

# Student's Hub - A mobile Chat Application with Chatbot

<sup>1</sup>Dr. .M.S. Chaudhari, <sup>2</sup>Prajwal Lanjewar, <sup>2</sup>Prasad Pusadkar, <sup>2</sup>Prem Thepale,  
<sup>2</sup>Pranay Choudhari, <sup>2</sup>Swapnil Nakade, <sup>2</sup>Harish Dharaskar

<sup>1</sup>Head & Assistant Professor, Department of Information Technology, Priyadarshini Bhagwati College of Engineering, Nagpur, Maharashtra, India

<sup>2</sup>UG Student, Department of Information Technology, Priyadarshini Bhagwati College of Engineering, Nagpur, Maharashtra, India

**Abstract:- This application is intended to serve as a platform for students to chat with each other in groups. Students can create their own groups or join existing ones. For ease of access, students can also access the college portal directly from the application. In addition to chatting, students will also have a Dialogflow-based chatbot that will provide information about the college campus.**

**Keywords:-** Flutter, Dart, Firebase Authentication, Firestore, Collection, Document, Dialogflow, API key, JWT.

## I. INTRODUCTION

Chatting is a popular mode of communication among the youth. StudentsHub is a platform that caters to this need by providing students with a dedicated space to chat with their classmates in groups, create their own groups, or join existing ones. Students can leave or join existing groups at their own discretion.

When a new student joins the college, they may not be familiar with any of the faculty or staff, which can make it difficult to get their problems solved quickly. To address this issue, StudentsHub also includes a DialogFlow-based chatbot that can assist students with their queries.

A Flutter application allows users to register and log in using their email address and password. Once logged in, users can read and send messages to a shared room with all other users of the application. Due to its Flutter implementation, the application is compatible with iOS, Android, and web devices.

The Flutter framework is used on the frontend of the application, and the Firebase platform is used on the backend. Since Firebase provides convenient APIs for accessing NoSQL databases, the application will access Firebase through the `firebase_core` package, which offers a straightforward way to read and write data to the database.

The user interface (UI) for the app has been designed to be simple and modern, with the user experience in mind. This means that the UI is easy to use and navigate, and that it provides all of the necessary information and functionality in a clear and concise way. The UI also uses a modern design aesthetic, which makes it visually appealing and engaging.

The chatbot integrated into the application is built using Google Dialogflow. Dialogflow is a conversational AI platform that allows developers to create natural language interfaces for chatbots, voice assistants, and other applications. The chatbot has been trained on a large corpus of text and code, including information about college, so that it can provide students with assistance on a variety of college-related questions

## II. LITERATURE SURVEY

Title of paper	Method/ Techniques used	Analysis and Observation
“AI CHATBOT USING DIALOG FLOW”IJCSMC, Vol. 11, Issue. 5, May 2022.	Collecting data,Data manipulation	The paper highlights the role of chatbots as AI-driven virtual assistants, emphasizing their utility in online product and service promotion. It also mentions that Higher Educational Institutes leverage websites and social networks like Facebook and WhatsApp to provide information to students. This reflects the increasing integration of technology and AI in educational and promotional contexts, enhancing accessibility and communication channels.
“Developing of Middleware and Cross Platform Chat Application”(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 12, No. 11, 2021	System Design,System Test	The research aims to develop a middleware chat application to bridge this gap, enabling communication between the new app and established ones like Telegram and LINE. Extensive testing is being conducted to ensure the seamless exchange of various types of messages. This research addresses a significant user pain point in the increasingly diverse landscape of chat applications.
“Android Based Instant Messaging Application Using Firebase”(IJRTE) ISSN: 2277-3878, Volume-7 Issue-5S2, January 2019.	Firebase Authentication,Real time database,	The paper highlighting the need for fast and convenient text, image, and file sharing with minimal delays. It points to Firebase as a platform that facilitates real-time database and cloud services for the development of such applications. The main objective of the paper is to introduce a software application for real-time communication.

## III. METHODOLOGY

### ➤ *Integration of Firebase SDK*

Seamless integration of Firebase into the Flutter application is pivotal. To achieve this, the Firebase SDK is incorporated into the project, ensuring that the app can effectively communicate with Firebase services.

### ➤ *Initialization of Firebase Authentication*

The Firebase Authentication instance is initialized within the Flutter application. This initialization process establishes the connection between the app and Firebase's authentication services, setting the stage for user authentication.

### ➤ *User Account Management*

The process of user authentication begins with user account management. Two primary operations are outlined:

a. **Account Creation:** The app facilitates the creation of new user accounts using the `createUserWithEmailAndPassword()` method. Users input their email address and password, which are then validated by Firebase Authentication. Upon successful validation, Firebase generates a JSON Web Token (JWT) associated with the user's account.

b. **User Sign-In:** For existing users, the app employs the `signInWithEmailAndPassword()` method. This method verifies the provided email and password against Firebase Authentication records and, upon successful validation, issues

a JWT to authenticate the user.

### ➤ *JWT Utilization*

Upon authentication, a JSON Web Token (JWT) is returned to the app and stored locally. The JWT serves as a digital signature, encapsulating crucial user information, such as their email address and unique user identifier (UID). Importantly, the JWT is signed by Firebase Authentication, ensuring data integrity.

### ➤ *Authentication and User Access*

The app employs the JWT to authenticate the user with Firebase services and access user-specific data and functionality. Firebase services like Firestore, Cloud Storage, and real-time authentication can be leveraged seamlessly using the JWT.

### ➤ *User Personalization*

The utilization of the JWT extends to personalized user experiences. By referencing the JWT, the app identifies the user, tailors content, and provides personalized experiences. This personalization can encompass user-specific settings, content recommendations, and user-level access controls.

### ➤ *Advantages of Firebase Authentication*

The research methodology acknowledges the advantages of employing Firebase Authentication for email and password authentication within the Flutter application. These advantages encompass:

- **Ease of Setup:** Firebase Authentication simplifies the setup process, ensuring that even complex authentication mechanisms can be implemented with relative ease.
- **Security and Reliability:** Firebase Authentication offers robust security mechanisms and reliability, safeguarding user data and authentication processes.
- **Scalability:** The service is inherently scalable, capable of supporting a substantial user base without compromising performance.
- **Feature-Rich:** Firebase Authentication presents a range of features, including real-time authentication and passwordless sign-in, enhancing the user experience and security standards.

#### ➤ *Firestore Database Configuration*

The proposed group chat application utilizes a Firestore database on the Google Firebase platform. The database schema comprises two primary collections: users and groups

#### ➤ *Users Collection*

The users collection stores essential user information, including fullname, email, and a unique identifier (UID). The UID facilitates user management and profile maintenance, while enabling personalized interactions within the group chat environment.

#### ➤ *Groups Collection*

The groups collection stores information about each group, such as its name, description, members, and chat messages. The representation of group members is flexible, with two prominent methodologies:

- **UID storage:** A user's UID is stored within the group document to denote membership. This facilitates member tracking and permits the retrieval of user details from the users collection.
- **Sub-collection creation:** A subcollection is created within each group document to encapsulate member particulars. This approach offers greater flexibility in managing group membership data.

#### ➤ *Group Creation*

Group creation is initiated by adding a new document to the groups collection. The document encompasses essential group attributes, including the group's name, description, and creator's UID. The creator's UID is also added to the group's membership list.

#### ➤ *Group Chat*

Group chat functionality is implemented through a designated "chat messages" field within each group document. This field stores the chronological sequence of messages, either as an array or a subcollection. As a user transmits a message, it is appended to the group's chat messages repository. Each message includes the sender's UID, message content, timestamp, and other pertinent information.

#### ➤ *User Authentication and Authorization*

Firebase Authentication is utilized to authenticate users and ensure application integrity. Only authenticated users can create groups and send messages.

#### ➤ *Real-time Updates*

Firestore Realtime Database is leveraged to implement real-time updates for group chat messages and memberships. All group members receive immediate notifications when new messages are sent or group memberships change. This enhances the application's interactive dynamics.

Overall, the proposed group chat application leverages the capabilities of Firestore to implement a user-friendly and interactive group chat experience.

Dialogflow, a natural language processing (NLP) platform from Google, allows developers to create conversational AI experiences into their applications. Flutter, a cross-platform development framework, enables developers to build native user interfaces for Android, iOS, Web, and desktop from a single codebase.

Dialogflow and Flutter integration creates conversational AI applications deployable on a variety of devices. For example, you could build a Flutter app with Dialogflow powering a chatbot that helps students find information about their college campus.

Dialog flow and Flutter integration steps include:

- Creating a Dialogflow agent.
- Training the Dialogflow agent with intents and entities relevant to the college campus.
- Exporting the Dialogflow agent as a JSON file.
- Adding the Dialogflow Flutter plugin to the Flutter project.
- Initializing the Dialogflow Flutter plugin with the Dialogflow agent JSON file Sending and receiving messages to the Dialogflow agent from the Flutter app.

To train Dialogflow for college premises, create a set of intents and entities relevant to the college campus. Here are some intent examples:

- **Find information:** To find information about the college campus, such as the location of a building or the hours of operation for a department.
- **Get help with a problem:** To get help with a problem that the student is having, such as a problem with their tuition or a problem with a class.
- **Provide feedback:** To provide feedback about the college campus or about the chatbot.

#### ➤ *Here are some entity examples:*

- **Building:** To represent the name of a building on the college campus.
- **Department:** To represent the name of a department on the college campus.
- **Course:** To represent the name of a course on the college campus.

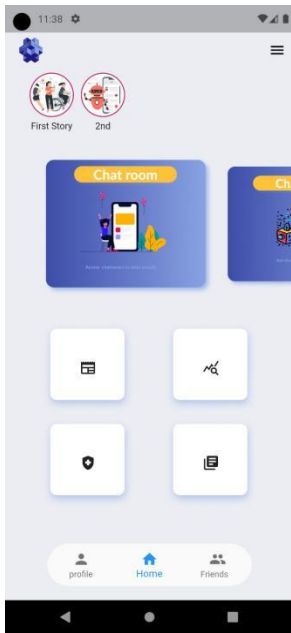


Fig 1. Home

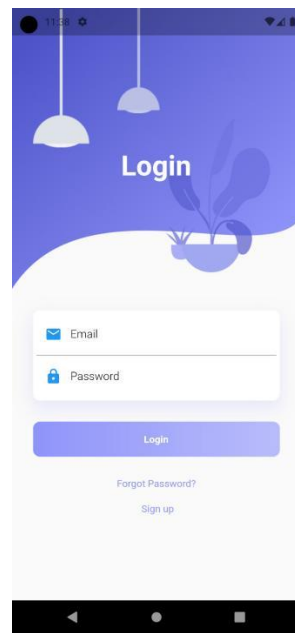


Fig 2. Login page

#### IV. CONCLUSION

The StudentsHub application uses Firebase, Flutter, and Dialogflow. These technologies provide a robust, scalable, and user-friendly platform for students to communicate and access information. Firebase Authentication provides secure and convenient user management, Firestore database offers a structured and flexible data storage solution, and Dialogflow enables natural language processing for the chatbot feature. These technologies offer several advantages, including ease of setup, security, reliability, and feature-rich capabilities.

Overall, the combination of Firebase, Flutter, and Dialogflow creates a powerful and user-centric platform that caters to the communication and information needs of college students, facilitating a seamless and engaging experience.

#### REFERENCES

- [1]. R. Chaocarro, M. Cortinas, and G Marcos-Matas, "Teachers attitude towards chatbots in education: a technology acceptance model approach considering the effect of social language, bot proactiveness, and user's characteristics", *Educational studies*, 2021.
- [2]. W. Villegas-Ch, A. Arias-Navarrete, and X. Palacios-Pacheco, "Proposal of an architecture for the integration of a chatbot with artificial intelligence in a smart campus for the improvement of learning," *Sustainability*, 2020.
- [3]. Sharma, D., & Agarawal, M. (2021). Chat Messenger using Flutter. *International Journal of Computer Applications*, 174(30).
- [4]. Sharma, D., Agarawal, M., Upadhyay, H., & Akilarasu, G. (2021). Developing Chat Application using Firebase. *International Research Journal of Engineering and Technology*,
- [5]. Developing Chat Application using Firebase. (2021). *International Research Journal of Engineering and Technology*.
- [6]. Chat Application Using Flutter and Firebase. (2020). *International Journal of Innovative Research in Computer Science & Technology*.
- [7]. Madhuram M., Ashu Kumar , Pandyananian. M, "Cross Platform Development using Flutter", *International Journal of Engineering Science and Computing* , April 2019.
- [8]. Vol. 05 Issue 01 2020 Chat Application Using Flutter and Firebase" by IJIIRD 3. This paper discusses the implementation of a chat room for Android mobile phones using Flutter and Firebas