Available Online at www.ijcsmc.com/journal

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 11, Issue. 4, April 2022, pg.61 – 68

Mobile Cloud Computing: Using Firebase Auth

Ishpreet Kaur
Department of Information Technology, Guru Tegh Bahadur Institute of Technology, New Delhi, India
ishpreet95kaur@gmail.com

DOI: https://doi.org/10.47760/ijcsmc.2022.v11i04.008

Abstract— Mobile Cloud Computing (MCC) is the combination of Mobile Computing and Cloud Computing. Mobile Cloud Computing can be used on different types of devices. It is not meant for a fixed platform which means applications created using MCC do not need a permanent storage area in devices. Storage and processing of data are done in the cloud, so it requires good internet connectivity and provides great support for online activities. This paper provides a case study of a video conferencing app known as Virtual Training Room (VTR). VTR uses Firebase Authentication which is an example of MCC. We will study the functioning of Firebase Auth. We will get to know about the advantages, challenges, and future scope of Firebase Auth. Code for the app is monitored and analysed through the Google Cloud Platform (GCP). Jetpack Libraries are used to reduce the complexity of the code and Jitsi SDK is used for the calling purpose.

Keywords— Mobile Cloud Computing, Firebase Authentication, Google Cloud Platform, Jetpack Libraries, Jitsi SDK

I. INTRODUCTION

In the contemporary scenario, Mobile Cloud Computing (MCC) is used by almost every application developer. Mobile Cloud Computing provides features like remote access to data that is data can be accessed anywhere anytime. MCC provides a huge amount of space because Cloud Computing uses data centers that are present in the Cloud Server. Google acquired Firebase in the year 2014. Since then, Firebase users are increasing day by day. Firebase uses Cloud Computing features like Auth, Cloud Functions, Cloud Firestore, Cloud Storage, messaging, Machine Learning, etc. for the purpose of building better applications. Firebase is mainly used for creating real-time apps. Firebase provides real-time synchronization which means developers can connect Firebase within their app and if any change is made in the application, that data automatically gets synchronized with the Firebase Cloud within milliseconds. Firebase apps need client-side coding only because server-side coding is handled by the Firebase database server. Data is stored in JSON format in Firebase. URL is created for every piece of data which helps to view the real-time modification of data present in Firebase libraries. Fig. 1 depicts the growth of monthly active Firebase applications from the year 2014 to 2021.

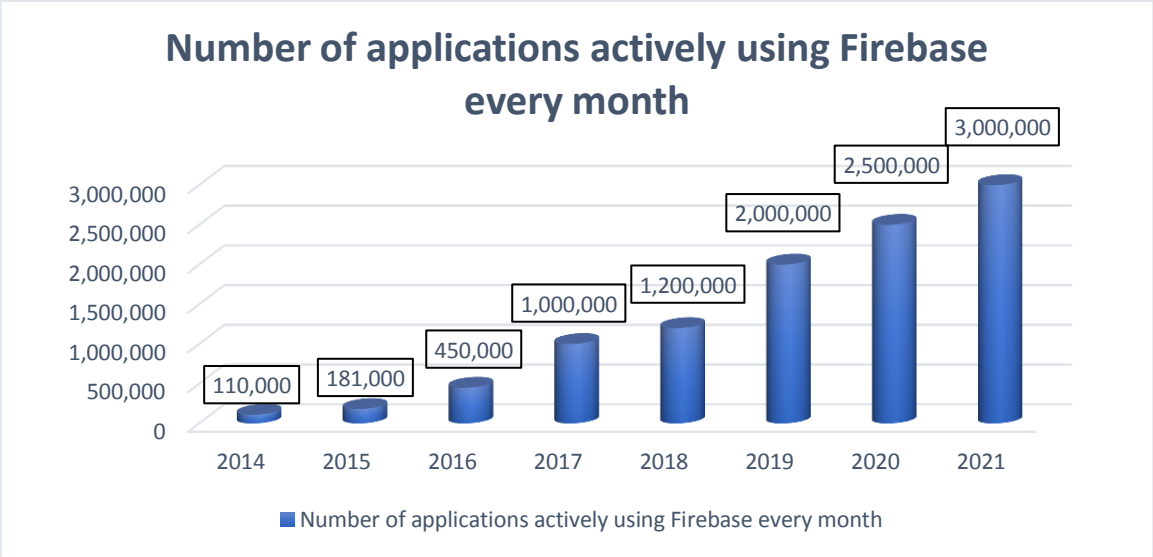

Fig. 1 Growth of monthly active Firebase applications from the year 2014 to 2021

Firebase launched Firebase Auth in 2014. Firebase Authentication provides server-side support by authenticating users who are using client-side UI. We can use authentication features like Signup and Login using E-mail Id, phone number, or different identity providers like Google, Facebook, Twitter, etc. Firebase Auth [1] provides an account recovery feature by sending password reset emails. It also provides different backend services, ready-made UI libraries, and many SDKs. Firebase Auth user's data gets stored in IndexedDB named as firebaseLocalStorageDb. If firebaseLocalStorageDb gets deleted either accidentally or intentionally, it will lead to the removal of all login user details from Firebase Auth. To test the app locally, Firebase Auth provides an Authentication Emulator which is part of the Firebase Local Emulator Suite.

II. RELATED WORK

As we have seen in Fig.1 that the number of applications using Firebase is increasing exponentially. So, there is a huge number of applications present in Playstore that uses different Firebase features. Fabulous [2] is one of those apps that use Firebase Auth. Fabulous selected Firebase Auth because Firebase Auth met all the 3 key requirements needed by Fabulous. These requirements were:

- To check the app the first day and to sign-up the next day after going through all the features of the app.
- To sign-up using different providers that is by using Google Sign-In, Facebook Login, and/or Email Password Sign-up methods.
- To preserve the UI or looks of the onboarding pages.

As per Comparitech [3], In 2020 there was a security breach in thousands of apps due to misconfigurations on Firebase databases. Unauthorized parties were able to easily access users' personal data. According to this report, around 24,000 apps published in Playstore leaked sensitive information. The exposed data included E-mail addresses, usernames, passwords, phone numbers, chat messages, IP addresses, credit card numbers, photos of government-issued Identification cards, etc. Most of the exposed databases provided write permissions to attackers, which allowed them to view, download, add, modify, or even remove data from the server. Write access allowed attackers to:

- Inject data into an application
- Phish and scam user's application
- Easily spread different malware
- Corrupt the entire application database

Attackers were able to easily find and steal data from the storage by appending ".json" at the end of a Firebase URL. These database URLs worked with other search engines except Google. Comparitech team searched exposed databases by using ".firebaseio.com" at the end of each application resource. Firebase provides REST API to access saved data. All the data was saved in JSON format, so by making a request to the database URL appended by ".json", public databases were accessed. This request returned the entire data present in the database if the database was publicly exposed. Otherwise, it returned an "access denied" message. Researchers searched for different patterns created by sensitive information such as passwords, email

addresses, secret tokens, phone numbers, and many more for the purpose of analyzing data stored in exposed databases. Empty databases were eliminated from the search results and all the leaked data was destroyed.

This paper provides steps to avoid security breaches by properly implementing Firebase Auth features. From the above 2 real-life cases, we got to know the pros and cons of using Firebase Auth. To get the proper implementation steps of Firebase Auth, Firebase Auth features are used in VTR (Virtual Training Room) app.

III. DESIGN OF PROPOSED SYSTEM

The app starts with a splash screen that contains the app name "Virtual Training Room". After the splash screen, the user can see swipe screens. These screens are used to display the features of the app. Users can see the 'Let's connect' button at the bottom of these screens. After clicking on this button, Login Screen will get open. Users need to first register themselves in the app using the Signup button present on the Login Screen. After successful registration, the user can then log in to the app. Firebase Authentication helps in Login, Signup, Sign Out, and Forgot Password functionality. In case the user forgets his password, then he will have to click on the Forgot Password button present at the Login and Signup screens. Instructions will be sent to the user's registered email ID in order to change the password. Screenshots of the Firebase Auth functionality used in the VTR app are shown in Fig. 2 and Fig. 3 [4].

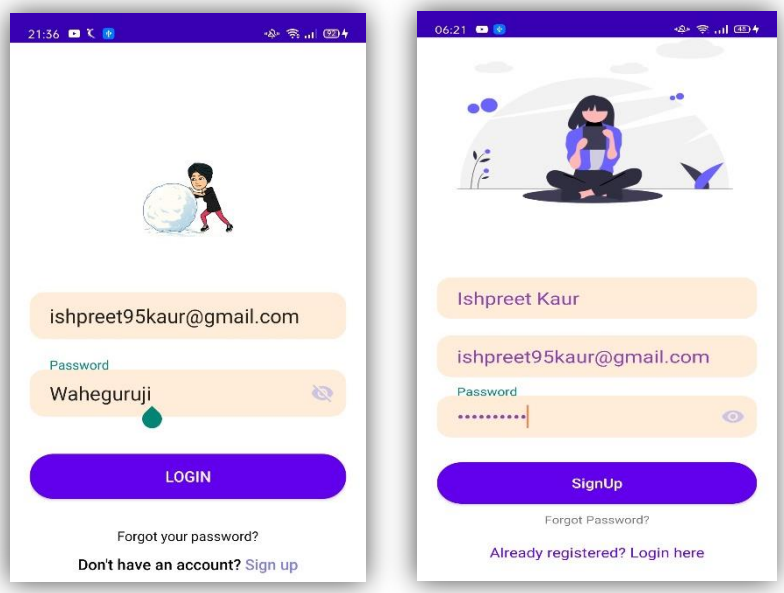

Fig. 2 Screenshots of Login and Signup pages of the VTR app

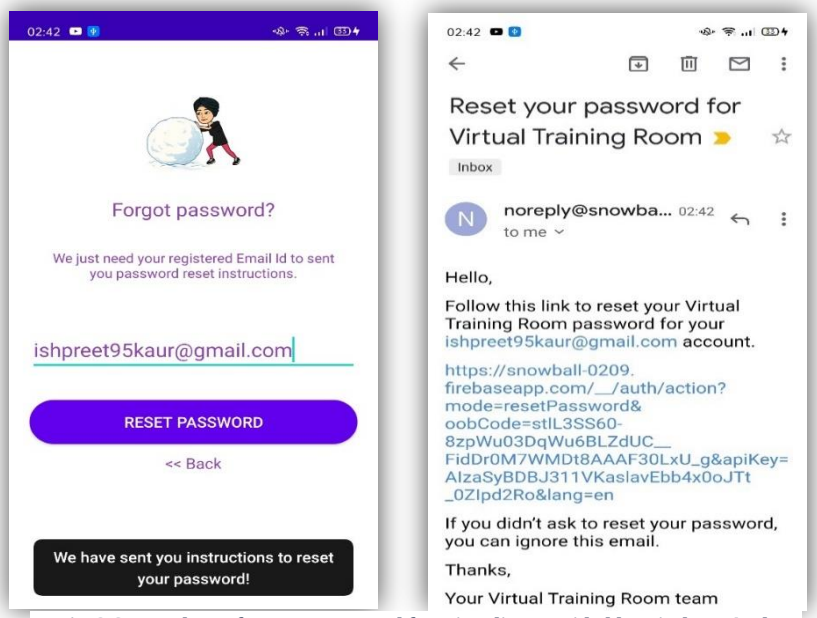

Fig. 3 Screenshots of Forgot Password functionality provided by Firebase Auth

After successfully logging in, users can see Home Screen which contains different rooms. Users can select any room according to their wishes. After getting entered the room, users can find the meeting code. They can use this code to join the meeting and can share the code with others also using the Share button present in that room. After clicking on 'Join a Meeting', the video call will get started. Users have to share the code with others in order to have more members in the call. Users can find different features present on the video call screen. Users can record the meeting, share the screen, chat with each other, raise their hands (in case of any query), use lobby mode (which allows only approved members to enter the meeting room), toggle the camera, etc. If the user wants to create his own meeting code, then he can either write meeting code in the box present on the Home Page or he can use the 'Join with a code' button present at the Side Navigation Bar of the Home Page. To start the video call, the length of the meeting code created should always be greater than 6, this feature provides the security of the code. In case of any help is needed, users can use the 'Get Help' button present at the Side Navigation Bar of the Home page. If the user wants to get any information about the developer, then he has to click on the 'About us' button present on the Side Navigation Bar of the Home Page. In case of a connectivity issue, a message 'Please check your internet connection' will get displayed on the screen. If the user wants to sign out of the app, then he has to click on the 'Sign Out' button present on the Side Navigation Bar of the Home page. A diagrammatic view of the Virtual Training Room (VTR) app is shown in Fig. 4.

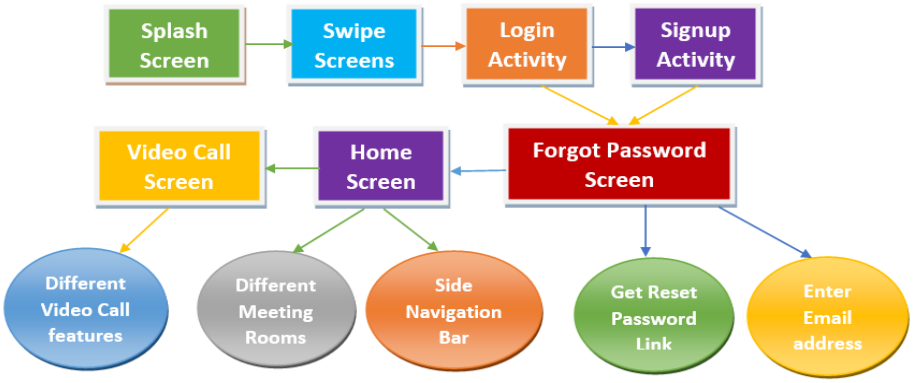

Fig. 4 Diagrammatic view of the VTR app

IV. IMPLEMENTATION DETAILS

There are many video conferencing apps like Google Meet, Microsoft Teams, Zoom, Skype, Webex Meet, etc. The limitation of these video conferencing apps is that they don't provide a feature of selecting our own theme for the meeting room. The aim of Virtual Training Room (VTR) is to provide different themes for different rooms which means users can select the theme of their meeting room according to their wishes or mood.

For the implementation of VTR, Android (Java and XML) is used. Firebase Auth is used for authentication and Firebase Firestore is used for storing user data in the database. A screenshot of Firebase Auth users of the VTR app is shown in Fig. 5. SQLite is used for creating unique meeting codes. Android provides inbuilt functionality of SQLite that is we just have to write SQLite code and it will get implemented automatically, there is no need for external setup. Jitsi SDK is used for the calling purpose. Jetpack Libraries [5] are used by adding Jetpack toolkit dependencies.

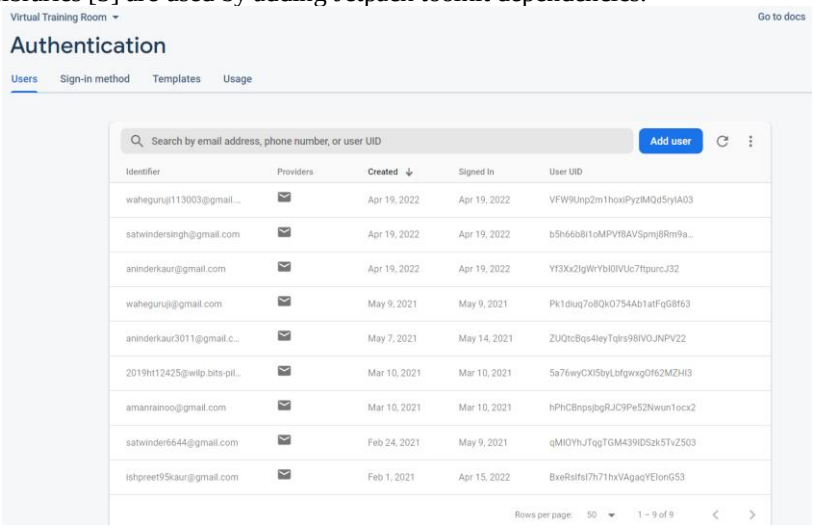

Fig. 5 Screenshot of Firebase Authentication Users of the Virtual Training Room app

Steps to properly implement firebase on the android app [6]:

- **A.** Add firebase authentication dependency in the Module Gradle file.
- **B.** Declare and initialize Firebase Auth instance.
- **C.** Check if a user is logged in or not. Steps for implementing Firebase Auth in Login, Forgot Password, and Signup screens are shown in Fig. 6.
- **D.** If a user is already logged-in, then use signInWithEmailAndPassword(email_id, password) method.
- **E.** If a user is new to the app, then he/she needs to create a user account with Email and Password. Before creating a new user account, the user's information needs to be stored in the database and for storing the user's information Firebase Firestore is used. So, the Firebase Firestore instance is created. After creating the Firebase Firestore instance, a user account will get created by using the createUserWithEmailAndPassword(email, password) method.
- **F.** After successful sign-in, the user can get his/her account data by using getCurrentUser method.
- **G.** If any user forgets his/her password then he/she has to use the sendPasswordResetEmail(email) method. This method will a send password reset email to the user's registered email.

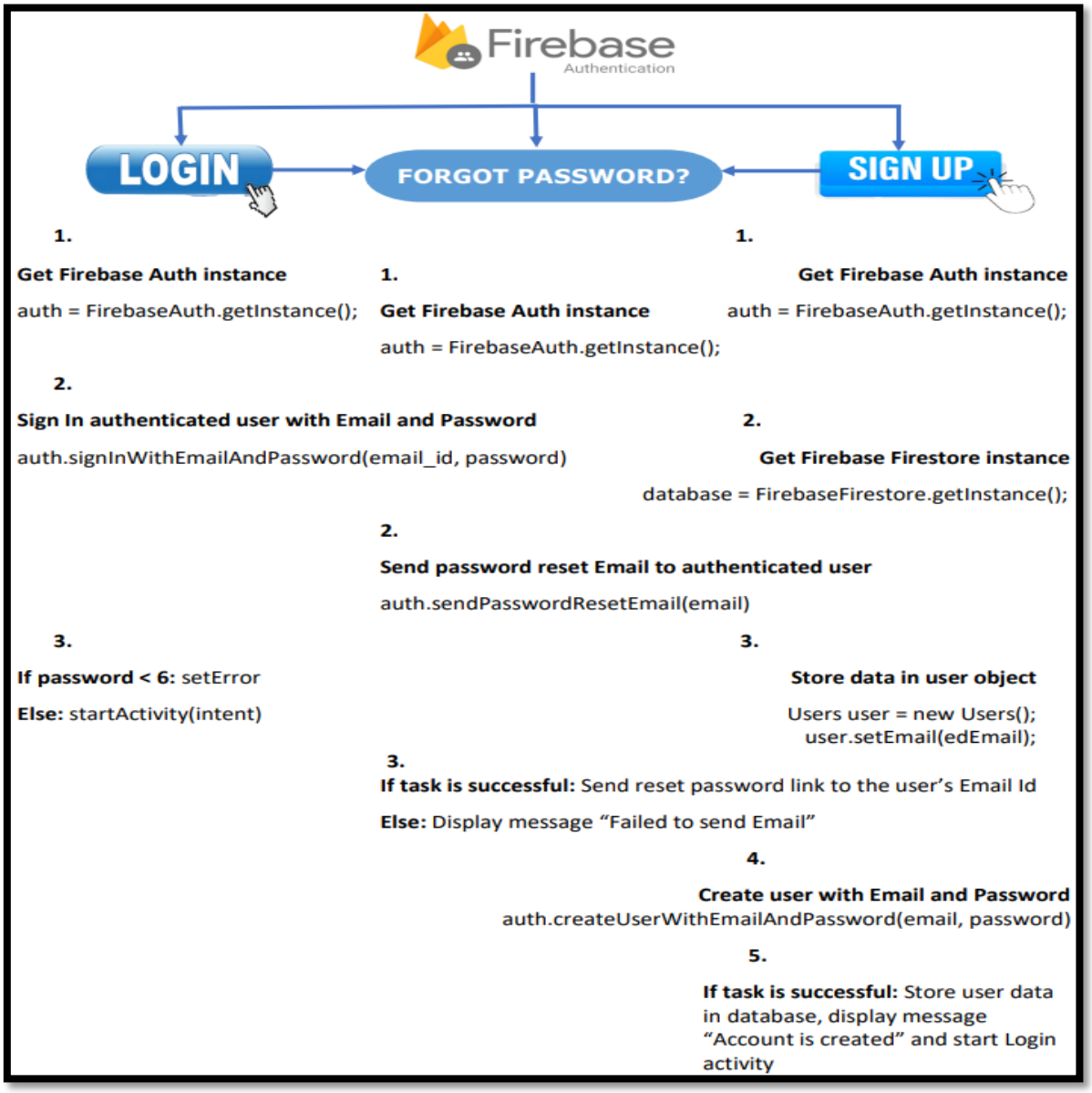

Fig. 6 Steps for creating Login, Forgot Password and Signup screens using Firebase Authentication

Features provided by Jitsi Meet SDK:

- **A. Enable lobby mode:** It allows only authorized users to get entry into the room by taking permission from the host.
- **B. Start Screen Sharing:** It shares the present screen of our device.
- **C. Toggle Camera:** It is used to switch between the front or back cameras.
- **D. Raise your hand:** In case any of the users have any doubt regarding anything, he/she can raise his/her hand.
- **E. Start recording:** A meeting can get recorded by clicking on this button.
- **F. Start live stream:** We can share our live meeting over YouTube by entering our Youtube live stream key.
- **G. Add meeting password:** To secure our meeting, we can add a meeting password.
- **H. Mute everyone:** If the host wants to speak and represent then he can use this feature to prevent unnecessary disturbance and echo.
- **I. Invite someone:** By clicking on this button, users can invite anyone or everyone to use this app.
- **J. Enable low bandwidth mode:** This feature allows users to get into a low bandwidth mode by turning off all the media streams.
- **K. Chat:** It allows users to chat while they are in a call or meeting.

The Virtual Training Room (VTR) app uses Firebase for authentication and Jitsi Meet SDK for calling purposes. Different features provided by the VTR app by making use of the Firebase Auth and Jitsi Meet SDK are shown in Fig. 7. Firebase Auth provides remote features because it uses Mobile Cloud Computing. There are various security features provided by Firebase Auth. A few of them are:

A. Developer can restrict read and write access to the Firebase data.
B. Users are authenticated by using emails and passwords or by phone numbers.
C. Developer can access security rules from the Firebase CLI.

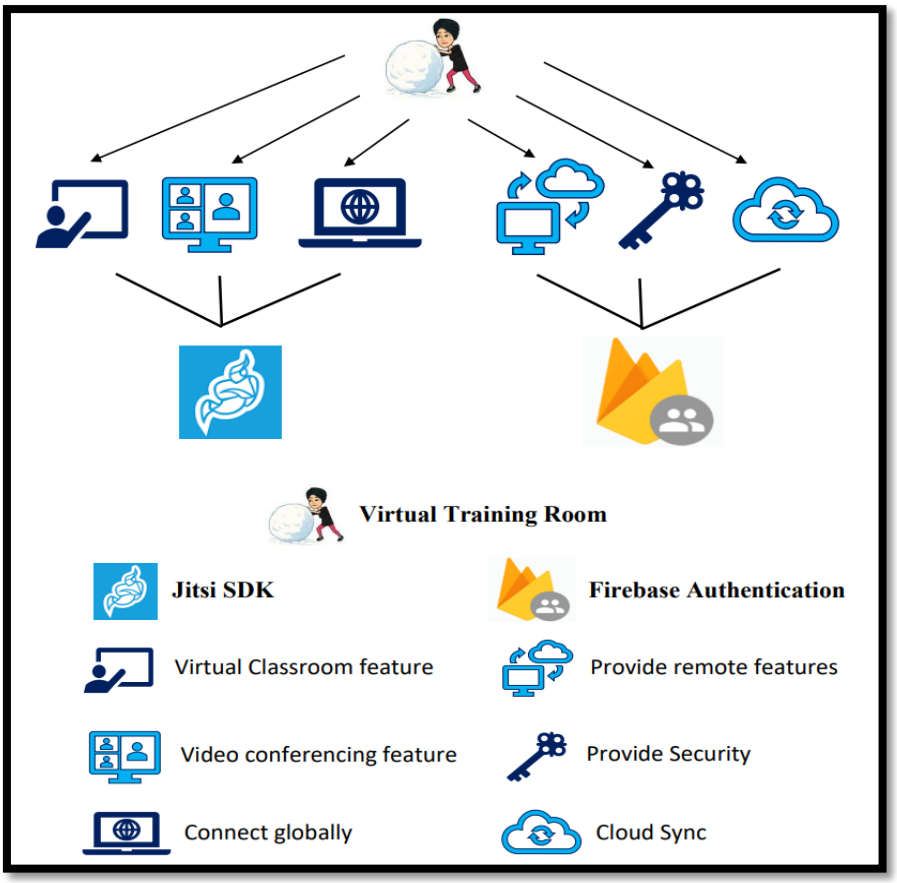

Fig. 7 Different features provided by the Virtual Training Room app

V. FINDINGS

Various video conferencing apps had faced many security breaches in the past. Zoom [7] had faced several privacy disasters due to which many companies banned the use of Zoom. From the Zoom credential hack, there are a lot of learnings. A few of them are:

A. **There should be proper waiting rooms:** Only authenticated users should be allowed to enter the room by the host.
B. **Only the host should have the right to start the meeting:** Firstly, the host should join the meeting. After that host should allow or decline access accordingly.
C. **There should be a unique meeting ID for every session:** To maintain security, there should be a unique meeting code for every session.
D. **There should be a lock option in the meeting room:** When all the users joined the room, there should be a lock option given to the host so that he/she can lock the meeting room to prevent unauthorized users from entering the room.
E. **End-to-end Encryption should be there:** To maintain privacy, there should be end-to-end encryption.

There are a lot of challenges that anyone can face while using Mobile Cloud Computing. A few of them are:

A. **Security:** There are many external factors that have access to private data while sending and receiving data from the mobile cloud. So, there should be additional security features provided to ensure high-performance levels.
B. **Network Availability:** MCC services are accessible only when the network is available. If we completely lose network access, then we cannot use the application.
C. **Performance:** Mobile cloud applications are accessible by remote servers across public networks which results in a slower response, due to which performance is compromised.
D. **Lack of Infrastructure:** For distributed applications, there is a lack of infrastructure because virtual devices cannot be created for such applications.
E. **Compatibility:** MCC supports multiple platforms, which can be expensive to implement because different network connections are required for different platforms.

Learnings from Firebase Auth used in the Virtual Training Room (VTR) app:

A. There should be proper implementation of Firebase Auth features. Proper Password Authentication [8] and Email Link Authentication [9] should be done to prevent sign-in from an unintended user or on an unintended device.
B. There should be proper password management which means no one should repeat passwords previously used, everyone should follow all the password creation rules provided, and everyone should change the password after every 3 months.
C. Same passwords should not be used across multiple accounts because if one account gets hacked then the password of other accounts will also get compromised.
D. One should not share their personal information like home address, photos of government identity cards, etc. with any of the applications.
E. For creating any application using MCC, security should not get compromised by any means. So, for security purposes, the following points should be taken care of:
   i) There should be proper encryption of data and strict key management to prevent unauthorized users from accessing the data.
   ii) There should be continuous security monitoring of the complete environment.
   iii) Passwords should not be stored as plaintexts. There should be proper encryption services throughout servers, databases, and networks.
   iv) Vulnerabilities and misconfigurations should be scanned regularly by conducting security audits and by performing penetration testing in the network environment.
   v) There should be proper firewalls, Intrusion Detection Systems, anti-malware, and access control. For better security, Endpoint Detection and Response (EDR) tools and Endpoint Protection Platforms (EPP) should be used.
   vi) Proper malware detection techniques should be used. There are 2 types of the malware detection technique. One is the static analysis technique and another one is the dynamic analysis technique. In the static analysis technique, malicious codes get detected by decompiling the application process. It is a relatively fast technique. Whereas in the dynamic analysis technique, malicious activity is detected by running the application on an emulator or a device.
   vii) There should be proper detection, monitoring, analysis, and identification of malicious code.
   viii) The zero-trust model should be used which means every person, device, or system needs to be cross-checked before connecting to any of the network systems or a device.

VI. CONCLUSION AND FUTURE SCOPE

Mobile Cloud Computing (MCC) is used by almost every industry whether it is the healthcare industry or finance and commerce industry etc. Security plays a major role while creating MCC apps. It should be taken care of by implementing proper security tools. In this paper, various challenges faced while implementing MCC apps have been covered. Video conferencing app named Virtual Training Room (VTR) has been created. In this app, Firebase Auth is used which provides authentication features by using MCC. Implementation steps of VTR are given in this paper. Basically, this app was created to get to know more about MCC. In the future, improvement of layouts can be done, more meeting rooms with more interesting background features can be added, more features like taking attendance, an inbuilt calendar for scheduling meetings, monitoring user's activity, etc. can be added to the app for better performance.

REFERENCES

[1]. Firebase Authentication. [Online]. Available: https://firebase.google.com/docs/auth

[2]. Case Studies: Firebase Authentication. [Online]. Available: https://firebase.google.com/docs/auth/case-studies

[3]. Android apps expose user data through Firebase blunders. [Online]. Available: https://www.comparitech.com/blog/information-security/firebase-misconfiguration-report/

[4]. Email Verification in Firebase Auth. [Online]. Available: https://firebase.blog/posts/2017/02/email-verification-in-firebase-auth

[5]. Android Jetpack. [Online]. Available: https://developer.android.com/jetpack

[6]. Firebase Authentication on Android. [Online]. Available: https://firebase.google.com/docs/auth/android/start

[7]. Zoom Credentials Hack. [Online]. Available: https://teampassword.com/blog/what-happened-with-the-zoom-credentials-hack

[8]. Password Authentication. [Online]. Available: https://firebase.google.com/docs/auth/android/password-auth

[9]. Email Link Authentication. [Online]. Available: https://firebase.google.com/docs/auth/android/email-link-auth

[10]. Muhammad Baqer Mollah, Md. Abul Kalam Azad, and Athanasios Vasilakos, "*Security and privacy challenges in mobile cloud computing: Survey and way ahead",* Journal of Network and Computer Applications, Volume 84, April 2017.

[11]. Abhiroop Saha and Nagaraj G Cholli, *"Mobile Cloud Computing: A Review",* International Journal of Advanced Research in Computer and Communication Engineering, Volume 10, Issue 5, May 2021.

[12]. Chunnu Khawas and Pritam Shah, *"Application of Firebase in Android App Development - A Study"*, International Journal of Computer Applications, Volume 179, No. 46, June 2018.

[13]. Abhinav Kathuria and Anu Gupta, *"Challenges in Android Application Development: A Case Study",* International Journal of Computer Science and Mobile Computing, Volume 4, Issue 5, May 2015.

[14]. Sai Spandhana Reddy Emmadi and Sirisha Potluri, *"Android Based Instant Messaging Application Using Firebase",* International Journal of Recent Technology and Engineering, Volume 7 Issue 5S2, January 2019.

