Vault: a Video Conferencing App

An Android Project built as a part of the course

CSE583: Mobile Computing

By

Anis Mishra | 2020026

Yashasvi Chaurasia | 2020159

Garima Chopra | 2020376

Ayush Chauhan | 2020188

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY DELHI

Table of Contents

Introduction

In today's interconnected world, communication has transcended geographical boundaries, with video conferencing becoming an indispensable tool for businesses, education, and personal interactions. The proliferation of smartphones and the availability of high-speed internet have made video conferencing apps accessible to a wide range of users. In line with this trend, our project focuses on the development of a video conferencing app tailored for the Android platform.

1.1 Objectives

- Our primary objective is to create a robust and user-friendly video conferencing application that meets the growing demand for seamless virtual communication.
- Utilizing modern technologies and the rich set of features offered by the Android platform, we aim to provide a comprehensive solution for individuals and organizations seeking efficient and reliable video conferencing capabilities on their mobile devices.
- This report provides an in-depth overview of the development process, including the functionalities implemented, the technologies and tools utilized, and the challenges encountered. By adhering to the guidelines provided, we have crafted an engaging and informative presentation that showcases the various activities within the app, highlighting their functionality and user experience. Additionally, we explore the additional Android features incorporated into the app, such as accessibility options, sensing capabilities, and location services, to enhance its usability and accessibility.

Through this project, we aim to contribute to the evolution of video conferencing technology on the Android platform, empowering users to connect, collaborate, and communicate seamlessly in an increasingly digital world.

Technologies Used

2.1 Libraries and SDKs

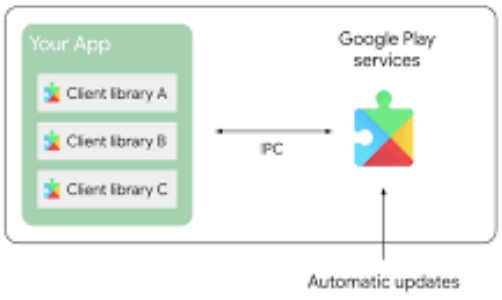
- Firebase Firestore: Employed Firebase Firestore as the backend database for storing user data, meeting information, and other app content. Firestore's real-time database capabilities ensured seamless data synchronization across devices.

- Jitsi Meet SDK: Integrated the Jitsi Meet SDK to enable video conferencing capabilities within the app. This SDK facilitated features like audio and video streaming, chat functionality, and screen sharing, providing a comprehensive video conferencing experience for users.

- Material Design Components: Utilized Material Design Components to design a visually appealing and intuitive user interface. Material Design components such as text fields, buttons, and surfaces were styled according to Material Design guidelines, ensuring consistency and familiarity for users.

2.2 APIs

- Google Play Services APIs: Integrated Google Play Services APIs, including location services, to access essential functionalities such as obtaining the device's current location accurately.



2.3 Android Features

1. Location Services

- *FusedLocationProviderClient*: Utilized the FusedLocationProviderClient class from the Google Play Services API to retrieve the device's current location.
    - requestLocationUpdates(): Requested periodic location updates using this function, specifying parameters such as update interval, fastest update interval, priority, and a callback to handle location updates.
    - lastLocation: Accessed the most recent location available to the FusedLocationProviderClient, enabling retrieval of the device's last known location when immediate updates were unnecessary.

- *LocationRequest*: Employed the LocationRequest class to define parameters for location requests, including update interval, fastest update interval, and priority.
    - create(): Created a new instance of LocationRequest with default parameters, allowing customization according to the app's requirements before requesting location updates.

2. Permission Handling

- Utilized Android's permission system to request and handle permissions necessary for accessing device features such as location services, camera, and microphone.
- *ActivityCompat.requestPermissions()*: Requested permissions from the user at runtime using this function, which takes an array of permission strings and a request code as parameters.
- *onRequestPermissionsResult()*: Overridden the onRequestPermissionsResult() callback method to handle the user's response to permission requests, ensuring appropriate actions were taken based on permission grants or denials.

3. Broadcast Receiving:

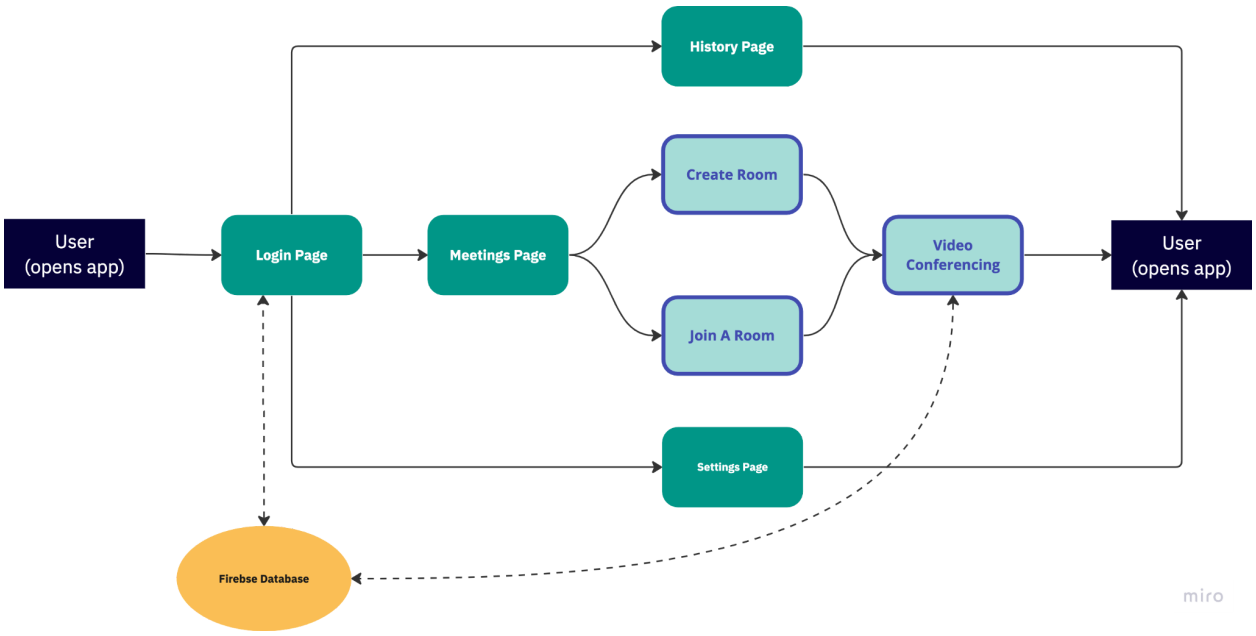
- Implemented a custom broadcast receiver using the BroadcastReceiver class to handle broadcast messages within the app, facilitating efficient communication and coordination between different components.

    - onReceive(): Overridden the onReceive() method to define the behavior of the broadcast receiver when receiving a broadcast message, including the logic to handle the message and perform necessary actions.

- *LocalBroadcastManager*: Utilized the LocalBroadcastManager class to manage broadcast messages within the app's process, confining broadcasts to the app's scope and improving security and efficiency.
    - *getInstance()*: Obtained an instance of LocalBroadcastManager using this static method.
    - *registerReceiver()*: Registered a BroadcastReceiver to receive broadcast messages with a specified IntentFilter.
    - *unregisterReceiver()*: Unregistered a previously registered BroadcastReceiver when it was no longer needed, preventing memory leaks and optimizing resource usage.

By leveraging these Android features, the video conferencing app effectively utilized location services, handled permissions, and implemented broadcast message handling, enhancing user experience and functionality.

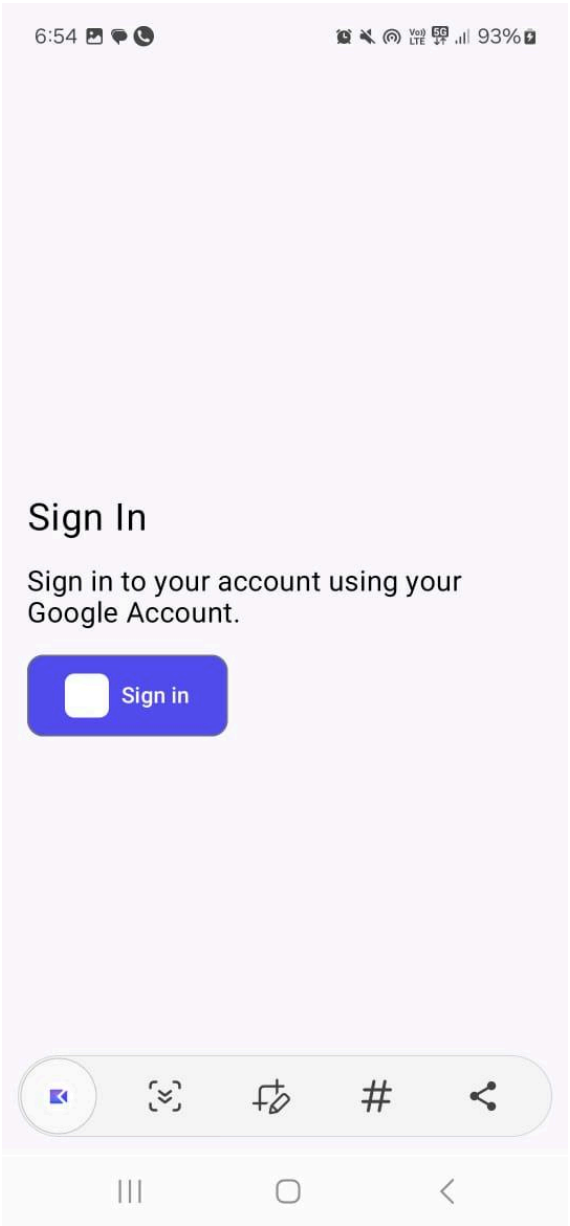
User Flow Diagram

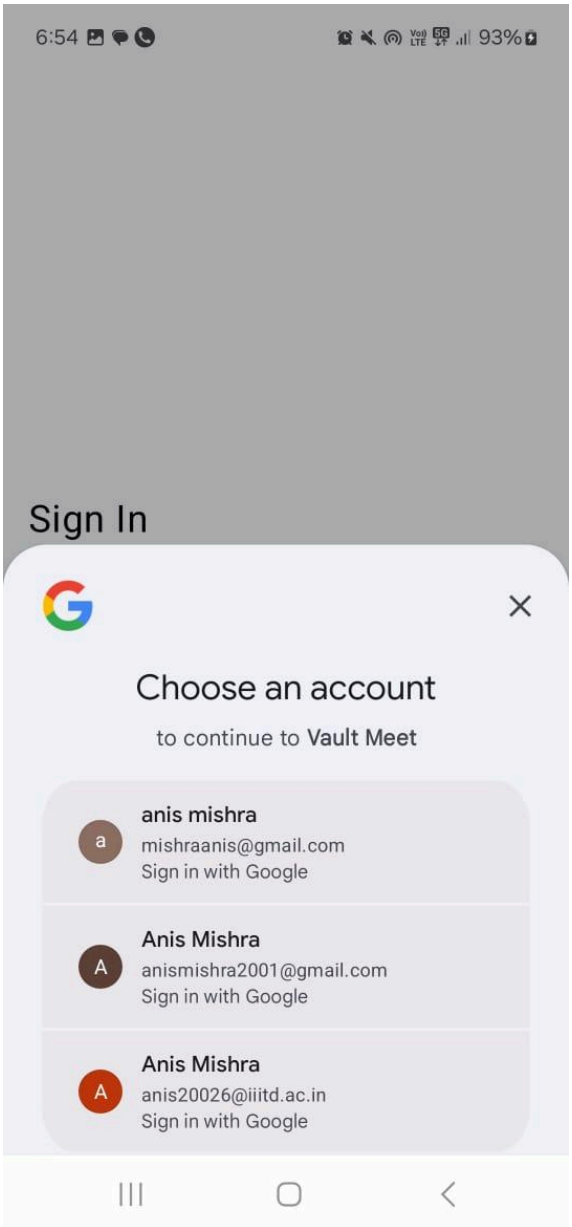


Activities and Functionality

3.1 Login Page Activity:

- The Login Page Activity serves as the entry point for users to authenticate and access the app's features.
- Functionality includes user authentication using Google Sign-In or other authentication methods, ensuring secure access to the app.
- Upon successful authentication, users are redirected to the Meetings Page Activity to explore and join video conferences.

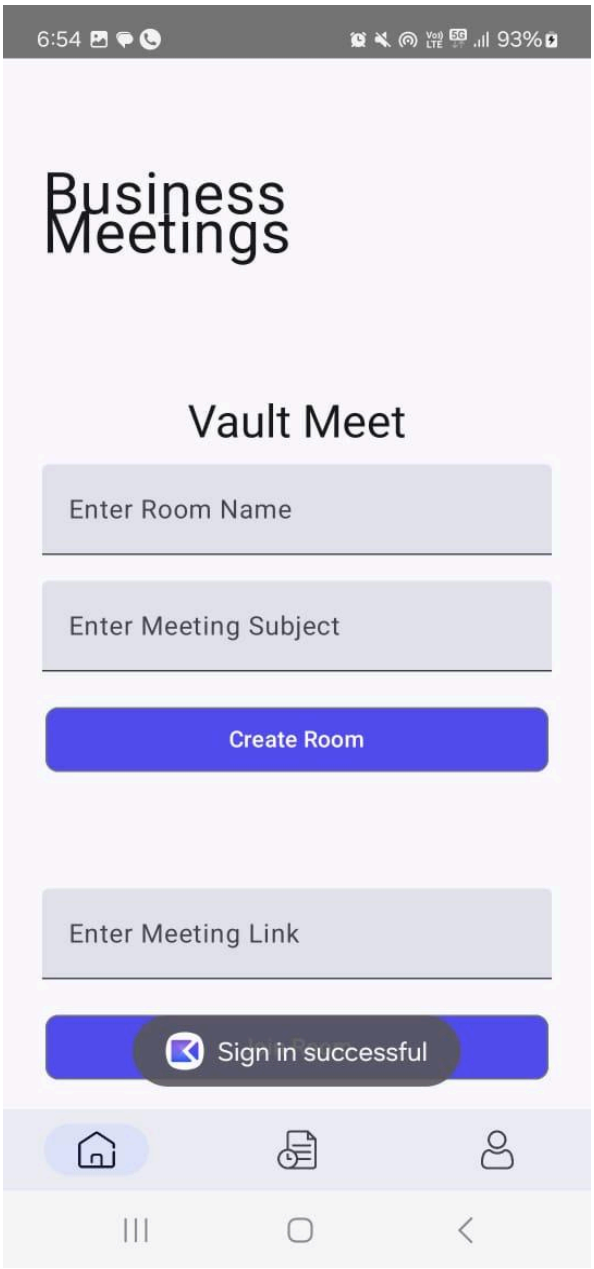


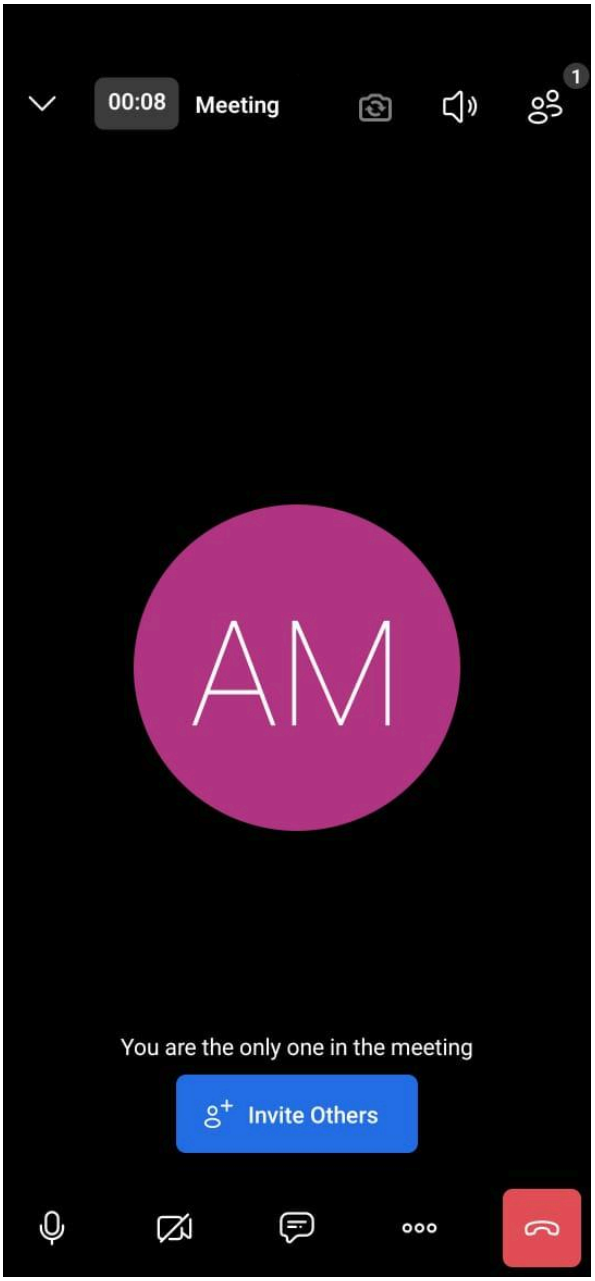


3.2 Meetings Page Activity

- The Meetings Page Activity enables users to join existing video conferences or create new meetings.

- Functionality includes:
    - Creating a new room: Users can initiate a new video conference by providing a room name and optional meeting subject.
    - Joining a room: Users can enter a meeting link or choose from a list of available meetings to join.

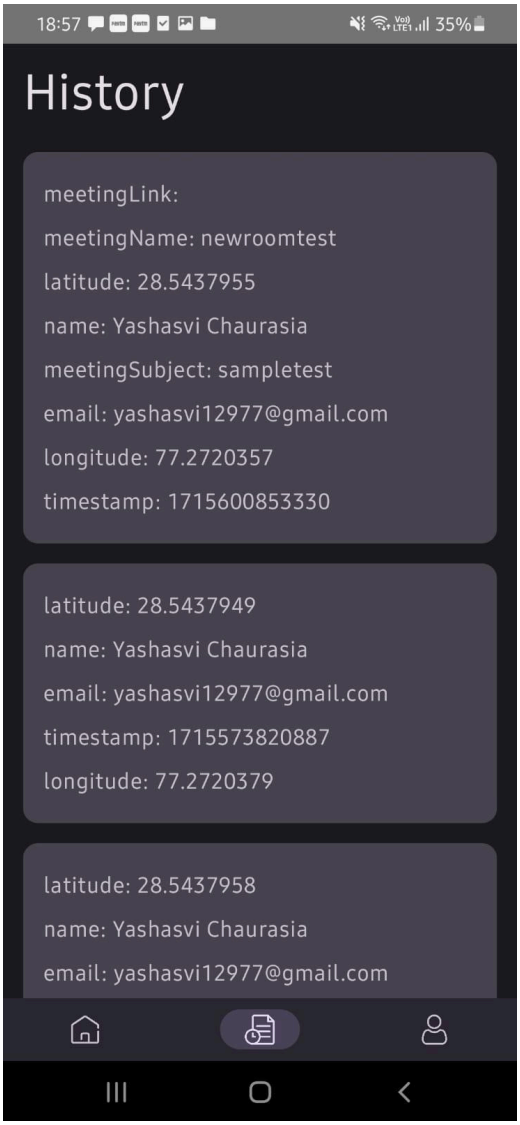




3.3 History page Activity

- The History Page Activity allows users to view their past meeting history, including details such as meeting name, participants, and timestamps.

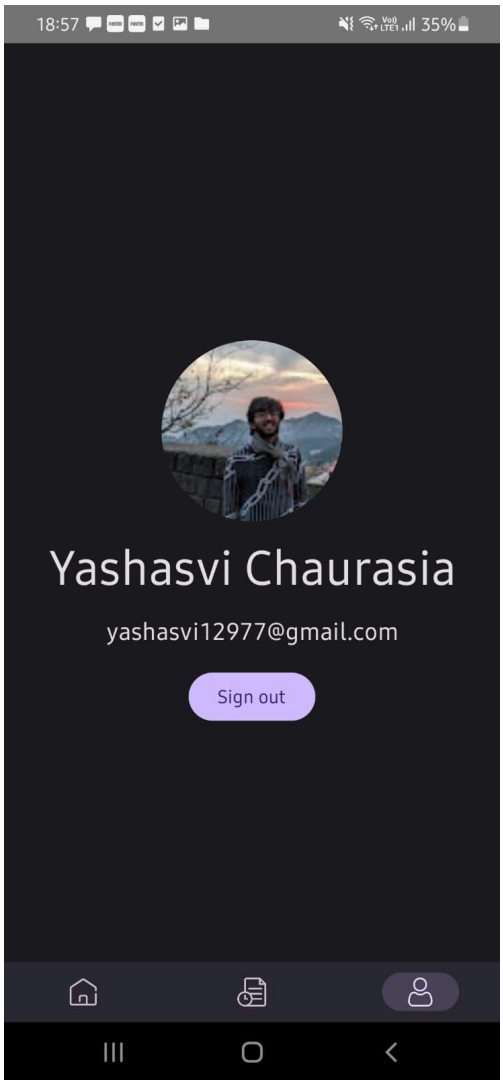
- Functionality includes:
    - Retrieving and displaying a list of past meetings from the Firebase Firestore database.
    - Providing options to view meeting details, including participants and timestamps.

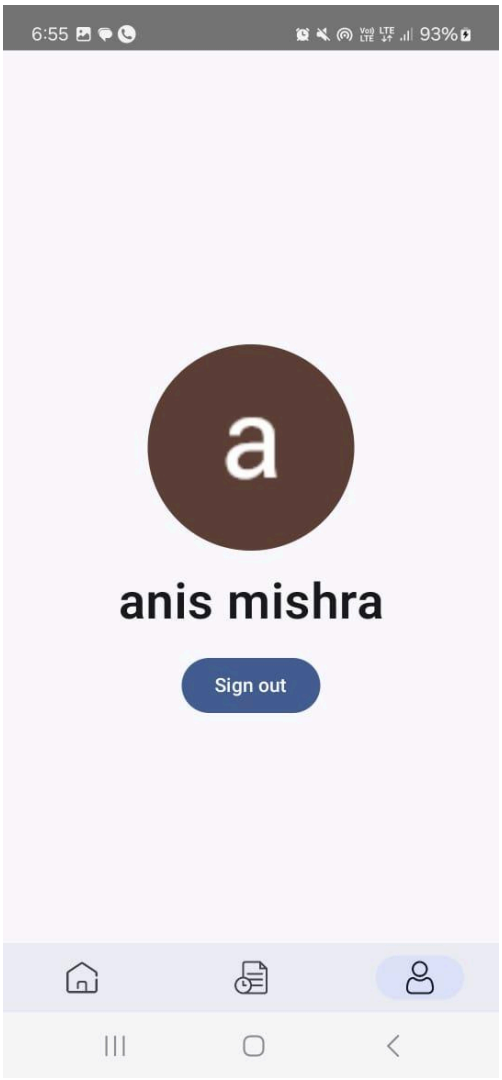


3.4 Settings/Account Page Activity

- The Settings/Account Page Activity offers users the ability to customize app settings and manage their account preferences.

- Functionality includes:
    - Accessing and updating user profile information, such as name and email address.
    - Configuring app settings, such as notification preferences and default meeting settings.
    - Managing account-related actions, such as sign-out or account deletion.





Conclusion

The development and integration of various components and features within the video conferencing app have resulted in a robust and user-friendly platform. Leveraging Firebase Firestore as the backend database ensured seamless data synchronization, while the integration of the Jitsi Meet SDK provided comprehensive video conferencing capabilities. The utilization of Material Design Components contributed to an intuitive user interface, enhancing user experience. Additionally, the incorporation of Google Play Services APIs facilitated access to essential functionalities such as location services, enriching the app's functionality. Through the implementation of Android features like permission handling and broadcast message handling, the app effectively managed user permissions and facilitated efficient communication between components. Furthermore, the diverse range of activities and functionalities offered, including the Login Page Activity for secure authentication, the Meetings Page Activity for joining or creating conferences, the History Page Activity for reviewing past meetings, and the Settings/Account Page Activity for customization, cater to the diverse needs of users, ensuring a seamless and engaging experience. Overall, the meticulous design and implementation of these elements have culminated in a feature-rich video conferencing app poised to meet the demands of modern communication.