6
2.2.1	Android Studio	7
2.2.2	Firebase	7
2.3	Implementation	8
2.4	Database	9
2.4.1	Audio files	9
2.4.2	Users	10
2.4.3	Messages	10
2.4.4	Permissions	11
2.4.5	Synchronized Playback room	12
2.4.6	Recommended audio	13
2.4.7	Reports	14
3	Feature Showcase	15
3.1	Login	15
3.2	Homepage	18
3.2.1	Module Switching	18
3.2.2	Recommendation	20
3.2.3	Playlist	20
3.2.4	Play function: play, pause, back, fast forward	21
3.2.5	Audio file pagination	23
3.3	Live chat room	23
3.3.1	"Save-to-read"	24
3.4	Direct message	26
3.4.1	Friends List	26

3.4.2	Message	27
3.4.3	Block	28
3.4.4	Invitation for synchronized playback	29
3.4.5	Recall	30
3.4.6	Report	31
3.5	Synchronized Playback	32
3.5.1	Enter the synchronized playback room	32
3.5.2	Components and Functions	33
3.6	Profile	34
3.6.1	Profile modify	36
3.6.2	Upload	39
3.7	Administrative Access	40
4	Feature List	42
4.1	Main features	42
4.2	Sign-up and Login	43
4.3	Settings	43

1 Introduction

We all know that due to the pandemic, people are not allowed to go to parties and stay with other people. It is easy for people to feel lonely and depressed under this circumstance. People need company. And as we all know, music always has this power to heal people, inspire, and reduce feelings of loneliness in people. But let's consider carefully: Does the music heal the lonely person or does the person who is listening to the music with them heal them? The most crucial element, in our opinion, is the person who listening to the music with them. So we made a decision to develop an application that allows people to listen to music with other persons.

Many reading enthusiasts believe that the three main benefits of choosing to listen to books are convenience, ease and freedom of the eyes. Whether you are on the road, walking in the crowd, or lying in bed relaxing your tired body to sleep, audiobooks can make your reading desire come true anytime and anywhere. As social media becomes more and more embedded in people's lives, it is difficult for people to have a lot of time to read quietly, but at the same time, the world is changing so fast that people's demand for knowledge is growing. In this situation, "listening to books", as a low-threshold learning method, has found a "balance" between fast-paced life and learning, meeting people's need to read. Listening to audiobooks is a new and increasingly popular form of leisure and relaxation in people's lives. However, there is no application, which can satisfy the needs of users that simultaneously listening to music and audiobook, chatting with others, making comment for the same audio, listening to the music with lover or friend in a synchronized room, which only their two people in it. Therefore, we think this is a project that is worth developing.

The goal of the **Internet Practical Course Telecooperation**, is letting each participating group creates an Android application with networking functionality. This semester the lecturers gave each group the topic of streaming application. For us, this was a great opportunity to achieve our idea. We decided to create an application that an application that simultaneously caters to users who could switch between listening to songs or audiobooks at will, communicating and chatting with everyone who likes the same audio

in a live room, and listening to music in a two-person synchronized playing room. We built our application with Android Studio and Firebase.



2 Technical Overview

2.1 Getting started(Tutorial)

There are two ways to install the app to the installation device:

1. Download the APK file from the repository to the Android device and install it.
2. Connect the Android device with a computer, run the project source code of the app with Android Studio, and run the app on that device.

Open the app after installing it, register or sign in and enjoy it.

Note that this app is compatible with Android devices with API 32 or higher, it may not work on lower version devices.

2.2 Technical & Architecture Overview



Figure 2.1: Technical architecture

As the figure 2.1 shows, we use a combination of Android Studio and Firebase to develop our App.

2.2.1 Android Studio

Android Studio is an IDE used for developing android apps, which is an integrated development environment for Google's Android platform.

2.2.2 Firebase

Services and tools provided by Google Firebase are used to help develop our App, which is a web and mobile application development platform. It not only gives a framework to create the apps but also has powerful tools to build and improve the app:

- Authentication: provides backend services and ready-made UI libraries to authenticate users to the app. It supports authentication using email/passwords, and other identity providers like Google and Facebook.
- Real-time database: is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. All of the clients share one Realtime Database instance and automatically receive updates with the newest data. The connection between the database and the application is carried out by means of the APIs and SDKs, which means that we don't need to program server-side.
- Cloud Storages: is for object storage service built for Google scale, which can store, sync, and query data, like music, audiobooks and user avatar.
- Crash reports, Analytics and Hosting etc. are also used in the development process.

2.3 Implementation

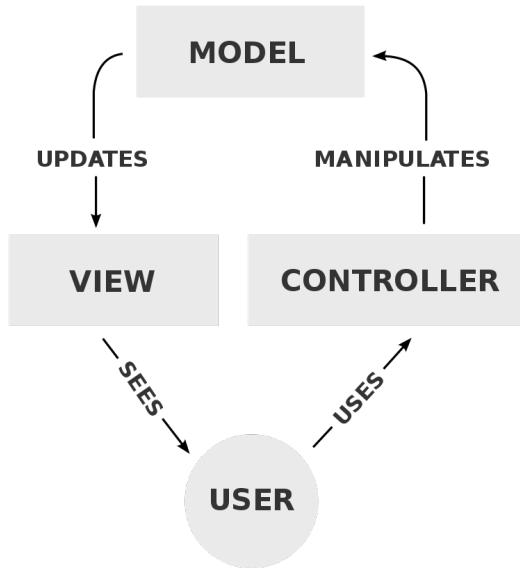


Figure 2.2: MVC-Process

The model-view-controller design pattern(2.2) is used for the implementation because of its code reusability and intuitive development process. The data model can be defined in the model, the View corresponds to the User Interface and gets the data from the model, and the controller(adapter) contains listeners and observers that can update the data of the model and update the presentation of the view.

When the user enters an interface through an operation, the default controller receives the action, and then calls the corresponding method in the module (the module encapsulates the data related to the business logic of the application and the processing method for the data). The data results obtained by the methods in the module will be transmitted to the view, and the view can display the data to the user purposefully. The user thus fetches the data and can choose to proceed to the next step, which triggers this cycle again.

2.4 Database

Audio Freak uses Firebase as Database and store all data there. Authentication Database, Cloud Storage Database and Realtime Database provide the storage and reading of all data.

2.4.1 Audio files

Audio files are all stored in Storage Database (in 2.3b). The following variables associated with each audio file are recorded in the Realtime Database (in 2.3b):

- **livechat** (*dictionary*) | live comments on this song
 - **livechatNum** (*int*) | number of live comments
 - **playedTimes** (*int*) | number of playing
 - **likedTimes** (*int*) | number of "Like"

Attributes of each audiofiles and detail of livechat are belows (in 2.3):

- **Context** (*string*) | context of live comments
 - **TimeStamp** (*long*) | time of live comments
 - **userID** (*string*) | who posted the comment (Uid)
 - **userName** (*string*) | who posted the comment (Displayname)

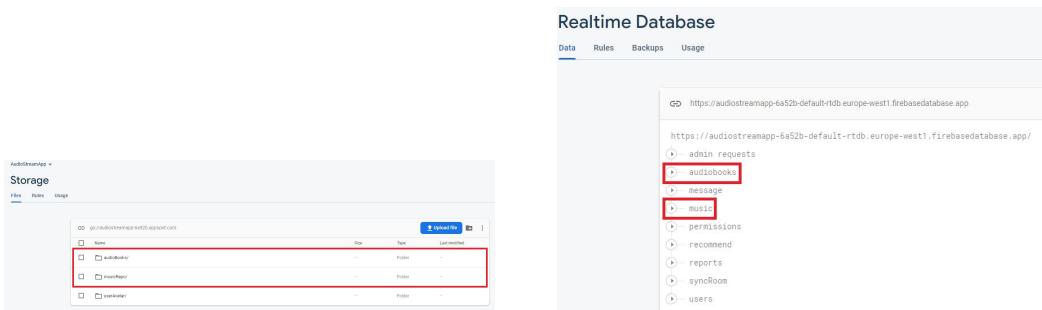


Figure 2.3: Audiofiles

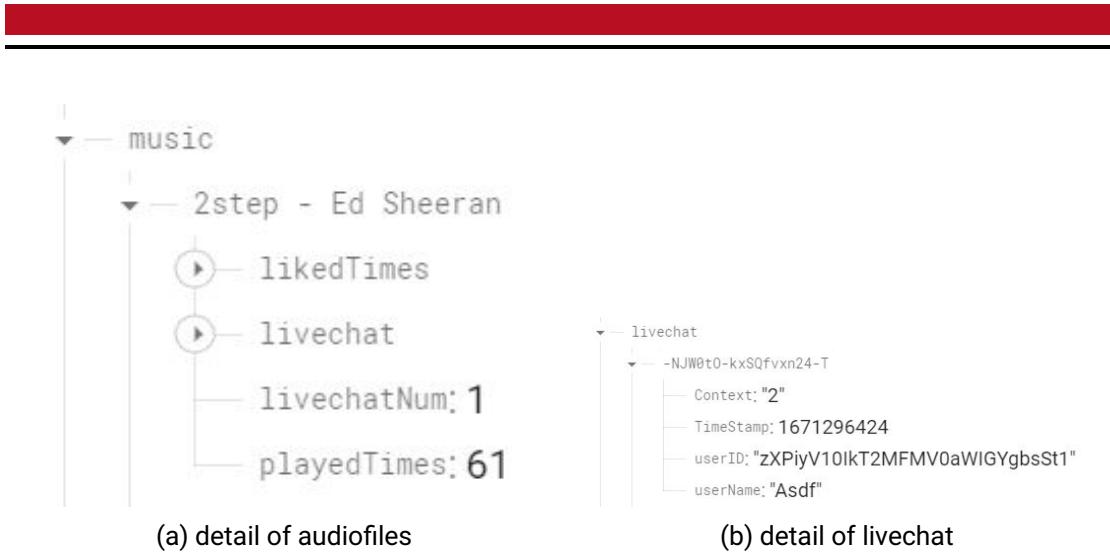


Figure 2.4: Attributes of audiofiles

2.4.2 Users

Data of users are mainly stored in Authentication Database (in 2.5a). And the Avatar of users are stored in Cloud Storage Database (in 2.5b).

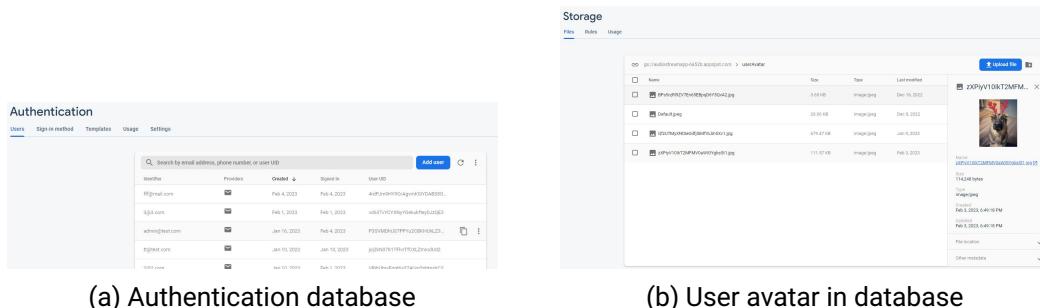


Figure 2.5: Data of users

2.4.3 Messages

Messages are all stored in Realtime Database (in 2.6).

If user A has a conversation with user B and user C, then there are information about A and B in the index "user A/user B" and "user B/user A", and in the index "user A/user C" and "user C/user A" contains the information of user A and user C. Each information are "dictionary" type and has Key and Value. The Key is timestamp and the Value are the attributes below:

- **Context** (*string*) | context of message
- **Receiver** (*string*) | who got the message (Uid)
- **Sender** (*string*) | who sent the message (Uid)
- **TimeStamp** (*long*) | time of the message

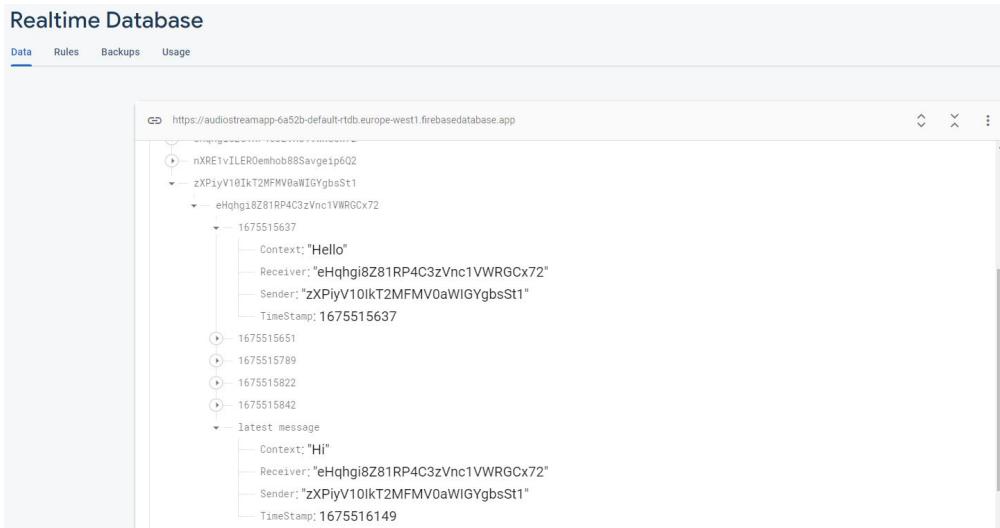


Figure 2.6: Message in Database

2.4.4 Permissions

Permissions are stored in Realtime Database.

If user A sends a message to user B, the state information will be created under the index "user A/user B" (in 2.7, default value is "Normal"). If user B blocks user A, the information will change from "Normal" to "Blocked", and A can no longer send information to B at this time. After user B unblock user A, the value will change back to "Normal".



Figure 2.7: Permissions in Database

2.4.5 Synchronized Playback room

The room information for synchronized playback is stored in Realtime Database.

When user A invites user B to listen to certain music together, an information about synchronized playback room will be established (in 2.8). This information has type "dictionary" and its Key is randomly generated. More details about its Value are below:

- **currentMusicName** (*string*) | name of current playing music or audiobook
- **playStatus** (*bool*) | whether this audio file is playing
- **pos** (*int*) | if playing, what is current position
- **repoName** (*string*) | where is this audio file stored in database
- **visitorStatus** (*bool*) | whether both users are in this room



Figure 2.8: Synchronized Playback room in Database

2.4.6 Recommended audio

The recommended audios are stored in Realtime Database (in 2.9). Each key-value pair (key: string; value: int) under the index "audio A" represents the number of times a certain audio is played as the next one.



Figure 2.9: recommendation in Database

2.4.7 Reports

The reports which are sent to the administrator will be stored in Realtime Database (in 2.10). When a report was sent, an information (type: dictionary) will be saved in database. Its Key is the Id of user who is reported. Its values are below:

- **isBlocked (string)** | the status of the user who is reported, administrator can change this value to punish that user if the report is confirmed
- **livecomments (string)** | the content which was reported



Figure 2.10: reports in Database



3 Feature Showcase

3.1 Login

Register a new account with email address or log in with google account.

For the users we provided two methods to sign in the application. User can log into the application by the way of a google account or their own email address, as the yellow block part shown in Figure 3.1a. Once user chooses to log in with their own email address, they can easily click to the button. They will be lead to sign in page as Figure 3.1c, user can easily type their email address and our database will check if the address has been registered or not. If this address has been registered then application will move to the next step. If this address has not been registered before, then the users will be lead to the Sign up page, as shown in figure 3.1b. If the password is not strong enough, it will hit user the password needs at least 6 characters and a mix of letters and numbers. Finally, click save, a new account has been set up.



(a) Login Page.



(b) Sign up for new users.



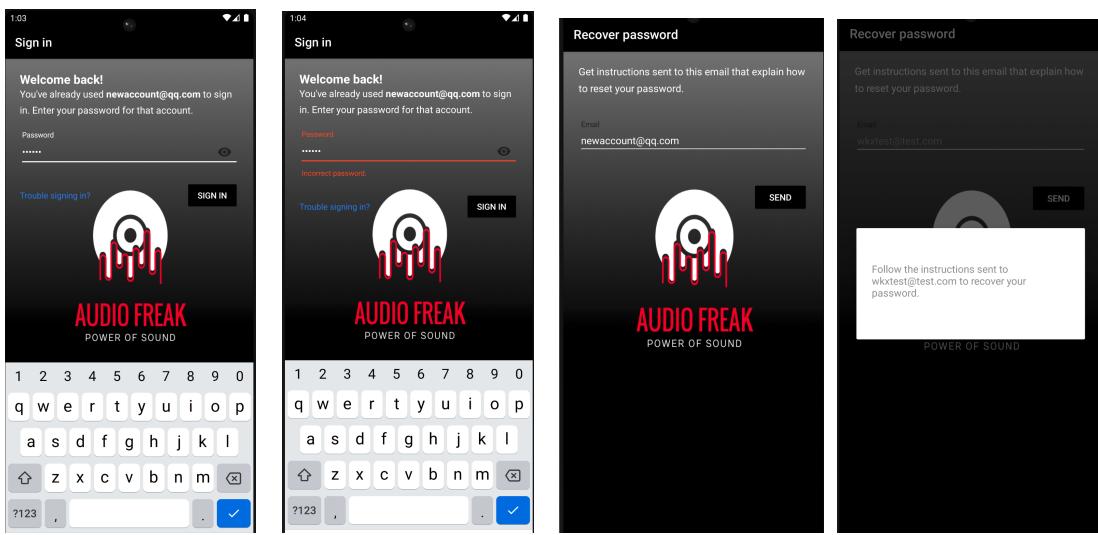
(c) Sign in Page

Figure 3.1: Sign up a new account and sign in with an existing account.



Log in with an existing account.

As the above said, when user has had a account, they can easily type their email address, which is shown as the figure 3.2a. After this step, the application will ask user to type their password in. The database will check if the password match the account or not. If the password is correct then the user login successfully. If the password is wrong then it will show incorrect password as shown in figure 3.2b. When user choose *Trouble signing in?*, which is the blue words in this page. It means user will recover the password, which as shown in figure 3.2c. Clicking send button, a notification will pop up as shown in figure 3.2d. The user's email will then receive an email, the user can recover the password easily by following the steps.



(a) Password Page. (b) Incorrect Password. (c) Recover Password. (d) Reset Password Notification.

Figure 3.2: Sign in an account and the password related issues.

Sign out

For Sign out, user can sign out easily in the Profile Page, which is as narrow points in the figure 3.3a. In case the user just made a mistake, we set up a pop-up prompt to indicate if the user really wants to log out of the account, as shown in figure 3.3b. If the user confirms logout, then the account will be logged out and return to the login screen, which is this page 3.1a.

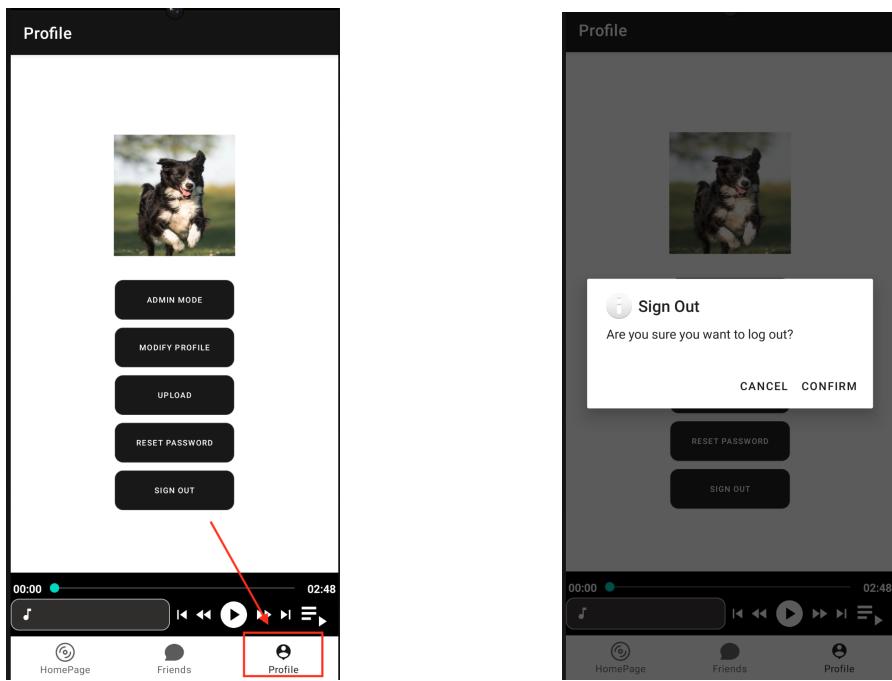


Figure 3.3: Sign out the account.

3.2 Homepage

Homepage is the main page of the application. After the users log into their account, they will automatically enter the homepage. The homepage will be switched to the music mode. See 3.4.

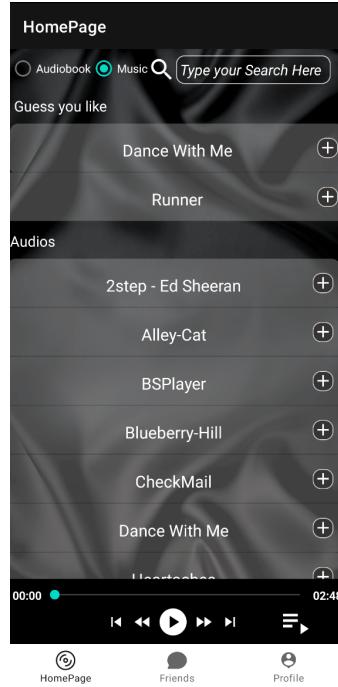


Figure 3.4: Homepage

3.2.1 Module Switching

As showed in 3.5. We can easily click the radio button to switch between audiobook and music. For example we choose audiobook 3.5a or music 3.5b.

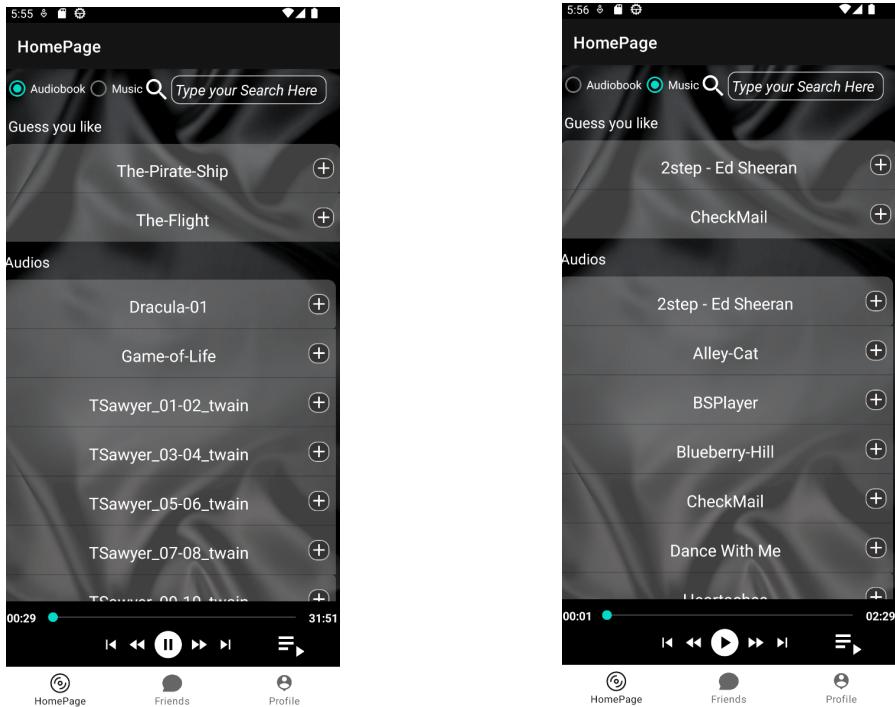


Figure 3.5: Module switching.

3.2.2 Recommendation

We use Bigram model to recommend music and audiobook. Let's have a look at Ngram model first. N-Gram model is based on the assumption that the occurrence of the N-th word is only related to the preceding N-1 words, but not to any other words, and the probability of the whole sentence is the product of the probabilities of the occurrence of each word. These probabilities can be obtained by counting the number of concurrent occurrences of N words directly from the corpus. As showed in 3.1 and 3.2, we set N = 2 here.

When N=2, a bigram model is:

$$P(w_1, w_2, \dots, w_N) = \prod_{i=1}^N P(w_i|w_{i-1}) \quad (3.1)$$

For bigram model:

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}|w_i)}{C(w_{i-1})} \quad (3.2)$$

The following algorithm is designed by referring to the Ngram model to deduce new words. Assume that the selection process for the user to listen to the next audio is similar to the process for the Ngram model to predict the next word. Based on this assumption, deduce the next audio that the user is likely to listen to from the previous audio. The model has good scalability and can extend more gram. We chose Bigram based on Firebase and other considerations. It's acceptable that complexity is $O(n^2)$. Bigram is an easy upgrade to trigram or ngram. Due to higher complexity so we give up trigram or ngram.

As showed in 3.4, From the bottom of "Guess you like", we can retrieve recommended songs and audiobooks based on the current playing song.

3.2.3 Playlist

Our application includes a playlist feature, as shown in 3.7. Users can open the playlist from the playlist button in the bottom right corner of the homepage. To the right of each audio, there is a "+" button that can add the audio to the playlist. The songs in the playlist can be switched to the previous or next song using the control bar at the bottom right of the homepage.

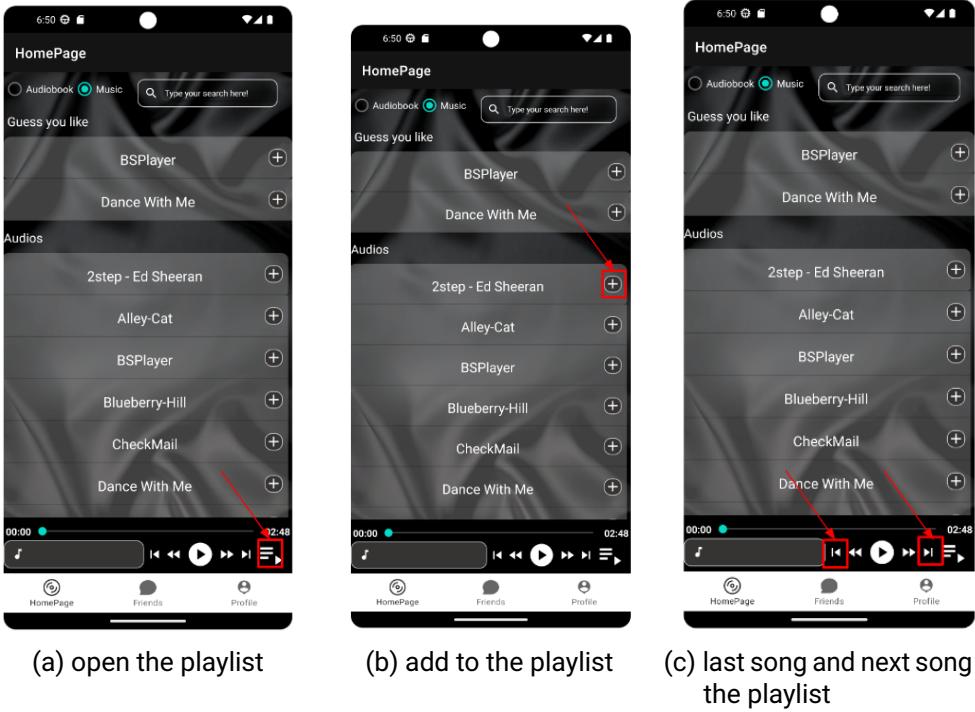
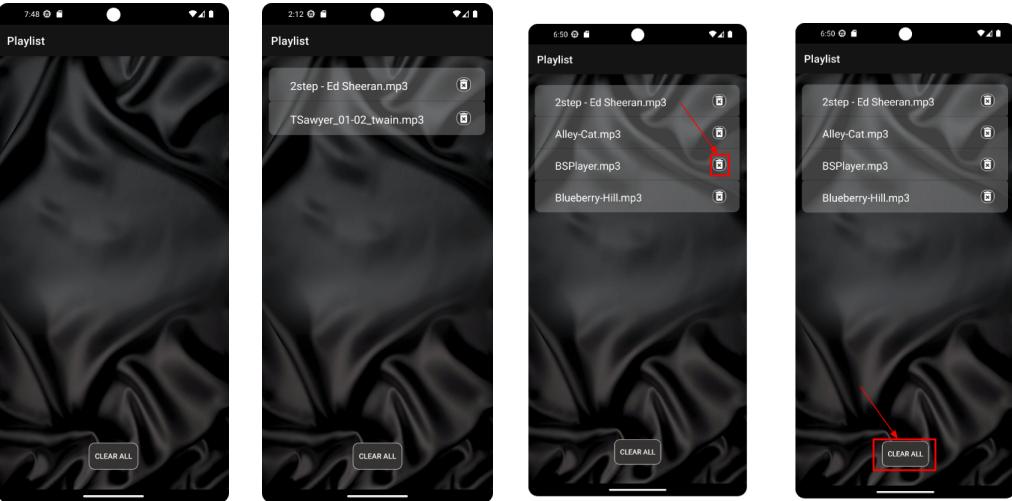


Figure 3.6: The playlist functions in homepage

In the playlist, the audios are played in the order in which they were added. Both audiobooks and music can be added to the playlist as shown in 3.7b. Users can select the audio to play immediately by clicking on the audio in the playlist. Audios can be deleted from the playlist using the trash-bin buttons to the right of the audio as shown in 3.7c. The entire playlist can be cleared using the "CLEAR ALL" button at the bottom of the playlist page as shown in 3.7d.

3.2.4 Play function: play, pause, back, fast forward

For play function we have play, pause, back and fast forward. As shown in 3.8a. When we click the play button, the music will be played smoothly;As shown in 3.8b. If we click the pause button, the music would immediately be stopped;In 3.8c. When we clicked the back button, the music could be back for 5 seconds. In 3.8d Similarly fast forward would let music fast forward for 5 seconds.



(a) an empty playlist (b) Both audiobooks and musics can be added to the playlist.
(c) delete from the playlist (d) clear the playlist

Figure 3.7: The illustration of playlist

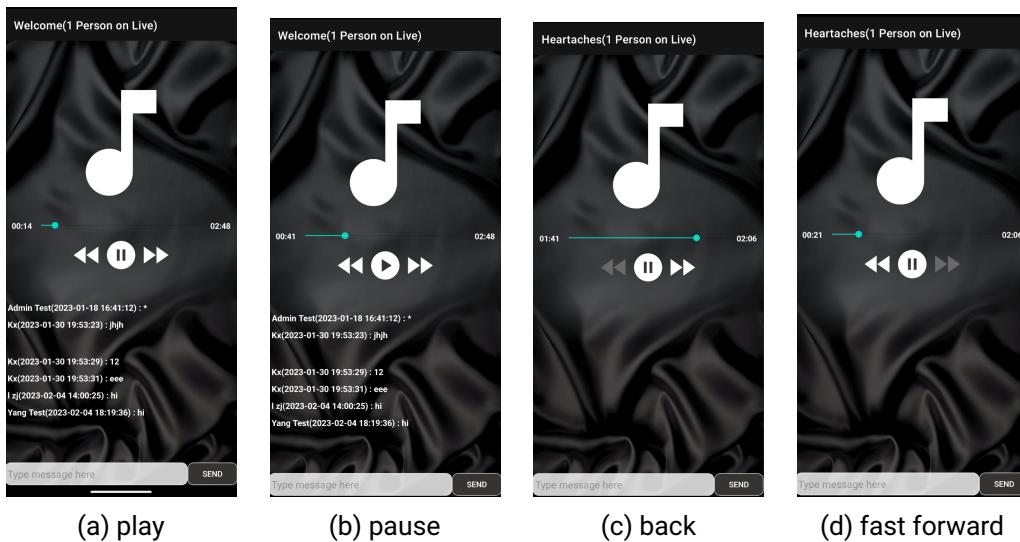
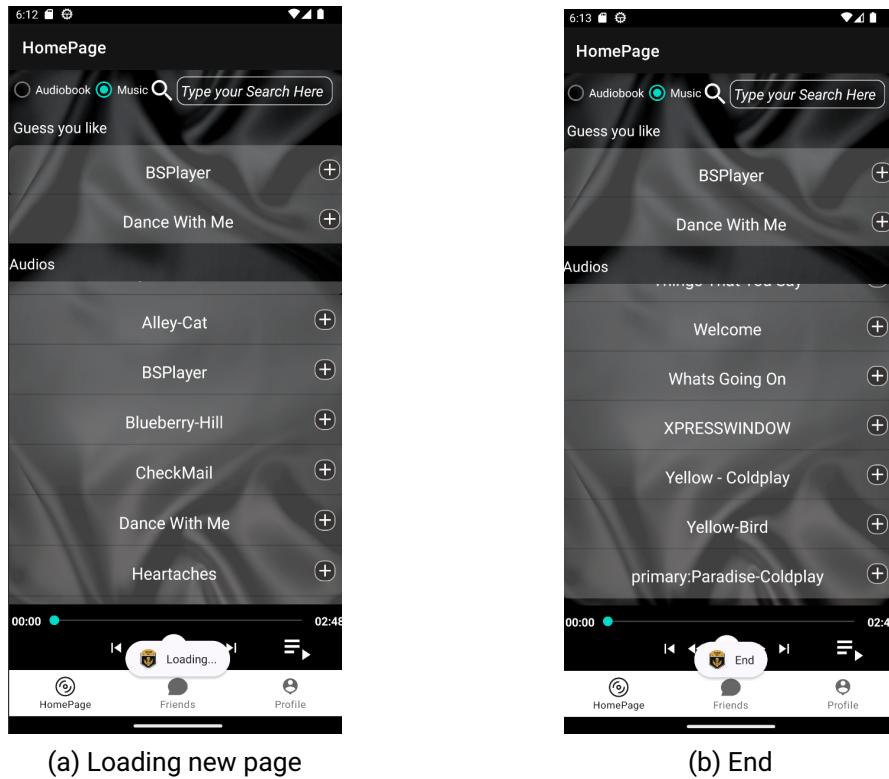


Figure 3.8: Play function

3.2.5 Audio file pagination

We set up paging with the limit size 7. The audios will show seven records at first. As shown in 3.9a, after we scroll down the page. The page will load seven new records. When "end" displays. As shown in 3.9b, all records are loaded.



3.3 Live chat room

As showed in 3.10, while the audio is playing, users can chat with other users in the live chat room. Each audio has its own chat room. The number of people online in the chat room for that audio is displayed in the top left corner of the chat room screen. The chat message is sent from the bottom. Each chat message is displayed after the username and timestamp of the posting. If a user want to send direct message to another user, the user

can click the chat message of that user to jump to the profile page of that user and send message. Up to 20 messages are stored in each chat room.

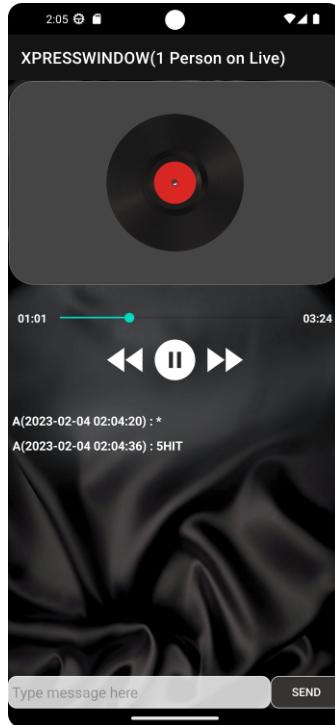
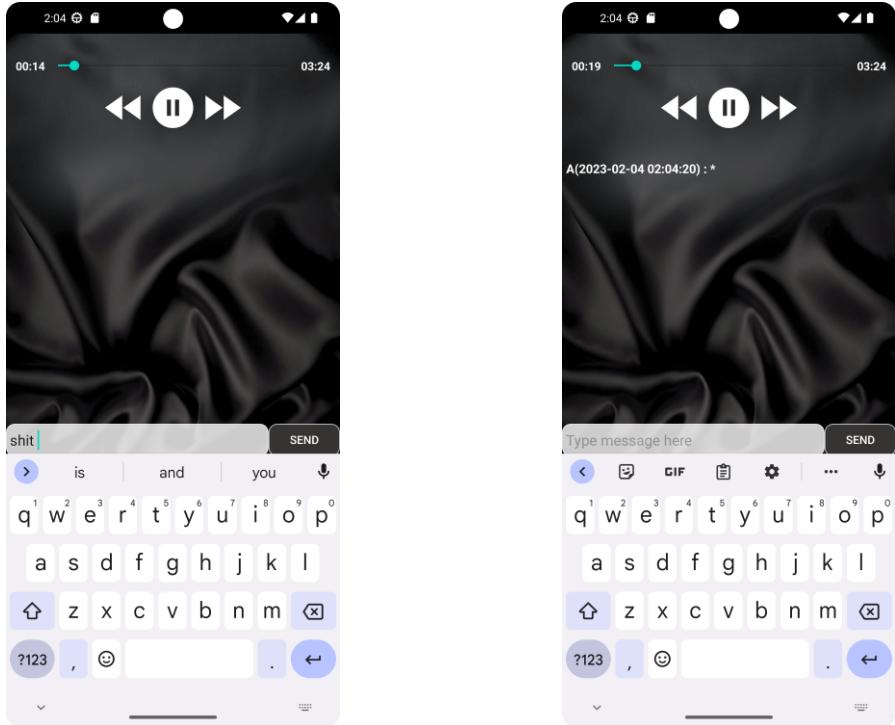


Figure 3.10: The illustration of live chat room

3.3.1 "Save-to-read"

An automatic inappropriate word filter is included for the live chat rooms. The filter is based on the list of bad words and top swear words banned by Google.¹. The detected inappropriate word is automatically replaced with “**” as shown in the 3.11b. The user can also report the inappropriate content to the administrator. With a long press on the inappropriate content, a confirmation box will pop up asking whether the user wishes report the inappropriate user, as shown in 3.12b. After confirmation, a message box informs

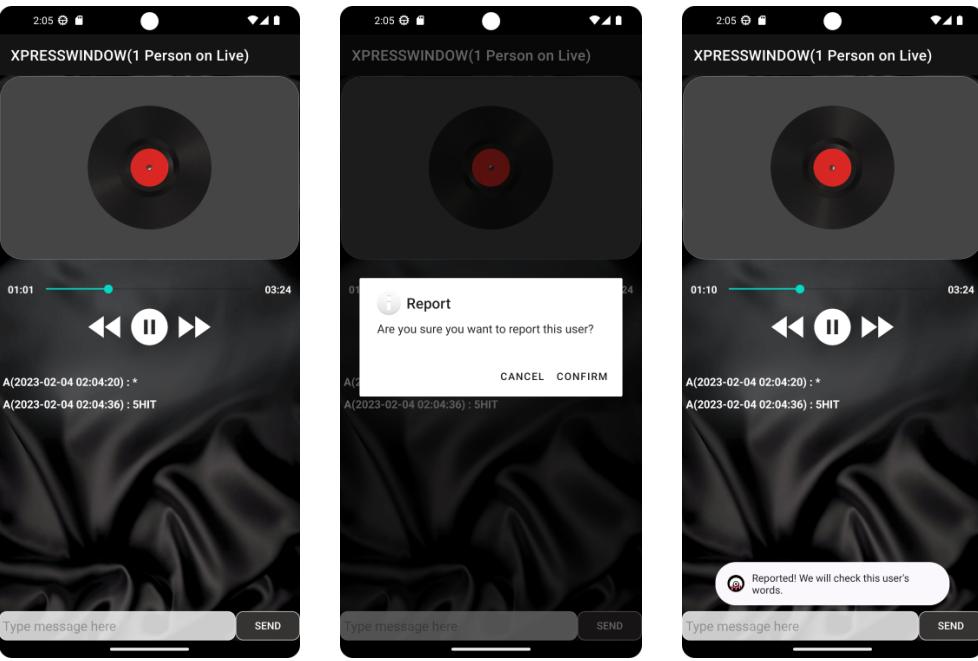
¹<https://github.com/coffee-and-fun/google-profanity-words>



- (a) The user is trying to send inappropriate content.
(b) The inappropriate content is substituted with a "*" after posting

Figure 3.11: The illustration of automatic inappropriate word filter

the user that the report is successful and the administrator will check the inappropriate user's post, as shown in 3.12c.



(a) an inappropriate content with substitution of the letter, which can not be detected by the automatic filter

(b) reporting confirmation

(c) reporting success

Figure 3.12: The illustration of manually reporting of the inappropriate content

3.4 Direct message

Users can send message to other user directly.

3.4.1 Friends List

Friends list will save the chat records within 30days. Users can see the last message here (in 3.13a). Click other user, it will jump to the profile of that user (in 3.13b). Then click "direct message", users can chat now (in 3.13c).

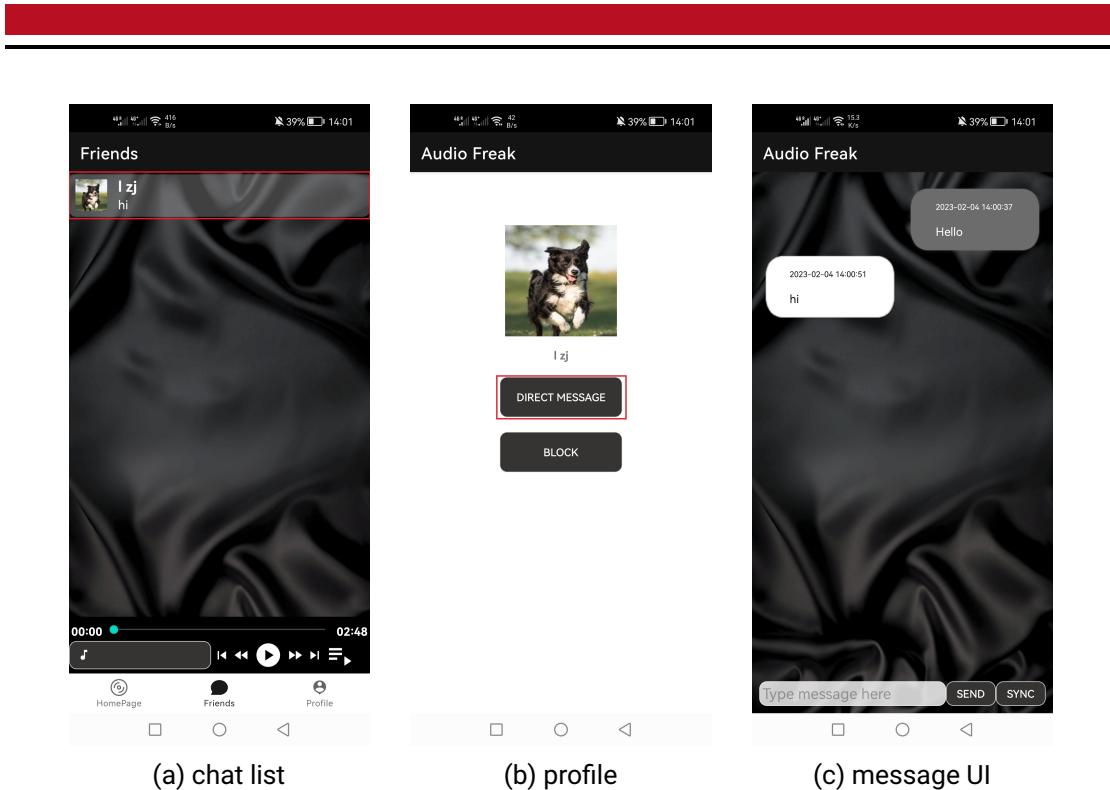


Figure 3.13: direct message

3.4.2 Message

By clicking send button users can chat with others. If the users want to listen to music together, they can click the sync button to send an invitation for synchronized playback. Users can enter the words here and click "send" button (in 3.14a), the message will be sent (in 3.14b). If the user has received the message, the system will give a reminder (in 3.14c) and a notification (in 3.14d).

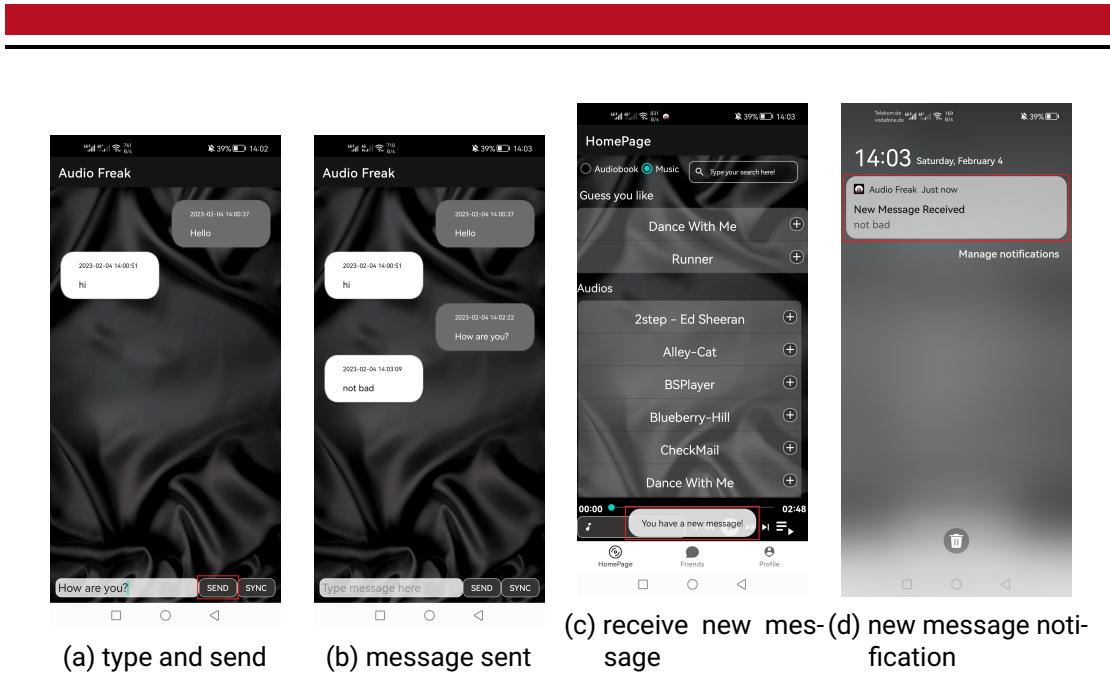


Figure 3.14: send and receive message

3.4.3 Block

Click the notification directly, it will jump to the profile of that user. Then users can decide chat with that user or block (in 3.15a). If users are blocked by a user, they will be no longer able to send message to that user (in 3.15b).

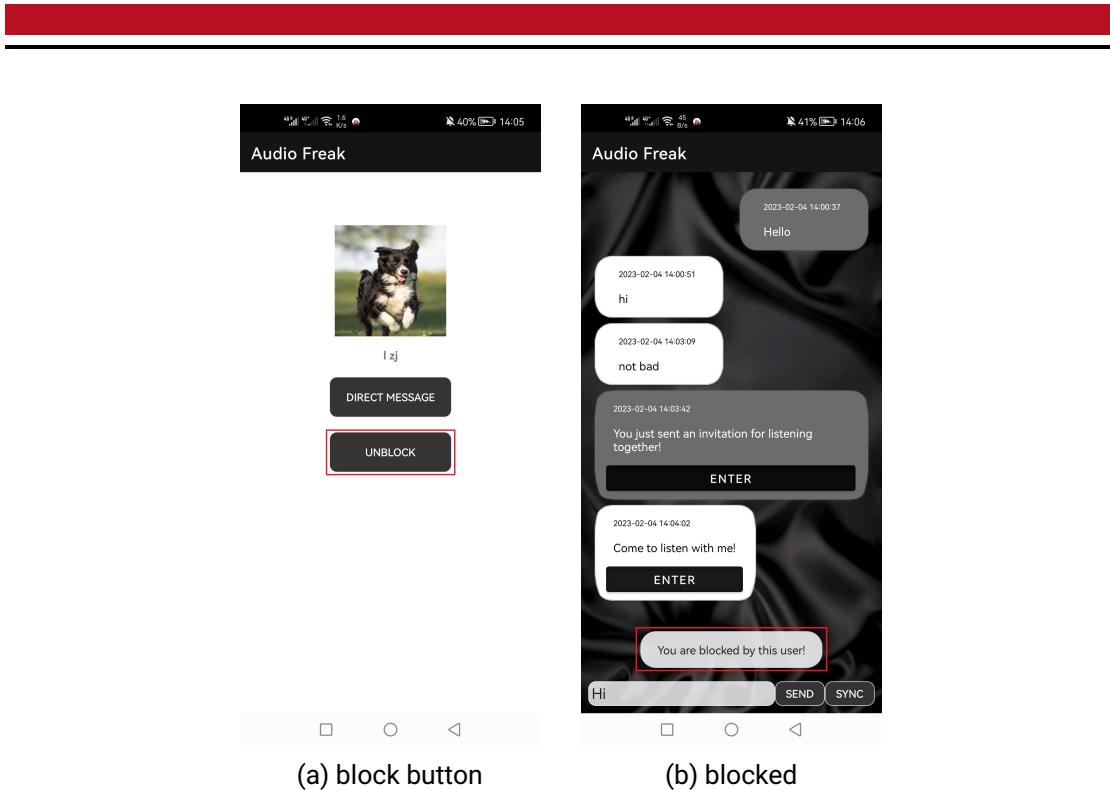


Figure 3.15: block

3.4.4 Invitation for synchronized playback

Users can send synchronized playback invitation by clicking "sync" (in 3.16a). When they received an invitation for synchronized playback, they can click "enter" to listen together (in 3.16b).

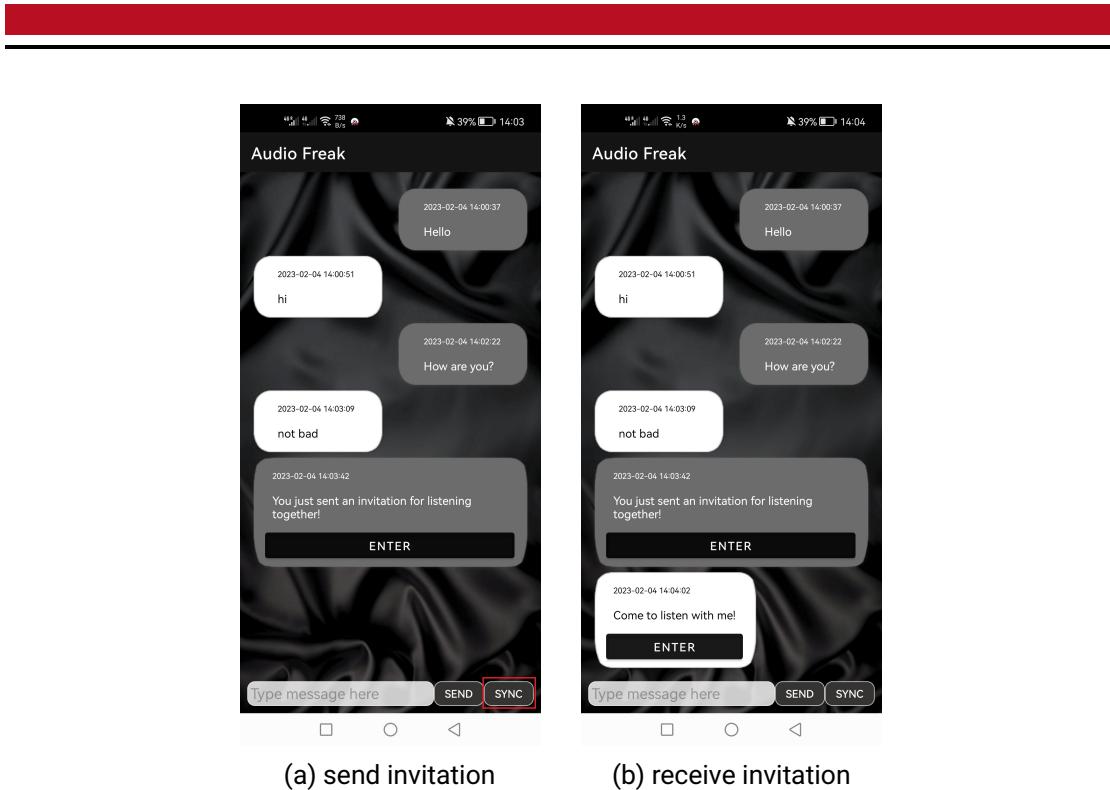


Figure 3.16: invitation for synchronized playback

3.4.5 Recall

If user wants to recall a certain message, they only need to press and hold his own message (in 3.17a), and then click "confirm" (in 3.17b). Then that message will disappear (in 3.17c).

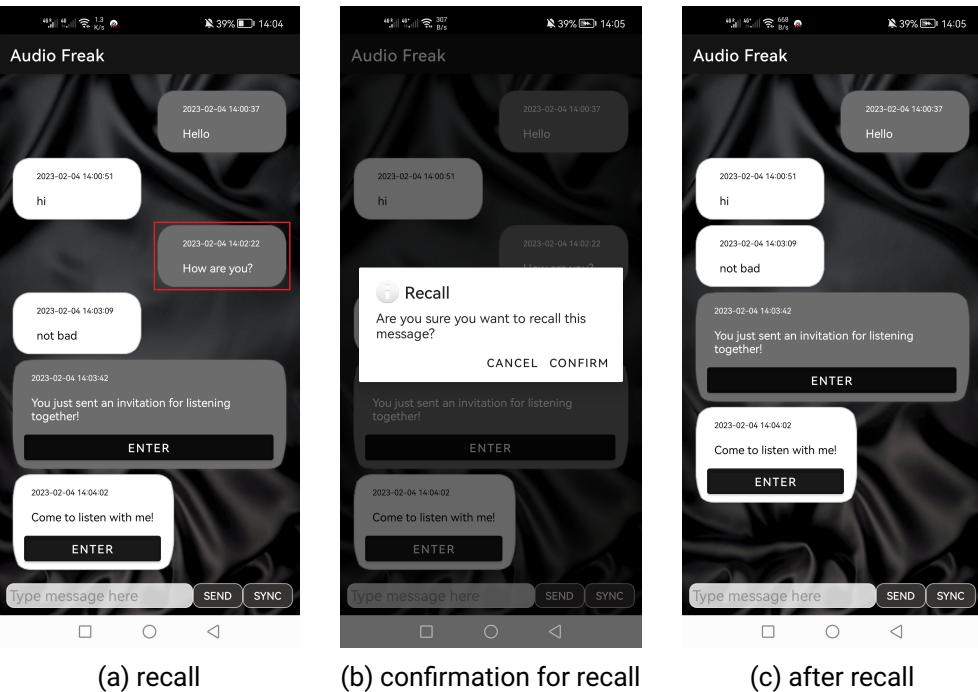


Figure 3.17: recall

3.4.6 Report

When finding inappropriate content, users can long press that information to report (in 3.18a). After confirmation (in 3.18b) the administrators will handle it (in 3.18c).

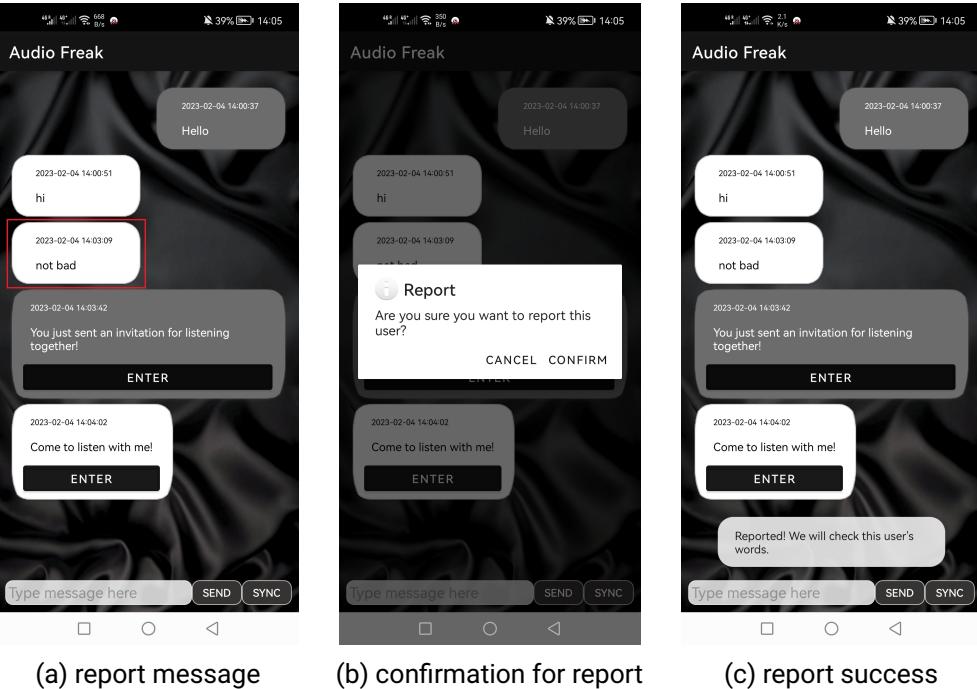


Figure 3.18: report

3.5 Synchronized Playback

The synchronized playback feature can be used when users want to share a favorite song or audiobook with a friend, especially if the user wants the friend to listen to the highlights specifically, or if they want a private playback room away from the public to enjoy music/audios with a friend in peace.

3.5.1 Enter the synchronized playback room

As shown in Figure 3.19a, in the private message window, click the SYNC button in the bottom right corner to send an invitation to a friend to join a synchronized playback room, and the inviter enters the room. As shown in Figure 3.19b, the other user receives the invitation and clicks the ENTER button to enter the room. After both users enter the room,



when one user plays, pauses, adjusts progress, fast-forwards and backwards, switches songs/audiobooks, the other user's player is synchronized in real time.

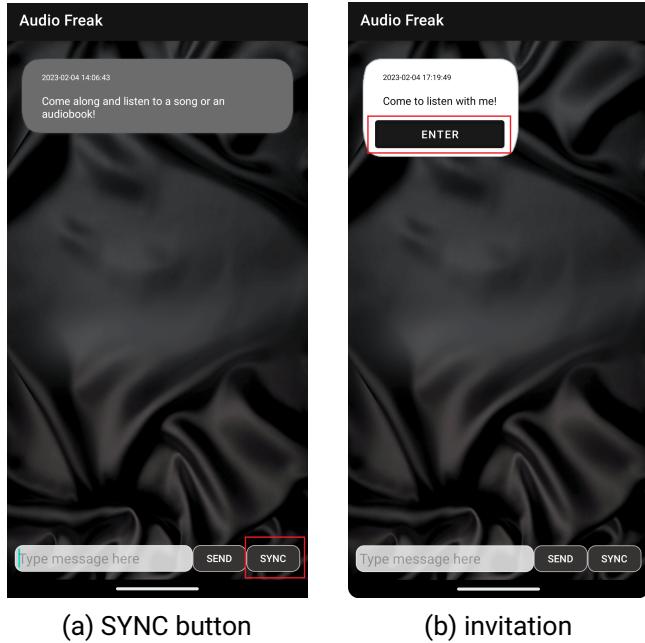


Figure 3.19: Invitation to synchronized playback room

3.5.2 Components and Functions

The synchronized playback room consists of four main parts. At the top is a notification message, if the invited person hasn't come in yet, the notification will show that 'Friend has not come :(' (in 3.20a). As the figure 3.20b shows, after the invited person comes in, the notification will be updated to show that 'Friend come'. There is a disc below the notification. When the song/audiobook is playing, the disc rotates, and when the song/audiobook is paused, the disc is stationary. In addition, click on the disc will jump to the media list, where user can freely select and switch songs/audiobooks (in 3.20c). In the middle is a player similar to live chat room, with functions of play, pause, adjust progress, forward and backward. At the bottom is a place to post comments with friends in real-time, where you can share your opinions on songs/audiobooks or remind others of upcoming highlights.

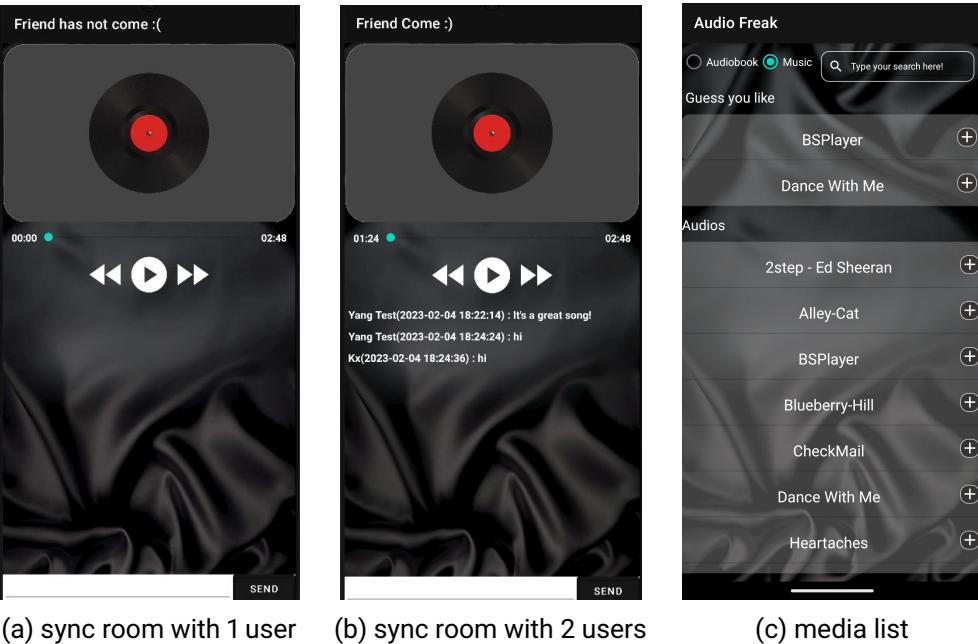


Figure 3.20: Synchronized playback room

3.6 Profile

Users can modify profile, reset password, upload audio file and sign out here (in 3.21).

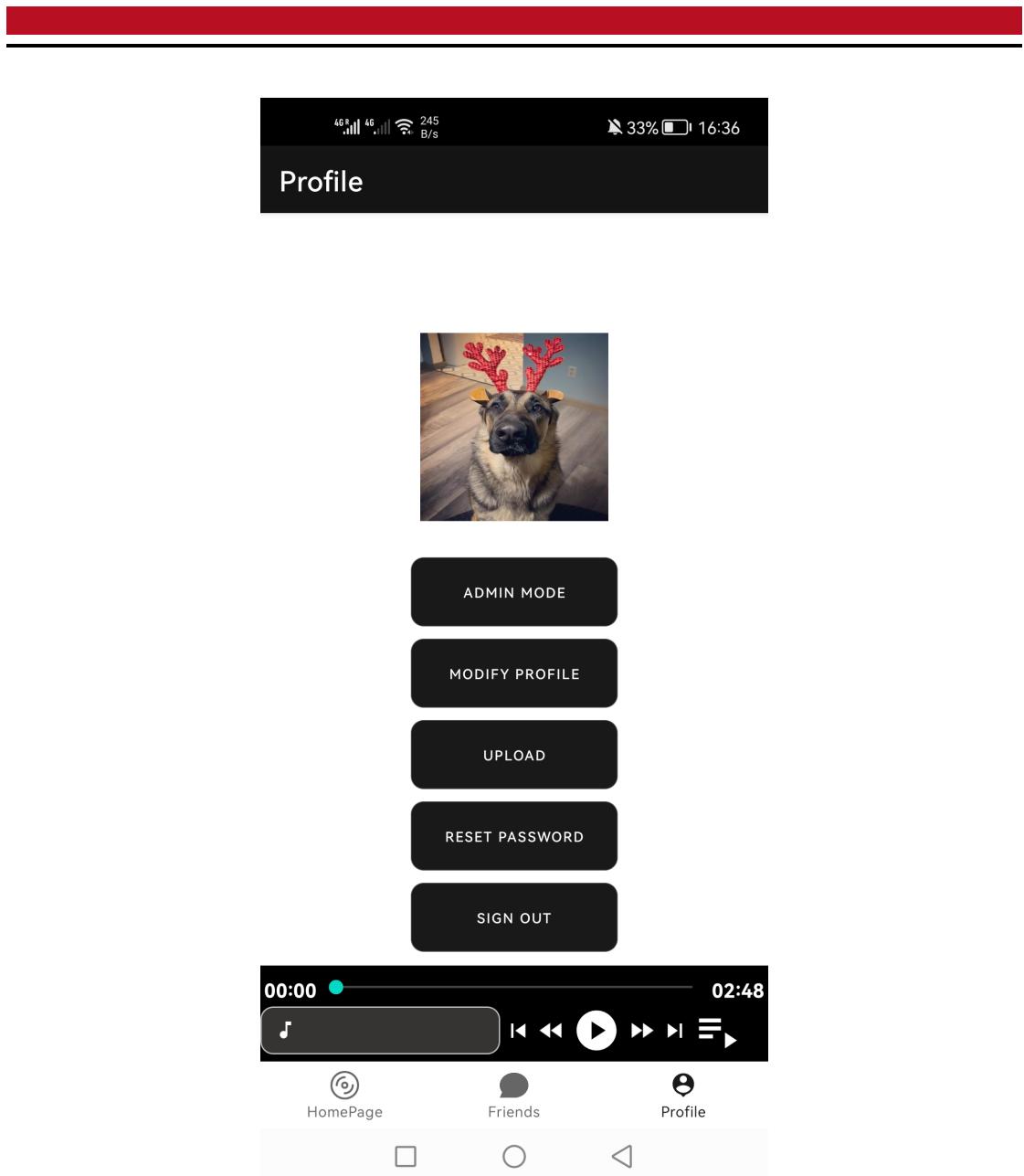


Figure 3.21: profile



3.6.1 Profile modify

Users can change avatar by clicking current avatar (in 3.22a). Then select a picture from device (in 3.22b) and upload (in 3.22c).

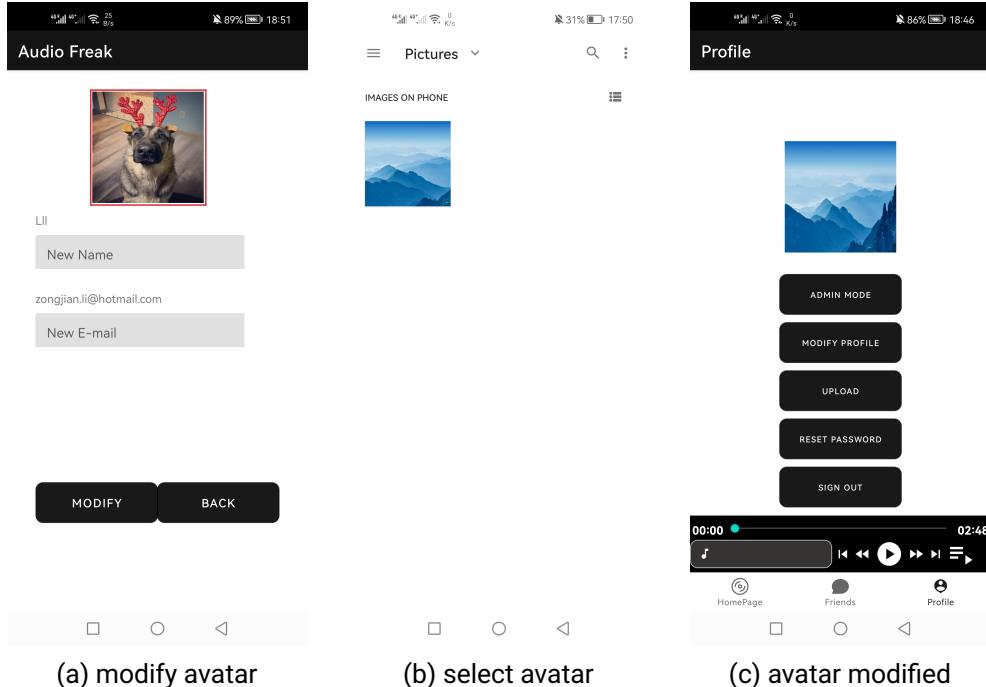


Figure 3.22: modify avatar

After enter the new data for modification (in 3.23a), users can click "modify" button and then get a Pop-ups for confirmation (in 3.23b). After confirmation for modification of E-mail address (name has been changed in 3.23c), profile will be modified(in 3.23d).

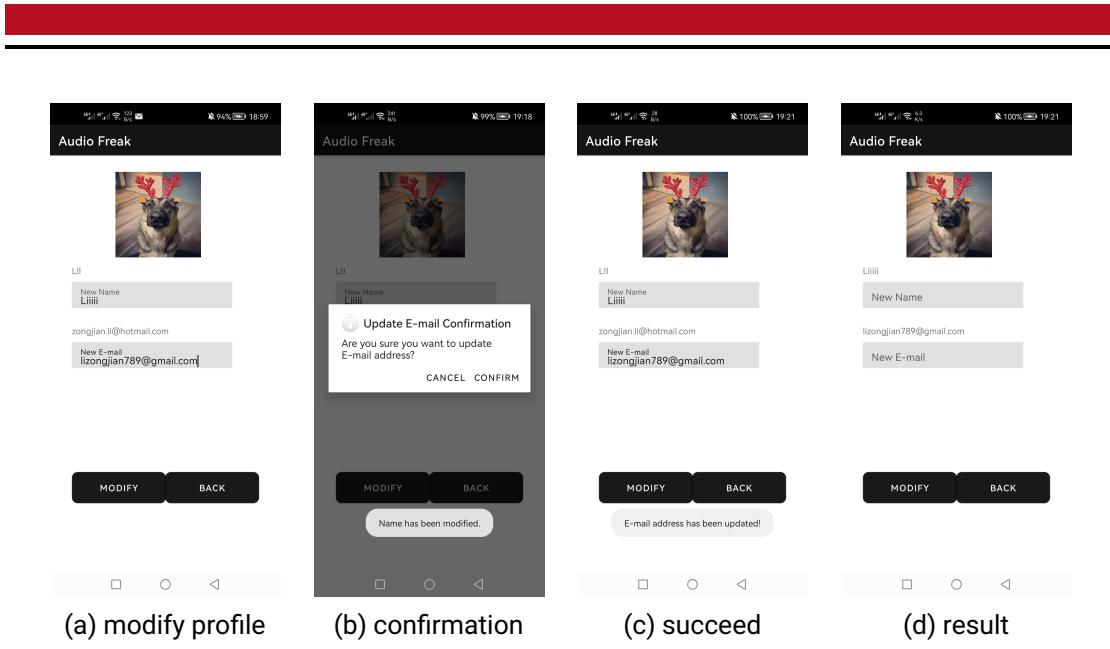
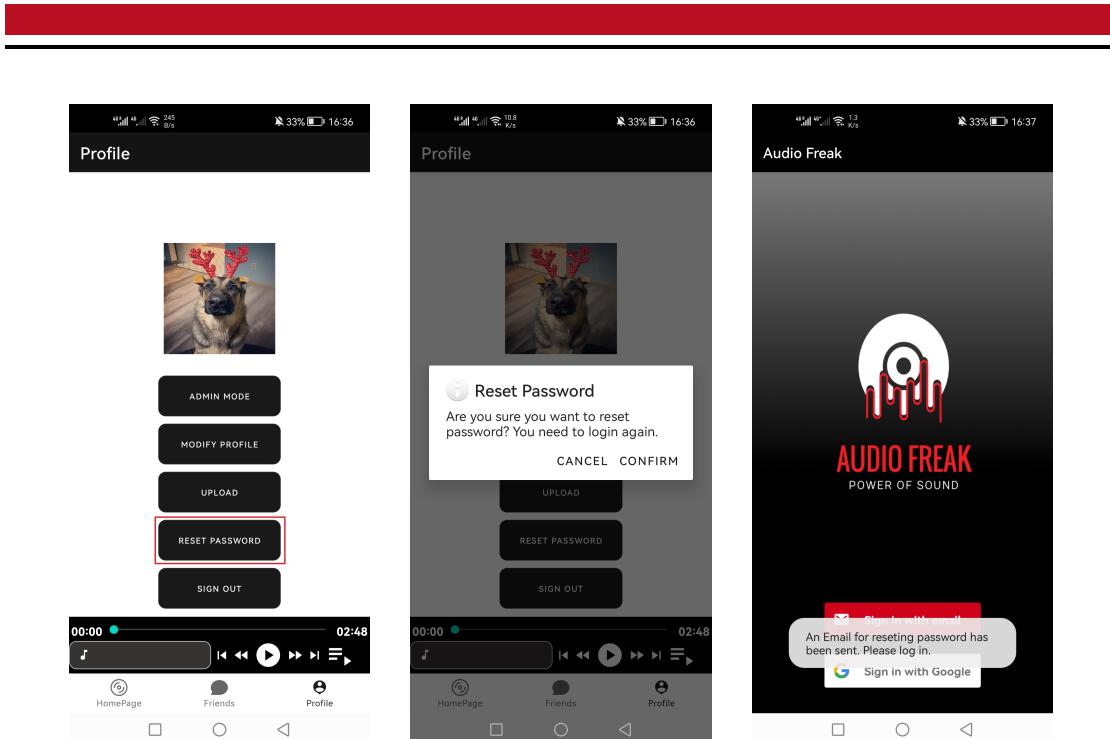


Figure 3.23: modify name and E-mail address

Users can reset password by clicking "reset password" button (in 3.24a). After clicking the button, a confirmation box will pop up (in 3.24b). After confirmation, it will be automatically logged out (in 3.24c), and a confirmation email for resetting the password will be sent to the current E-mail address at the same time (in 3.25a). In the link from this email the user can reset the password (in 3.25b).



(a) reset password button (b) Pop-ups for confirmation (c) automatically log out

Figure 3.24: reset password 1

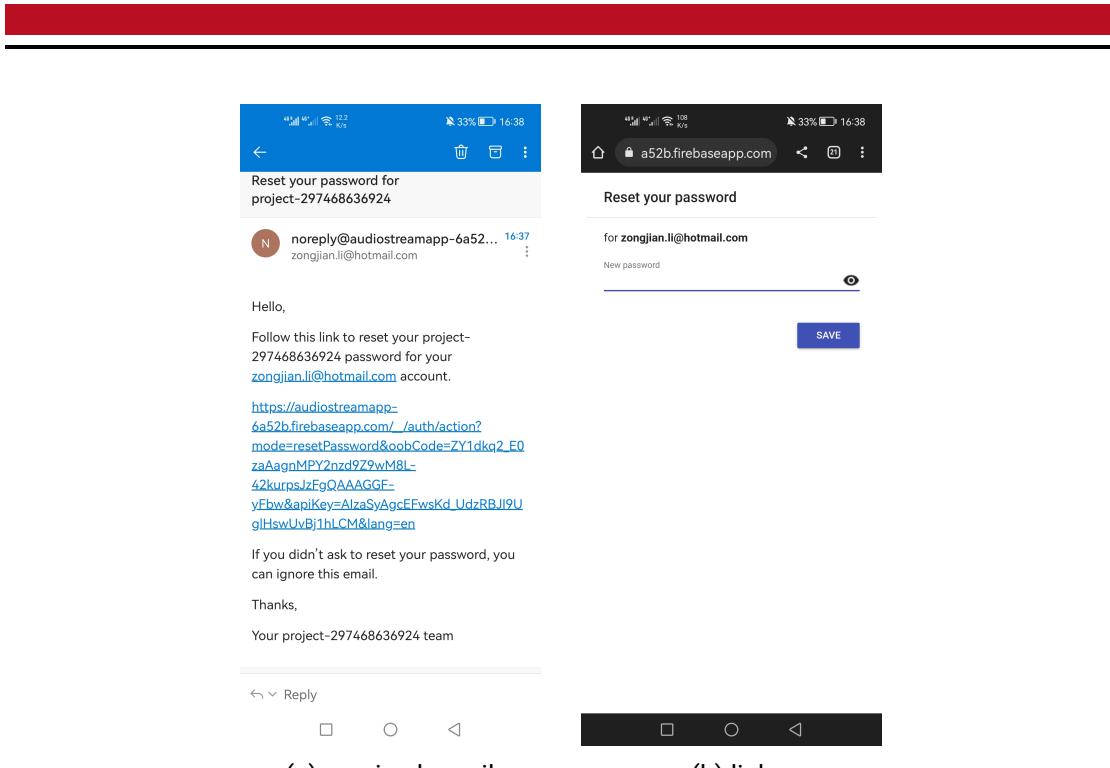


Figure 3.25: reset password 2

3.6.2 Upload

Users can upload music or audio books by clicking "upload" button (in 3.26a). Then users need to select a audio file (in 3.26b) and choose its type (in 3.26c), then it will be uploaded. Now users can search for this audio file (in 3.26d).

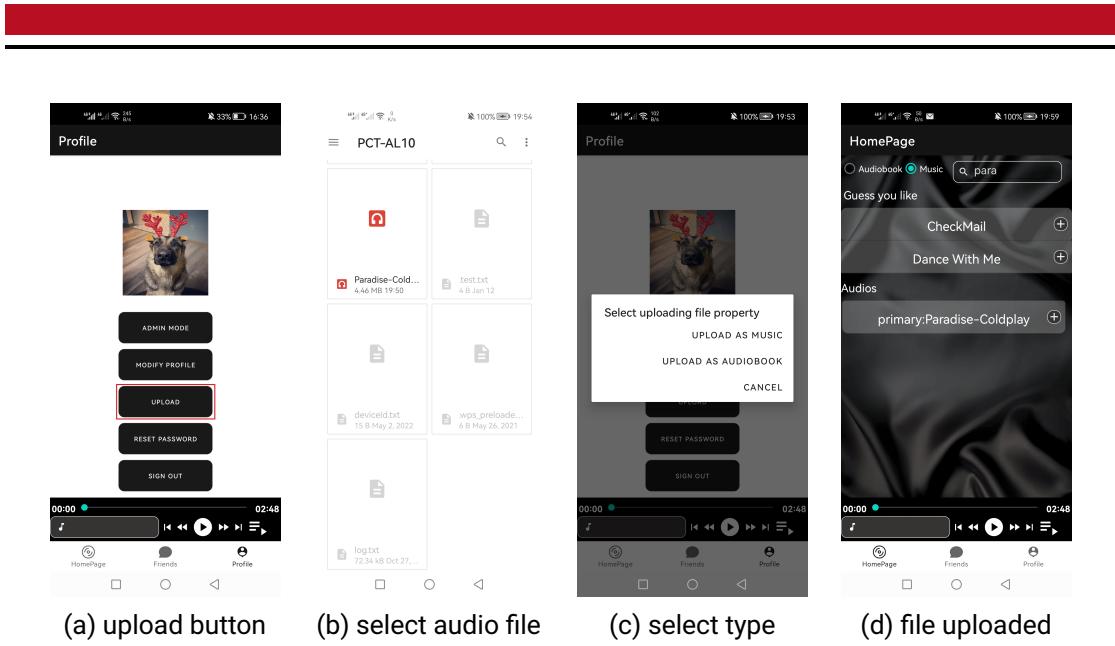


Figure 3.26: upload

3.7 Administrative Access

The live chat room is set up to filter malicious comments and block some swear words. However, some users may be using non-offending words or combinations of words that fit into an implicit swear word that the system may not be able to recognize and thus display, so the user can report the comment and the administrator will check the reported content, and review it manually.

As shown in the figure 3.27a, enter the admin mode from the ADMIN MODE button on the profile page, after that a confirmation window will pop up (in 3.27b), if it is not an admin account then there is no permission to enter (in 3.27c), as shown in the figure 3.28a, if it is an administrator account, then there is permission to enter admin mode and view the report, which including the reported comments, user UID and current user status(if the user is blocked).

If the comment is indeed found to be profanity or insulting/inappropriate content, the administrator can block the user (of course, can also unblock the user), and the user's status will change accordingly (in 3.28b).

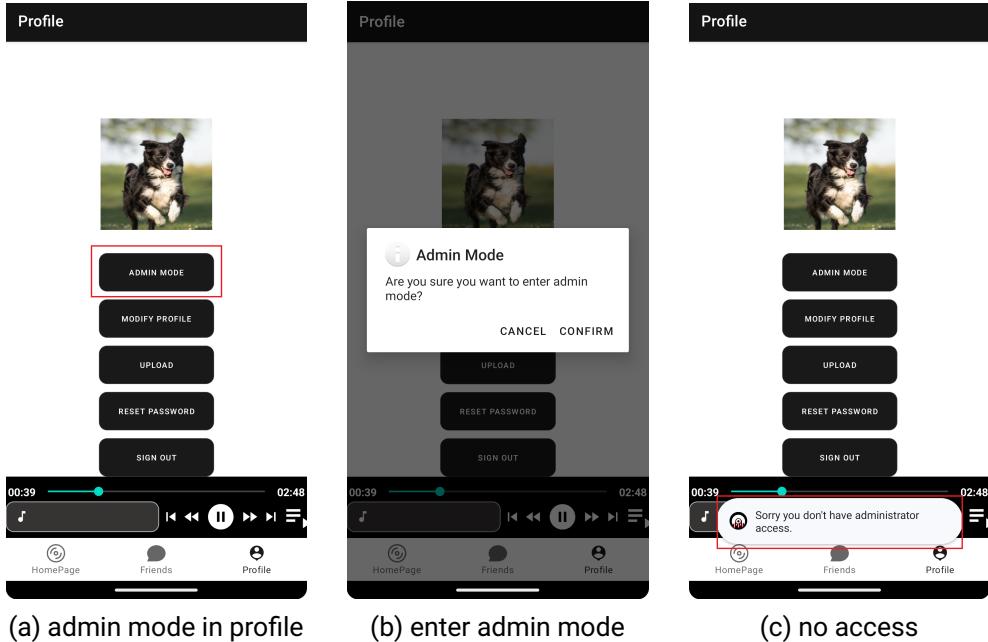


Figure 3.27: Administrative access

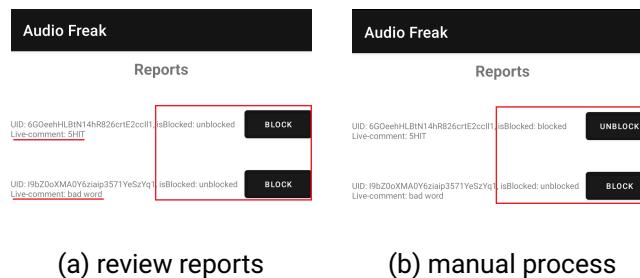


Figure 3.28: Review and process reports

4 Feature List

4.1 Main features

- Media Streaming Functionality
 - Content streamed from server
 - Playlist
 - Media upload
- Social Interaction
 - Live Chat
 - "who's watching?"
 - Direct Message
 - "Save-to-read"
 - Blocking users
 - Reporting to administrators and flag the content
- Synchronized playback
 - Live Chat while synchronized playback
- Content Recommendations
- Administrative access
 - Account managing
 - Reviewing flagged content

-
-
- Secure login
 - Notifications

4.2 Sign-up and Login

- Safe registration with email and password
- Sign in with Google account
- Storing hashed password in Firebase
- Keeping the login state between usages

4.3 Settings

- Changing avatar, username for logged-in users
- Changing password with validation of email
- Login out feature