Anoniv Chat: A Cloud-Based iOS Application for Communication without Revealing the Identity

Apurva Anand, Prasoon, Bipul Kumar, R.M. Rani

Abstract: Communication is the part and parcel of everyone. In organizations including educational institutes, group communication is a common phenomenon. This type of communication happens between employees of the same team or organization, students of the same course or department, etc. Many mobile applications provide a group communication facility. However, the privacy of users is often compromised. For instance, in the WhatsApp Group, the mobile numbers are visible to every recipient. That is not always desirable. So, in this paper, a cloud-based iOS application is proposed for group communication in which the contact numbers are not visible. The administrator will control the read and write access to the recipients. Even, the identification label of the users inside the chat screen will be decided by the administrator. The application is coded in Swift. Cloud Firestore, one of the tools of the Firebase is used as the database and Firebase SDK is integrated with the code using CocoaPods.

Keywords - iOS, Swift, Cloud, Firestore, Firebase, SDK, CocoaPods.

I. INTRODUCTION
Communication is very important for any organization. There are innumerable applications of communication. It plays a very important role in every aspect of the functioning of an organization like Human Resources, Finance, Manufacturing, Research and Development, Information Services, and, not but limited to Consumer Services[1]. For better growth and development of any organization, people’s participation is a prerequisite and the medium is communication though which they share their thoughts and ideas. The decisions of the superiors should be properly announced and explained in such a way that they do not arouse antagonism. Hence, communication becomes very necessary for the process of Decision Making. Group Communication is one of the most common ways of interaction among the employees of the same team, students of the same department, among family members, etc. Nowadays, several communication groups are formed by the people who are unknown to each other but have common work-interests, or maybe they are enrolled in the same course, or just for some random discussion as well. The invitation link to join these kinds of groups are open to everyone. The sizes of these groups can vary from few to several hundred. If the size of the group is small or if the recipients are known to each other then there is no issue at all in general. However, if the size of the group is large and the recipients are not known to each other, then it can be catastrophic for the users especially, female users.

In many of those applications, the contact numbers are visible to every user. Thus, some disturbing elements may obtain confidential information of others[2]. Hence, it becomes very necessary to protect the privacy of the users. Even, many applications coded for a particular operating system do not run on all the devices supporting that very operating system. Also, the User Interface is not completely flexible according to the organization. Even, the number of recipients can be limited in a single group.

Hence, an iOS application is proposed that provides a chatting facility by fully tackling the privacy concern. The contact number of one user will not be visible to other users. In the chat screen, their unique identification label will be provided by the administrator. Different users will have either read or write or both access and this will be controlled by the administrator. Backend storage, retrieval, and processing of the data are done in the Firebase. This application will run on all the devices that support iOS 10 or later versions. The looks, logo, and other User Interface items can be customized according to the institution and its needs. It can have an unlimited number of users. This application is developed in the Xcode IDE and is coded in the Swift language. Different third party libraries are integrated with the code using CocoaPods and Swift Package Manager. In section II existing applications are highlighted. In section III, the proposed application is discussed in detail with the involved technologies and advantages of this application. Section IV deals with the system architecture. System Implementation is mentioned in section V. Future Scope and Conclusion is mentioned in section VI and VII respectively. The paper ends up with the Acknowledgement and References.

II. LITERATURE SURVEY
Many applications provide Group Chatting facility. However, none of them is cent percent perfect. They lag in one aspect or the other.

II.1. WhatsApp Messenger[3]
It is a messaging application used for sending text messages, images, audio, and video to a single recipient or in groups. It is owned by Facebook Inc and is available for iPhones, Android Devices,
BlackBerry Devices, and Window Phones. It’s very easy and convenient to use. The data is stored in the cloud storage account of the user. But in Groups, the phone numbers are visible to everyone. Also, the maximum number of recipients is limited to 256[4].

II.2. Telegram[5]
It is an application used for communication in various forms like individual chatting and group chatting. The users can be found from their Usernames. It is available for iOS, Android, Windows Phone, Windows NT, macOS, and Linux. However, it doesn't provide end-to-end encryption and all the data including passwords is permanently stored to its servers.

It is a mobile application especially designed for corporates by Microsoft Corporation for messaging and workflow management. It is available for iOS, Android, and Web Platform. Messaging can be done either by admins or everyone. It’s not that few non-admins can be selected for messaging over others by the admins. Hence, it has flexibility issues.

It is a messaging application owned by Rakuten, a Japanese MNC. It is available for Android, iOS, Windows, macOS and Linux platforms. End-to-End encryption has recently been added. However, the User Interface is not customizable. Also, the read and write access is applicable for all non-admins.

II.5. Skype[8]
It is a communication application that was once especially used for video conferencing. It is available for iOS, Android, Windows Phone, Windows OS, macOS, and Linux. It operates on the peer-to-peer model and not on the client-server model. RSA is used for key negotiation and AES to encrypt conversations. But the correct usage of these algorithms is not verified.

II.6. Facebook Messenger[9]
It is a messaging application used mostly for informal communication. However, a Facebook account is needed. This is a limit. Although, it provides end-to-end encryption instead the privacy can be compromised to a large extent because the details of users can be known like their display picture.

II.7. Wire[10]
It is a communication application available for Android, iOS, macOS, Linux and, in the form of Web Application. Its features include file sharing, video conferences, and messaging. End-to-End encryption is provided for the user’s data. However, it has flexibility issues. And the pro version is paid.

It is a communication application available for iOS, Android, and Jio Phones developed by Reliance Industries Limited. It provides a Messaging facility, HD Video Calling, and File Sharing. It also has Group a Communication facility. However, it is exclusively for Jio Sim users and the contact numbers in a Group are visible to every recipient. Thus, privacy is compromised.

It is an open-source communication application that is end-to-end encrypted. It is available for iOS, Android and Desktop. Its features include Video Calls, Voice Calls, Messaging, and Group Chatting. It has one special feature of Disappearing Messages. Messages will disappear after the disappearance timer. However, in Groups, the contact number is visible to every recipient and that is a drawback. Also, the User Interface can’t be customized for individual usage.

It is a communication application developed by Google. It can be used for individual or group conversations, video calls, voice calls, and file sharing. It is available for Chrome OS, iOS, Android, and Web Platform. But the contact numbers are visible to every recipient of groups and hence, it is a serious concern.

II.11. WeChat[14]
It is a messaging application developed by Tencent Holdings Limited. Its features include voice calls, and video calls with multiple persons up to 9 members, group chatting with up to 500 members, and location sharing. However, it is not flexible.
It is a messaging application that is available for Windows OS, Windows Phone, macOS, iOS, and Android. It is developed by Slack Technologies. Its features include group communication and organization of conversations according to the topic and integration with other standard applications. It doesn’t provide end-to-end encryption and stores user data exclusively on cloud servers under its control.

III. PROPOSED APPLICATION
We propose a cloud-based iOS application for group communication. It is a central administrator controlled SaaS application. The users can communicate with each other without revealing their contact numbers. They shall register and login by using their email address and password and the email address will not be known by anyone except the user and the administrator. They shall be provided with a unique identification label by the administrator that will be displayed automatically in the right of their message cell after they send their message and other’s label will be on the left side of their respective messages. Sample message cells are presented below in Fig.2 with Me Avatar as the identification label for oneself and in Fig.3 with You Avatar as the identification label for the other user.
Thus, the contact number needs not to be visible at all. The app is coded in the Swift language in the Xcode IDE. Cloud Firestore is used as the database for storing and retrieval of messages. Firebase Authentication is enabled to let the users authenticate with Firebase using their email addresses and pass - words. Firebase is all in one suite of tools that is backed by Google allowing us to do a lot of things like storing of data in the Cloud Firestore, authentication of users, encryption of passwords. It also helps in incorporating Google Analytics, In-App Messaging, Predictions, A/B Testing, etc.

III.1. Technologies Involved
III.1.i. Cloud Computing[16]
It is a technology that enables us to store and process data by using computing resources available on remote servers. Its application ranges from storing data to the building of applications using remote development platforms. It even includes remote infrastructures where those development platforms are installed. The cloud services can be accessed through the internet in general or through intranet in case of on-premises cloud storage.

III.1.ii. iOS[17]
The second most popular mobile operating system iOS is developed by Apple Inc. especially for its devices like iPhones, iPods touch and iPads as well before the launch of iPadOS in 2019.

III.1.iii. Xcode
It is an Integrated Development Environment for the development of applications for iPhone, iPad, Mac-Book, Apple Watch, and Apple TV. Its development user interface comprises of Status Bar, Navigation Pane, Document Outline, Interface Builder, Code Editor, Inspector Pane, and Debug Pane.

III.1.iv. Firebase[18]
Firebase is all in one suite of tools that is backed by Google and allows us to do a lot of things like storage of data in the Cloud Firestore, authentication of users, encryption of passwords. It also helps in incorporating Google Analytics, In-App Messaging, Predictions, A/B Testing, etc.

III.2. Advantages of the proposed application
• It is a cloud-based application. Hence, an account can be accessed from multiple devices.
• The delivery of messages is very fast.
• This application provides a very smooth User Experience as the design is very simple and it is free from ads.
• The maximum number of recipients is unlimited.
• Firebase Authentication has been incorporated that enables the user to log in or register with an email address and password. It supports multiple levels of complexity.
• A user can’t find the contact numbers and any information about the other user. Hence, the emphasis has been laid to maintain privacy.
• This application runs on all the devices supporting iOS 10 or later versions including iPad due to the incorporation of Auto Layout.
IV. SYSTEM ARCHITECTURE

The architecture of the entire application is presented through 3-tier architecture

1. Presentation Tier - This tier represents the User Interface elements or the elements that are seen by the User. As seen in Fig. 1, it has a Welcome Screen, Login Screen, Registration Screen, and Chat Screen. The application opens up with the Welcome Screen with two buttons Login and Register. Depending upon the new or existing user, either register or login button is pressed respectively. After that, either registration or login screens opens up as seen from Fig. 4. Both have two text fields for email id and password and Register and Login button respectively to register or login to the system. They also have a Back button for moving back to the Welcome Screen. After pressing either of the buttons, Chat Screen opens up. It has a text field, Send button, and a Log out button. The user can write a message and send and the messages appear in a message cell with a different identification label for different users as shown in Fig.2 and Fig.3. Or, the Logout button can be pressed to return to the Welcome Screen.

2. Application Tier - This tier contains the backend implementation of the application. Some sort of backend implementation is always associated with every activity of the users. The Following functions are called during the entire application lifecycle.
   • application(_didFinishLaunchingWithOptions:) - As soon as the application is launched, this is the first function called. It is called to establish connection with the Firebase.
   • scene(_:willConnectTo:options:) - This function is called when the storyboard is used, the `window` property needs to be automatically initialized and attached to the scene.
   • sceneWillEnterForeground(_:) - This function is called as the scene transitions from the foreground to the background.
   • viewDidLoad() - IBOutlets, IBActions, and all the view related objects need to get connected up with the application and is done by the invocation of this method. IBOutlets integrates the static view elements with the application. IBAction enables the application to detect any input given by the user like tapping a button and trigger the required action. This method is only called once when the view is created.
   • viewWillAppear(_:) - This method is invoked up just before the Welcome Screen shows up on the screen. In this method, certain User Interface elements can be hidden that the developer doesn’t want to show on the screen. Here, a Back button is hidden from the top left of the Chat Screen.

![Fig.5. Backend Flowchart](image)

   • sceneDidEnterBackground(_:) - This method is called when the application appears on the screen. It is called when the scene has moved from an inactive state to an active state.
   • viewWillDisappear(_:) - This method is called when the user moves on to a different page other than the Welcome Screen.
   • loginPressed(_:) - This method is called after the user taps on the Log In button after entering the login credentials. Segue is used to connect the login page and Registration page to the Chat Screen. The Authentication of the user is done by the Firebase.
• loadMessages() - This method retrieves the previously sent messages and loads them on the Chat Screen.
• tableView(_:cellForRowAt:) - This method is called to load message cells dynamically on the Table View element on the Chat Screen.
• awakeFromNib() - This method is overridden to resize and load the message cell which is a XIB file. XIB files are the Graphical User Interface developed by the XML interface builder.
• setSelected(_:animated:) - This function is overridden to configure the view for the selected state. For instance, when the Chat Screen is scrolled up or down to view the old or new messages respectively.
• sendPressed(_: ) - This method is invoked when the sender presses on the Send button to send a message. The message is saved to the cloud storage.
• registerPressed(_: ) - This method is called when the new user presses the Register button after entering the registration credentials on the Registration page. A new account is created after the validation steps done by the Firebase and the credentials are saved if they match the required criteria.
• sceneWillResignActive(_: ) - This method is called when the scene moves from an active state to an inactive state. This may occur due to temporary interruptions (ex. an incoming phone call).
• logOutPressed(_: ) - This method is invoked after the user presses the Log Out button. The Chat Screen is redirected to the Welcome Screen.
• sceneDidEnterBackground(_: ) - During exiting the application, this method is used to save data, release shared resources like the current memory, and store enough scene-specific state information and to restore the scene back to its current state. The backend flowchart is shown in Fig. 5.

3. Data Tier - This tier contains the database. In this application, Cloud Firestore is used as the database. It saves messages and login credentials.

VI. RESULT

1. The application opens up with the Welcome Screen as shown in Fig.6.

2. New users click on the Register button to open the registration page as shown in Fig.7.
3. Existing users click on the Log In button to open the login page as shown in Fig.8.
4. Both types of users enter their respective credentials to open up the Chat Screen as shown in Fig. 9.

**VII. CONCLUSION**

An application is developed that secures the privacy of the recipients. It has a very simple and easy to use User Interface. It is highly secured and the delivery of messages is done very quickly. As of now, it is developed for iOS devices only but in the future, it can also be developed for other mobile and web platforms. In addition, several additional features can be added like the incorporation of the transaction of multimedia files, voice call, video call, etc. Also, the performance can be further improved by adding features like Crash Analytics and Performance Monitoring.
REFERENCES


Fig.9. Chat Screen