**JSS MAHAVIDYAPEETHA**

**JSS Science and Technology University**



***A*** ***Report*** ***on***

**“CHAT APP WITH SPAM DETECTION”**

**Subject : Mini Project**

**Subject code: 20CS69P**

***Submitted by***

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **USN** | **NAME** |
| **1.** | **01JCE21CS061** | **Manoj D Maiya** |
| **2.** | **01JCE21CS090** | **Sharath HK** |
| **3.** | **01JST21CS031** | **Darsh Khetan** |

***Under the guidance of***

**Prof. G N Shwetha**

Assistant Professor

Dept. of CS&E

SJCE, JSS STU, Mysuru

**Department of Computer Science & Engineering**

**2023-2024**

**JSS MAHAVIDYAPEETHA**

**JSS Science and Technology University**



**CERTIFICATE**

Certified that the project entitled " **CHAT APP WITH SPAM DETECTION”**, conducted for the course "**MINI PROJECT(20CS69P)**", was undertaken by **Manoj D Maiya** [**01JCE21CS061**], **Sharath HK [01JCE21CS090]**, **Darsh Khetan** [**01JST21CS031**], in partial fulfillment of the requirements of [VI/2024], as prescribed by the JSS Science and Technology University, Mysore, during the year 2023-2024. It is certified that all corrections/suggestions indicated for the project have been incorporated. The project report has been approved as it satisfies the academic requirements for the respective course.

**Smt. G N Shwetha Dr Srinath S**

Assistant Professor Associate Professor and Head

Department of CS & E, Department of CS & E

JSS STU, Mysuru-06 JSS STU, Mysuru-06

**DECLARATION**

We do hereby declare that the project titled **“**CHAT APP WITH SPAM DETECTION**”** is carried out by Manoj D Maiya [01JCE21CS061], Sharath HK [01JCE21CS090], Darsh Khetan [01JST21CS031], under the guidance of **Smt**. **G N Shwetha, Assistant Professor**, **Department of** **Computer Science & Engineering,** in partial fulfillment of the requirements of [VI/2024], as prescribed by JSS Science and Technology University, Mysore, during the year 2023-2024.

**Date: 11 June 2024**

**Place: Mysuru**

**Manoj D Maiya** [**01JCE21CS061**]

**Sharath HK [01JCE21CS090]**

**Darsh Khetan** [**01JST21CS031**]

**ABSTRACT**

This project builds a secure and dynamic chat application for seamless communication across devices. It utilizes React Native for a unified codebase, Firebase for real-time functionalities and secure authentication, for instant message delivery. The key feature is the integration of spam detection algorithms to ensure a safe user experience. This is particularly important as the prevalence of social media spam continues to grow, posing threats like malicious links, fake accounts, and misinformation. This innovative chat app prioritizes user safety by filtering out unwanted content, creating a more secure and enjoyable online communication experience.

**ACKNOWLEDGEMENTS**

An endeavor is successful only when it is carried out under proper guidance and blessings. We would like to thank few people who helped us in carrying this work by lending invaluable assistance. We thank Dr. C Nataraju, Principal, JSSTU, Mysuru and Dr. Srinath S , Associate Professor and Head, Department of Computer Science and Engineering, JSSSTU, Mysuru who encouraged us at this venture. It is our foremost duty to thank our project guide Smt. G N Shwetha for her encouragement, effective guidance and valuable suggestions right from the beginning of this project to till its completion. We also extend our regards to all the teaching and nonteaching members of Department of Computer Science and Engineering for their direct or indirect support towards the completion of this project. We would also like to thank our family and friends for their constant support.

**TABLE OF CONTENTS**

## INTRODUCTION

* 1. [Problem Domain **1**](#_TOC_250012)
  2. [Aim **2**](#_TOC_250011)
  3. [Statement of the problem **2**](#_TOC_250010)
  4. Objectives of the project work **3**
  5. [Application **4**](#_TOC_250009)
  6. [Existing Solution Methods **4**](#_TOC_250008)
  7. [Limitations **5**](#_TOC_250007)
  8. [Proposed Solutions **6**](#_TOC_250006)

## LITERATURE SURVEY

### 2.1 Introduction 7

### 2.1 Literature Survey Table 8

### SYSTEM DESIGN 12

### 

### RESULTS AND DISCUSSION 13

**CONCLUSION**  19

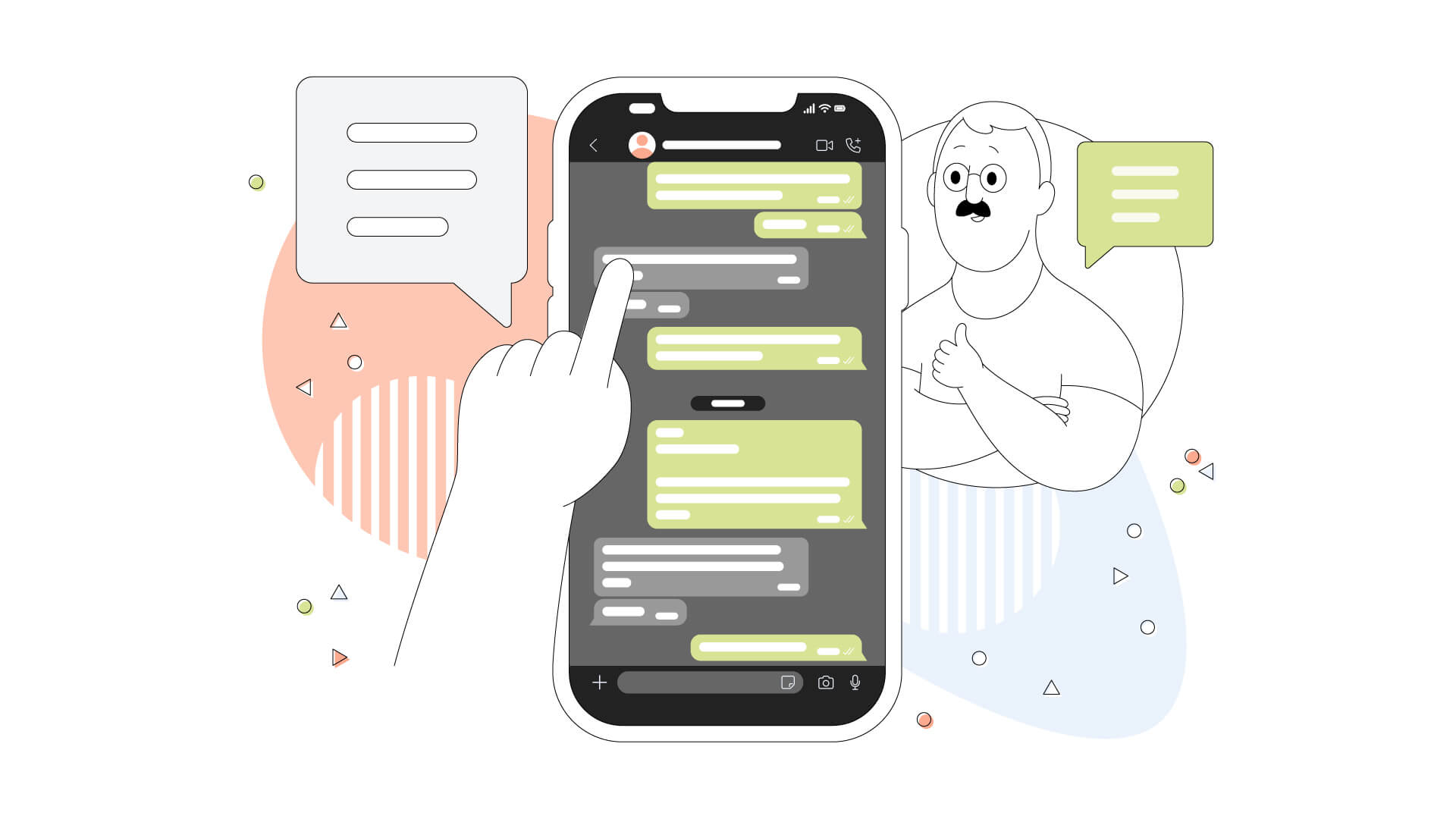
**FUTURE WORK 20**

**REFERENCES 21**

**CHAPTER 1**

**INTRODUCTION**

React Native and machine learning can be fused into a conversation app with spam detection, thereby merging current mobile app development with superior data handling. Consequently, through React Native, it becomes possible to produce a user-friendly and cross-platform interface which is compatible both with iOS and Android. Additionally, such libraries allow us to create real-time messaging applications. It handles authentication of users as well as message storage from the backend side using such technologies as Node.js and Express which provide the supported database like Firebase Firestore. So, spam detection depends on machine learning models created in TensorFlow for eliminating undesired messages. The said model can reside in a server and be made available via an API for efficient communications enhanced user safety purposes.



* 1. **Problem Domain**

The problem domain for a chat app with spam detection, focusing on React Native and machine learning, involves seamlessly integrating real-time messaging in a cross-platform app while efficiently identifying and filtering spam messages. Challenges include developing a responsive interface with React Native and deploying a machine learning model that adapts to evolving spam patterns without compromising app performance. Additionally, ensuring smooth interaction between frontend and backend systems is crucial for reliability and security.

* 1. **Aim**

The aim of developing a chat application with spam detection using React Native and machine learning is to create a seamless and secure communication platform. This involves integrating real-time messaging functionalities across iOS and Android devices while effectively identifying and filtering out spam messages. The goal is to provide users with a responsive and intuitive interface, ensuring efficient spam detection without compromising app performance. Additionally, the aim is to establish smooth interaction between the frontend and backend systems to deliver a reliable and secure chat experience.

* 1. **Statement of the Problem**

Developing a chat app with spam detection using React Native and machine learning poses challenges in integrating real-time messaging and accurate spam identification. Balancing performance with real-time demands while ensuring user data security is crucial. Limited mobile device resources complicate efficient machine learning model implementation. Prioritizing an intuitive user interface and robust spam filtering is essential for optimal user experience. Collaboration between frontend and backend developers is vital for effectively tackling these challenges and delivering a reliable, secure, user-friendly app.

* 1. **Objective of the project work**

The objectives of our chat application project, leveraging React Native, Firebase, are multifaceted and aimed at delivering a robust and user-centric communication platform:

1. **Seamless Cross-Platform Communication:** Our primary objective is to develop a chat application that enables seamless communication across iOS and Android devices. By leveraging the cross-platform capabilities of React Native, we aim to create a unified user experience regardless of the operating system.
2. **Real-Time Message Delivery:** We strive to implement real-time message delivery using Firebase Realtime Database, ensuring that users can exchange messages instantly without delays. This objective focuses on enhancing user engagement and responsiveness within the application.
3. **Secure User Authentication:** Ensuring the security of user data is paramount. Our objective is to implement secure user authentication mechanisms using Firebase Authentication, safeguarding user accounts and personal information from unauthorized access or data breaches.
4. **Spam Detection and Prevention:** An important objective is the integration of spam detection mechanisms within the application. Leveraging machine learning algorithms and natural language processing, we aim to detect and filter out spam messages in real-time, enhancing the overall user experience by mitigating unwanted content.
5. **User-Friendly Interface:** We aim to design an intuitive and aesthetically pleasing user interface that enhances user experience and encourages engagement. This objective emphasizes user-centric design principles, prioritizing simplicity, clarity, and accessibility.

By aligning our objectives with these key principles, we aim to create a chat application that not only meets but exceeds user expectations, delivering a secure, responsive, and enjoyable communication experience for users worldwide.

* 1. **Application**

The project's application is to develop a chat app with spam detection using React Native and machine learning. It aims to provide secure communication by integrating real-time messaging and efficient spam filtering. With an intuitive interface, it caters to personal communication, business collaboration, and customer support, addressing the demand for secure messaging solutions.

* 1. **Existing Solution Methods**

Existing solution methods for chat application spam detection include rule-based filtering, keyword matching, and machine learning techniques like natural language processing (NLP) and supervised learning algorithms. While rule-based approaches are simplistic, machine learning offers more advanced solutions by analyzing message content or training models on labeled datasets. However, these methods require computational resources and model tuning. Integrating spam detection in a chat app also involves real-time processing and user privacy considerations. The challenge lies in selecting the most effective method that balances accuracy, efficiency, and user experience.

* 1. **Limitations**

While our chat application harnesses cutting-edge technologies to provide a seamless user experience, several limitations merit consideration:

1. **Limited Spam Detection Accuracy:** Despite employing machine learning algorithms and natural language processing techniques, our spam detection mechanism may not detect all spam messages with absolute precision. False positives or false negatives could occur, potentially impacting user trust and experience.
2. **Dependency on Internet Connectivity:** As our application relies on real-time communication via Firebase, uninterrupted internet connectivity is crucial for optimal performance. Users in areas with poor network coverage may experience disruptions or delays in message delivery.
3. **Platform Compatibility Challenges:** Although React Native facilitates cross-platform development, ensuring uniform functionality and user experience across iOS and Android devices can be challenging. Device-specific nuances and compatibility issues may arise, requiring additional testing and optimization efforts.
4. **Scalability Concerns:** While Firebase offers scalable real-time database solutions, handling a large volume of concurrent users could strain system resources and affect performance. Scalability considerations must be addressed to accommodate future growth and user demand effectively.
5. **Security Vulnerabilities:** Despite Firebase's robust authentication services, our application may remain susceptible to security vulnerabilities such as data breaches or unauthorized access. Continuous monitoring, updates, and adherence to best security practices are essential to mitigate these risks effectively.
6. **Resource Intensiveness:** Utilizing multiple technologies like React Native, Firebase, imposes resource requirements on users' devices, including memory and processing power. This could pose challenges for older or low-end devices, impacting overall performance and user satisfaction.
   1. **Proposed Solutions**

To improve spam detection in chat applications, combining rule-based filtering with machine learning, leveraging unsupervised algorithms for emerging spam patterns, and implementing lightweight models for mobile devices are viable solutions. Enhancing user privacy through local processing or encryption techniques is also crucial. These approaches aim to boost detection accuracy while addressing concerns about computational resources and privacy.

**CHAPTER 2 :**

**2.1 Introduction**

**LITERATURE SURVEY**

The literature surrounding the integration of spam detection in chat applications using React Native and machine learning offers valuable insights into addressing modern communication challenges. Existing studies explore various methodologies to enhance spam detection accuracy while maintaining user privacy and application efficiency. Rule-based filtering, machine learning techniques, and hybrid approaches have been extensively researched to mitigate the evolving nature of spam. Additionally, advancements in unsupervised learning algorithms and lightweight model implementations cater to the constraints of mobile devices. This literature survey aims to synthesize the existing research landscape, identify gaps, and propose innovative solutions to improve spam detection efficacy in chat applications.

* 1. **Literature Survey**

**• In paper [1]** Randell mentions the history of the internet. All the political, technical and social developments that led to the development of internet are discussed in his The Soul of Internet. In his text, he mentions his thoughts about the social media applications that would run on the internet. It covers the interviews of the great personalities that were behind the internet technology.

• **In paper [2]** The process of multithreading and its benefits are mentioned in Intel Hyper Threading Technology. The authors have explained the process multithreading and hyperthreading, its advantages and disadvantages and compared the results of CPU performances with one thread and multiple threads running on a single CPU core. Multithreading is used in our project for performing the sender and receiver tasks concurrently.

• **In paper [3]** Michael Hauben, an internet theorist and author in his text described the social impact of internet. He developed the term and concept of ‘Netizen’ which means a citizen of the net or an internet user who actively contributes to the development of the net. He mentions the increase of popularity and usage of chat applications due to users being free and fearless to communicate with others.

• **In paper [4]** Richard Stevens provides a guide on Unix Network Programming using sockets and Network APIs. A socket object represents a low-level access point to an IP Stack. A socket can send and receive data which forms the basis of our project,for data exchange. A socket can be open or closed. This book is a must read to understand the network programming in order to build web server apps, client-server apps or any other network apps.

• **In paper [5]** Vincent Cerf and Robert Kahn provided a guide on the TCP protocol. This protocol is used by World Wide Web, File Transfer Protocol, peer to peer file sharing and streaming media. SSL/TLS runs on top of this protocol. TCP provides a reliable, error checked and ordered transfer of bytes of data. This makes it an important part in the working of our application.

**CHAPTER 3 :**

**PRESENT WORK CARRIED OUT**

The project focuses on developing a chat application with real-time messaging capabilities and integrated spam detection. React Native is used for building the mobile application, and Firebase provides backend services such as authentication, database management, and real-time updates. A machine learning-based spam detection mechanism is implemented to filter out unwanted or harmful messages.

**Technologies Used**

* **React Native**: Framework for building cross-platform mobile applications.
* **Firebase**: Backend services including authentication, Firestore database, and real-time messaging.
* **Machine Learning Libraries**: TensorFlow, Scikit-learn, or similar libraries for building and training spam detection models.

**Key Components and Their Roles**

**1. React Native Application**

**Authentication**

* **Firebase Authentication**: Manages user sign-up, login, and authentication state to ensure only authenticated users can access the chat functionalities.

**Chat Interface**

* **Chat Screen**: Displays messages and allows users to send new messages. Integrates with Firebase Firestore for real-time message storage and retrieval.

**2. Firebase Backend**

**Firestore Database**

* **Messages Collection**: Stores all chat messages with metadata such as timestamp, sender information, and message content.
* **Users Collection**: Maintains user profiles and authentication details.

**Real-time Updates**

* **Real-time Database**: Ensures that messages are instantly updated across all devices, providing a seamless chat experience.

**3. Spam Detection**

**Machine Learning-based Spam Detection**

* **Model Training**: Develop and train a spam detection model using historical chat data. The model can be trained on various features extracted from the text messages, such as word embeddings, TF-IDF scores, or other NLP-based features.
* **Model Deployment**: Deploy the trained model in the chat application to classify incoming messages in real-time.

**Text Preprocessing and Feature Extraction**

* **NLP Techniques**: Use NLP libraries to preprocess text data (e.g., tokenization, stopword removal, stemming) and extract features for model training. These features are crucial for the accuracy and performance of the spam detection model.

**4. Data Handling**

**Data Storage**

* **Training Data**: Store historical chat data and labels (spam or not spam) for training the spam detection model. This data can be stored in CSV files or directly in Firebase.

**5. Machine Learning Implementation**

**Model Development**

* **Feature Engineering**: Extract relevant features from the text messages using NLP techniques. These features are used as input for the machine learning model.
* **Model Selection**: Choose an appropriate machine learning algorithm (e.g., logistic regression, decision tree, neural network) based on the complexity and requirements of the spam detection task.
* **Training and Validation**: Train the model on labeled data and validate its performance using metrics such as accuracy, precision, recall, and F1-score.

**Model Integration**

* **Real-time Inference**: Integrate the trained model into the React Native application to classify messages as they are sent. If a message is detected as spam, it is blocked or flagged accordingly.

**Ethical Considerations**

* **Respect for User Privacy**: Ensure that user data is handled securely and in compliance with privacy regulations. Avoid storing sensitive information unnecessarily.
* **Respecting Terms of Service**: Adhere to Flipkart's terms of service and robots.txt file if scraping data for training purposes.

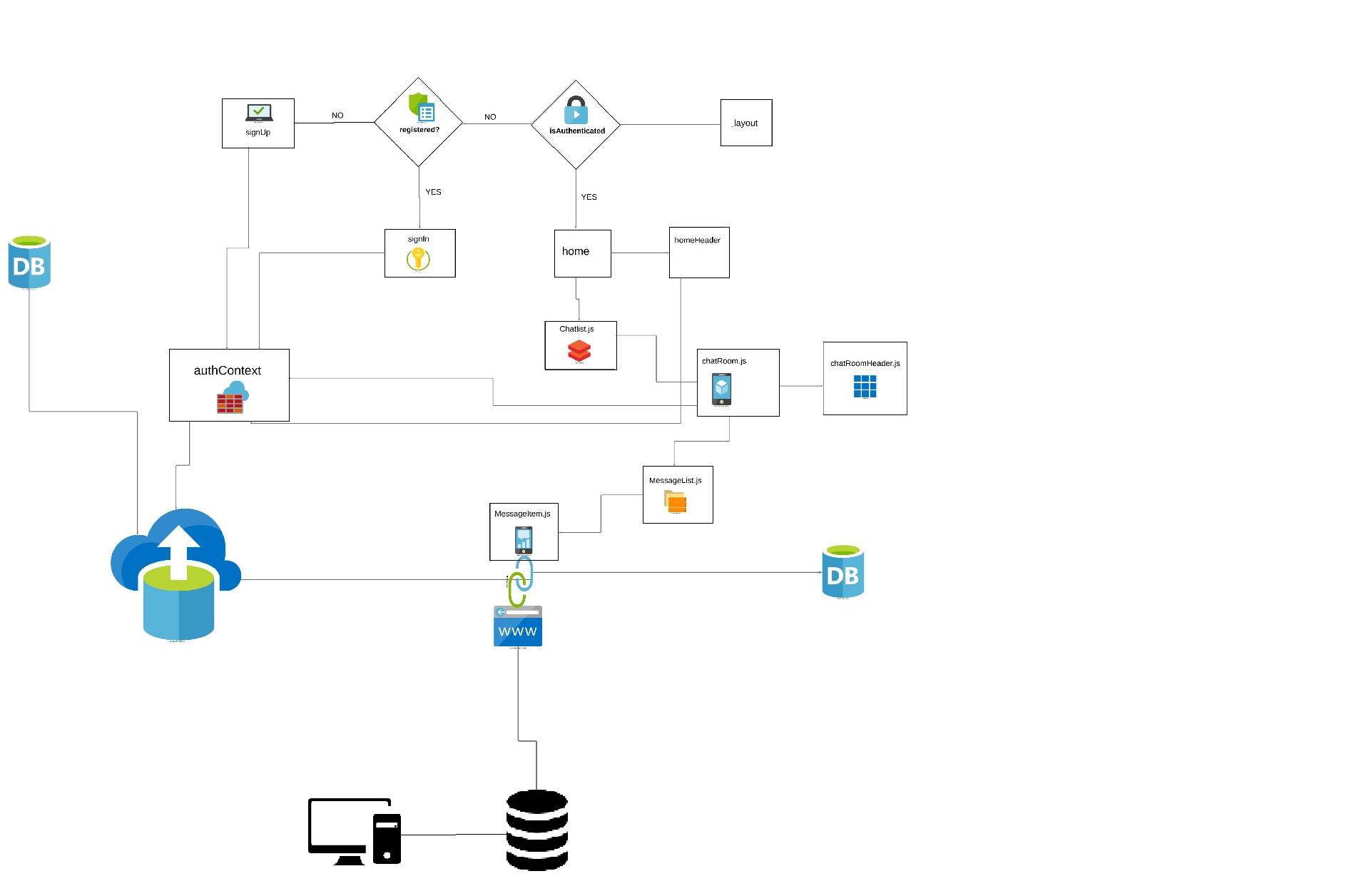
**Challenges and Solutions**

* **Handling Website Changes**: Be prepared to update the web scraper if Flipkart changes its website structure or introduces anti-scraping measures.
* **Dealing with Large Datasets**: Optimize the data scraping and processing pipeline to handle large volumes of data efficiently.
* **Model Performance**: Continuously monitor and improve the spam detection model to maintain high accuracy and low false positive rates.

**Summary**

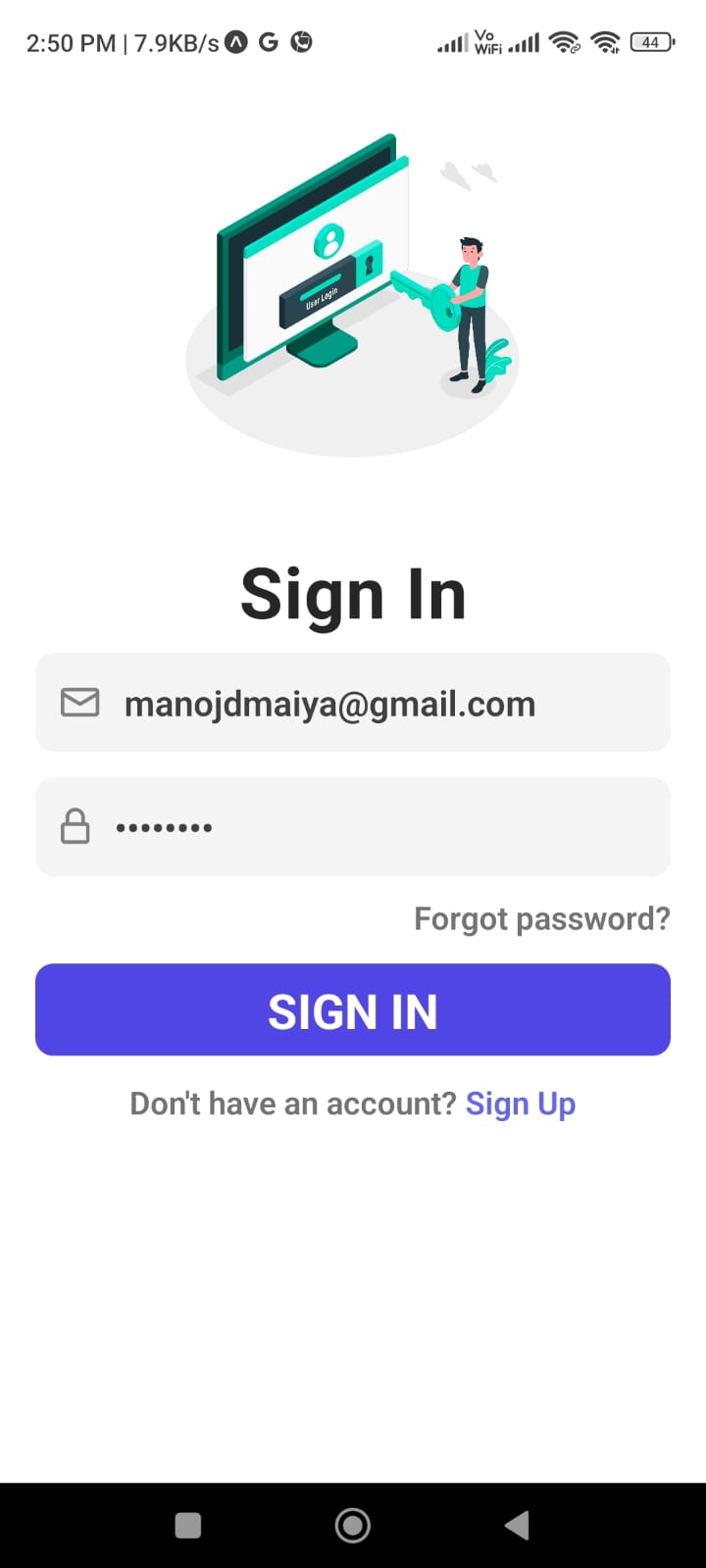
The chat application with spam detection involves building a robust messaging platform using React Native and Firebase, with an integrated machine learning model for spam detection. By leveraging NLP techniques and machine learning libraries, the project aims to create a secure and user-friendly chat experience. Ethical considerations and potential challenges are addressed to ensure the project's success and sustainability.

**3.1 SYSTEM DESIGN**

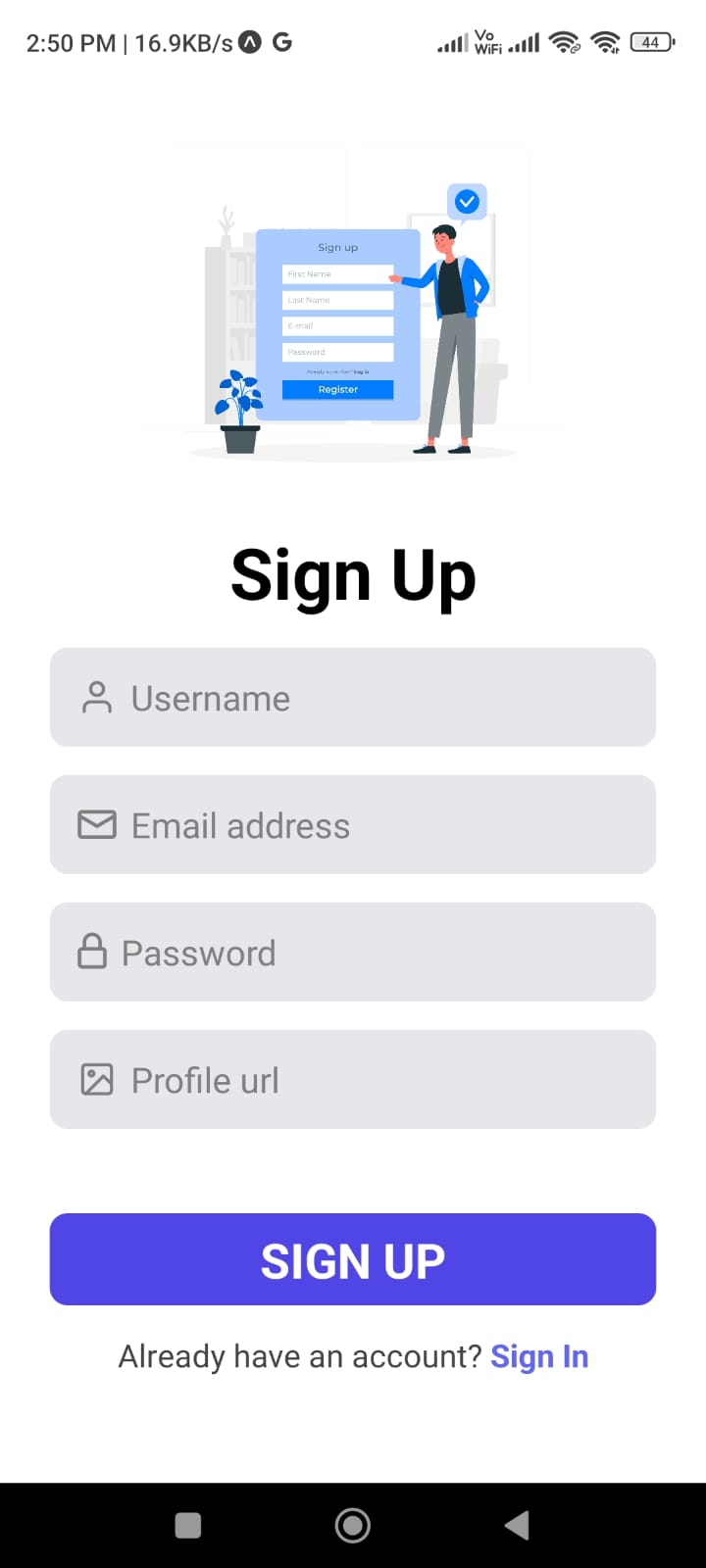
****

**CHAPTER 4 :**

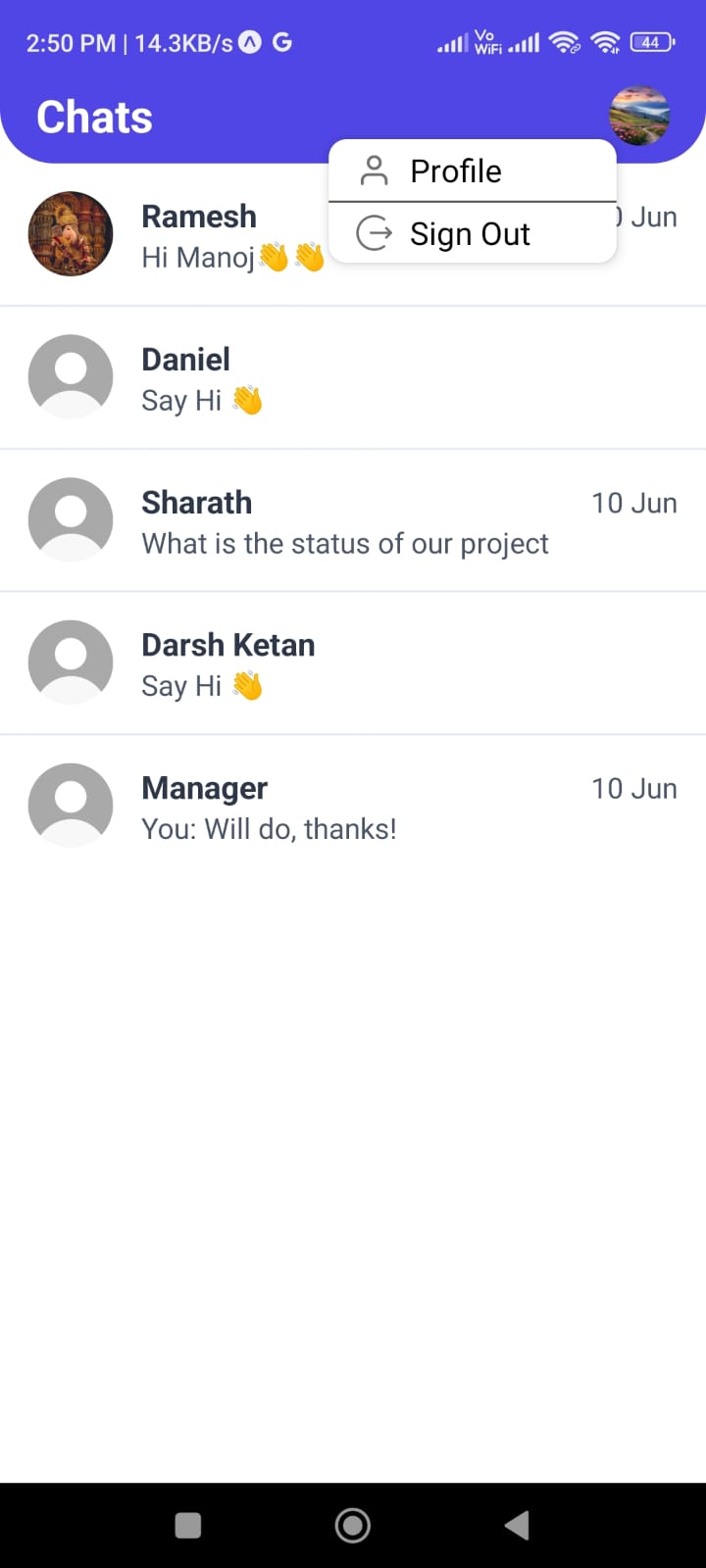
**RESULTS AND DISCUSSION**

****

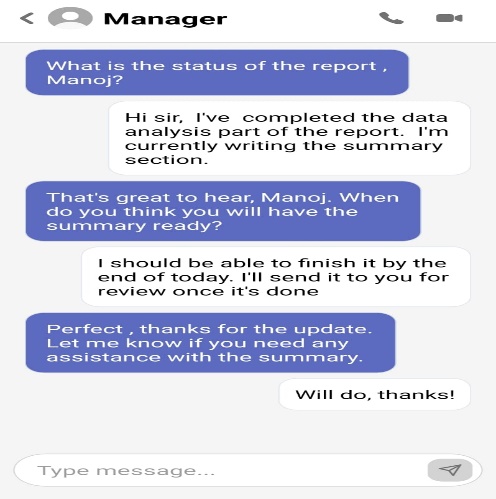
**SIGN IN WINDOW**



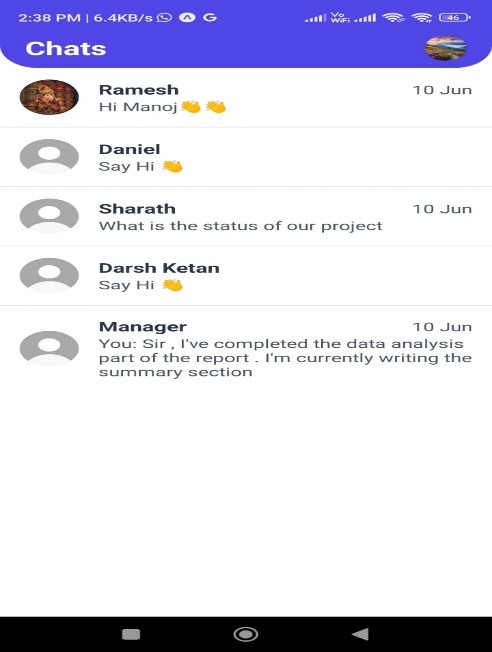
**SIGN UP WINDOW**

****

**PROFILE VIEW & LOGOUT WINDOW**

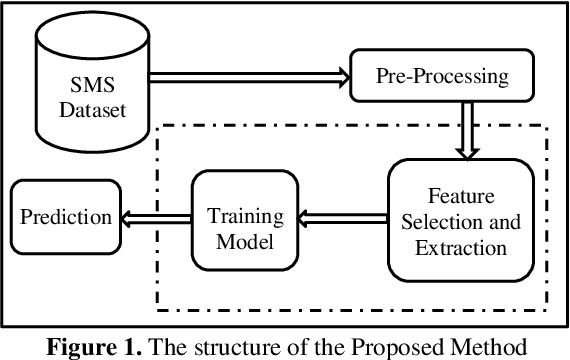
****

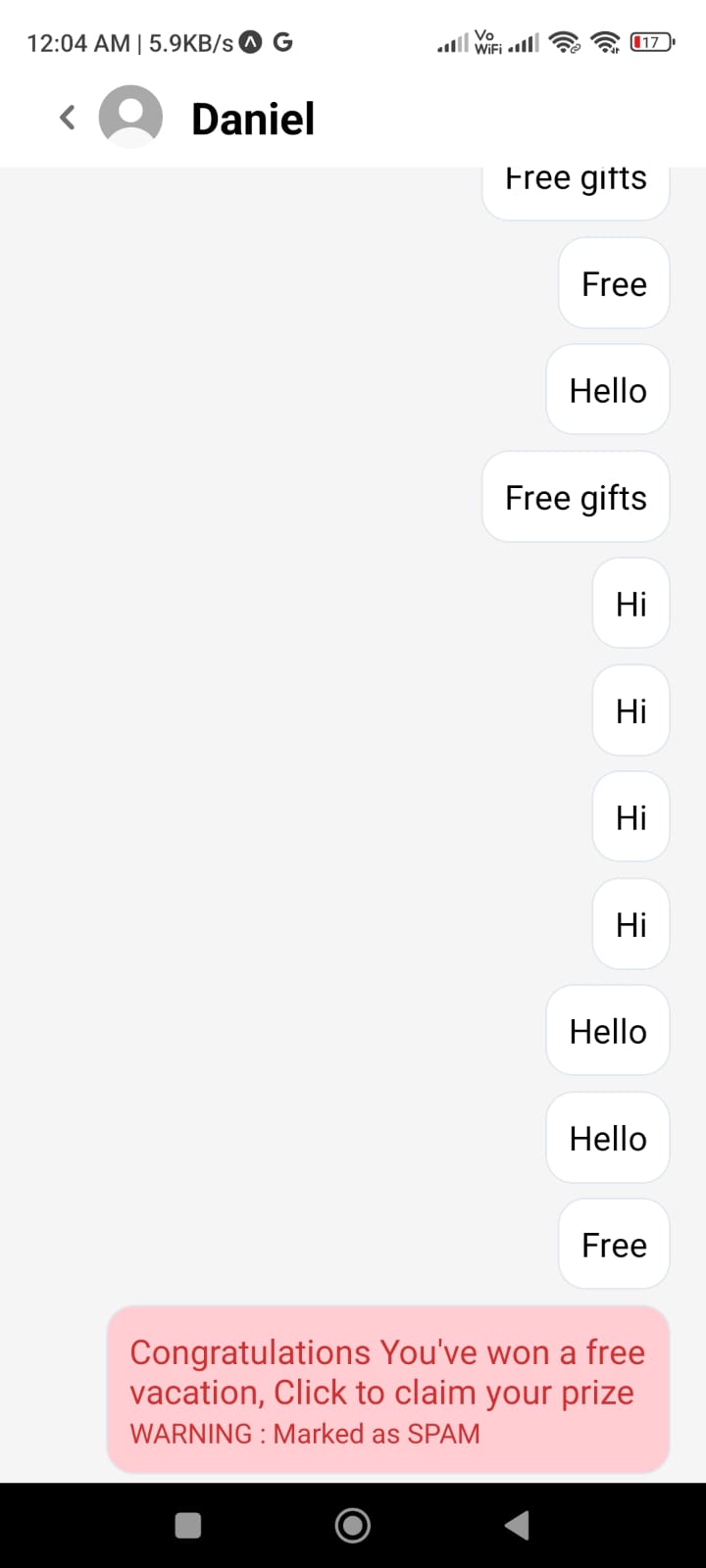
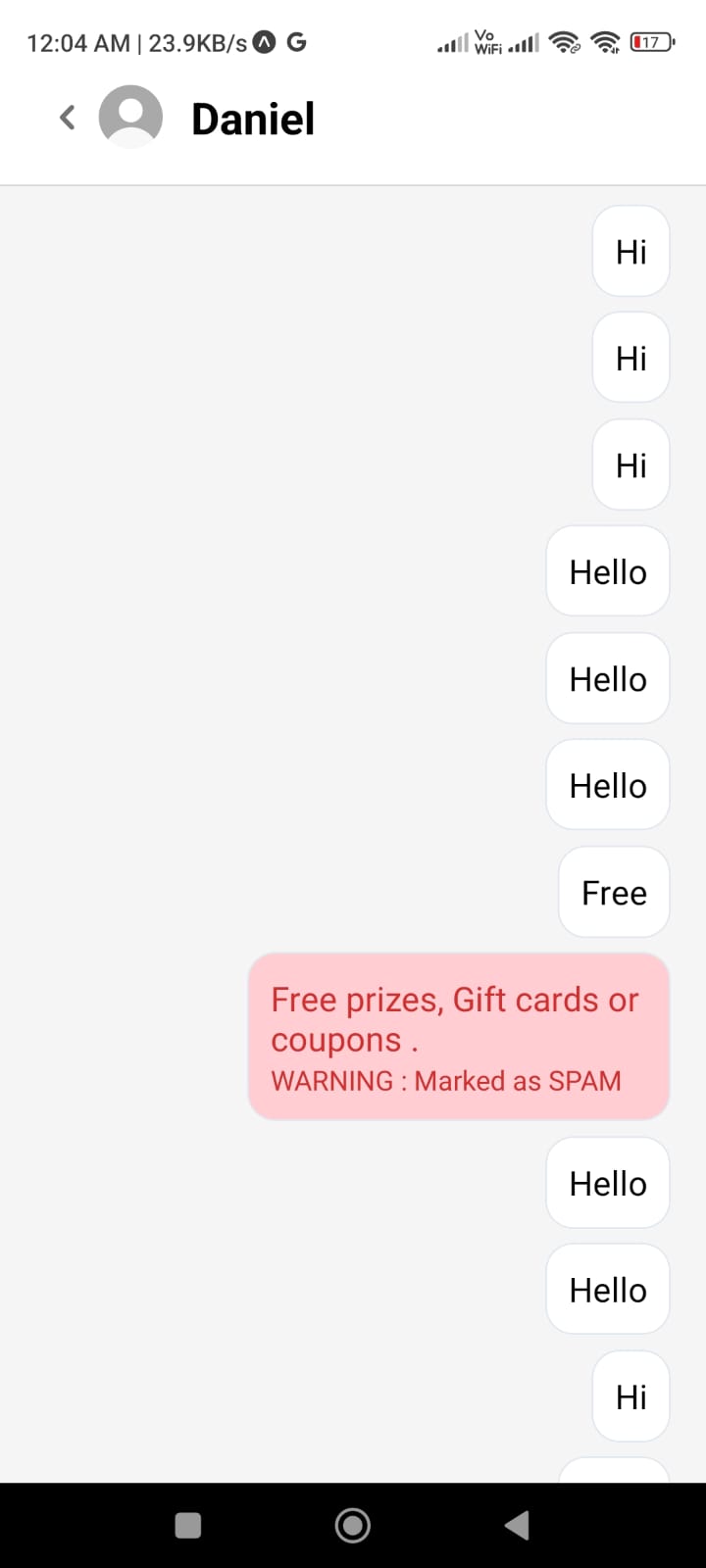
**CHAT WINDOW**

****

**HOME WINDOW**

**SPAM DETECTION**

****

** **

**CONCLUSION**

In conclusion, our project to develop a chat application using React Native and Firebase has successfully created a modern, cross-platform communication platform. Integrating advanced technologies has significantly enhanced user experience, security, and performance.

React Native enables seamless communication across iOS and Android devices, while Firebase Realtime Database ensures instant message delivery. Secure user authentication through Firebase Authentication safeguards user data, and spam detection mechanisms enhance interaction quality.

Our user-centric design has produced an intuitive, accessible interface. Scalability and performance optimization ensure smooth operation under high loads. We are committed to continuously enhancing our application and embracing emerging technologies to meet evolving user needs.

**FUTURE WORK**

* **Enhanced Machine Learning Integration**:

Improve spam detection accuracy and adaptability using advanced machine learning techniques.

* **Real-time Spam Detection**:

Develop mechanisms to identify spam in milliseconds for efficient message analysis.

* **Personalized Spam Detection**:

Customize spam detection algorithms to individual user preferences and communication patterns.

* **Privacy-preserving Techniques**:

Ensure user data privacy through methods like federated learning and differential privacy.

* **Integration with Chatbots**:

Automate spam reporting and mitigation by developing intelligent chatbots.

**REFERENCES**

**[1]** Modupe, A., Olugbara, O. O., & Ojo, S. O. (2014). Filtering of Mobile Short Messaging Communication Using Latent Dirichlet Allocation with Social Network Analysis. In G.-C. Yang, S.-I. Ao, & L. Gelman (Eds.), Transactions on Engineering Technologies: Special Volume of the World Congress on Engineering 2013 (pp. 671–686). Springer Science & Business.

**[2]** Shirani-Mehr, H. (2013). SMS Spam Detection using Machine Learning Approach.

**[3]**  Abdulhamid, S. M., et al. (2017). A Review on Mobile SMS Spam Filtering Techniques. IEEE Access, 5, 15650–15666.

**[4]**  Aski, A. S., & Sourati, N. K. (2016). Proposed Efficient Algorithm to Filter Spam Using Machine Learning Techniques. Pacific Science Review. Natural Science and Engineering, 18(2), 145–149.

**[5]** Narayan, A., & Saxena, P. (2013). The Curse of 140 Characters: Evaluating The Efficacy of SMS Spam Detection on Android (pp. 33–42).