

Kurdistan Regional Government – Iraq  
Ministry of Higher Education and Scientific Research  
Salahaddin University-Erbil College of Engineering  
Department of Software and Informatics



# Chat Application

A Project Submitted to the Department of Software and Informatics  
Engineering\ University of Salahaddin-Erbil  
In the Partial Fulfillment of the Requirement for the Degree of Bachelor of  
Science, in Software and Informatics Engineering

## Prepared By:

Koshyar Abdulrahman

Rawa Osman

## Project Supervisor:

Lec. Kanar Shukr

April, 2024

## **ABSTRACT**

This project is to create a seamless and efficient chat platform that offers real-time messaging and calling capabilities while ensuring a user-friendly interface.

The development process involves utilizing Flutter for building the front-end interface, Firebase for managing user authentication, data storage, and real-time database functionalities, Agora for implementing real-time audio and video communication features, and Figma for designing the user interface and user experience. The project will follow a systematic approach, including requirement analysis, design implementation, coding, testing, and deployment.

The implications of this project extend to the field of mobile application development, particularly in the realm of chat applications. This project highlights the potential for creating innovative and user-centric chat platforms.

At the end the project aims to develop a robust chat application offering real-time messaging and communication features. Through systematic development and user feedback analysis.

## **Dedication**

Our parents who have taught us the way of live, brothers, sisters. And our supervisor who tell us science with all other teachers, and our dear friends that help us in preparing this project, and those who want to learn.

## **SUPERVISOR CERTIFICATION**

I certify that the engineering project titled " " was done under my supervision at the department of Software and Informatics Engineering, College of Engineering, Salahaddin University –Erbil. In the partial fulfillment of ‘The requirement for the degree of Bachelor of Science in Software and Informatics Engineering’.

**Supervisor**

Signature:

Name: Kanar Shukr

Date: ?/4/2024

# LIST OF CONTENTS

1	Introduction .....	1
1.1	Background .....	1
1.2	Problem Statement .....	1
1.3	Aims and Objectives .....	2
1.4	Chapter Summery.....	4
2	Methodology.....	5
2.1	Software Model.....	5
2.1.1	Requirement gathering and Analysis .....	6
2.1.2	Software Design.....	8
2.1.3	Implementation .....	9
2.1.4	Testing.....	11
3	Analysis and Design.....	12
3.1	Use Cases .....	12
3.2	NoSQL Collectives Diagram .....	14
4	Results and discussions .....	17
4.1	Creating Account .....	17
4.2	(Short Message/Messaging Service) SMS Verification.....	18
4.3	Home Screen .....	19
4.4	Creating Groups .....	20
5	Conclusion and Recommendations .....	21
5.1	Conclusion: .....	21
5.2	Recommendations:.....	21

## LIST OF FIGURES

Figure 2.1 Agile Methodology .....	6
Figure 3.1 Use Case diagram .....	12
Figure 3.2 Use Case Diagram .....	13
Figure 3.3 User Collection Diagram .....	14
Figure 3.4 Group Collection Diagram .....	15
Figure 3.5 Call Collection .....	16
Figure 3.6 Status Collection .....	16
Figure 4.1 Create Account Image 1 .....	17
Figure 4.2 Create Account Image 2 .....	17
Figure 4.3 Verification .....	18
Figure 4.4 Home Screen.....	19
Figure 4.5 Creating Group .....	20

## **LIST OF ABBRIVATION**

<b>GIF</b>	Graphics Interchange Format
<b>IOS</b>	iPhone Operating System
<b>NoSQL</b>	Not only SQL
<b>SQL</b>	Structured Query Language
<b>OOP</b>	Object Oriented Programming
<b>UI</b>	User Interface
<b>SMS</b>	Short Message/Messaging Service
<b>SRS</b>	Software Requirement Specification

# CHAPTER ONE

## 1 INTRODUCTION

In the ever-evolving landscape of communication technologies, the development of efficient and user-friendly chat applications has become increasingly vital. This introduction provides an overview of the project, which focuses on the creation of a modern chat application utilizing cutting-edge technologies.

### 1.1 BACKGROUND

With the proliferation of smartphones and the growing need for instant communication, the demand for intuitive and feature-rich chat applications has surged. Traditional messaging platforms often lack real-time communication capabilities and user-friendly interfaces, prompting the need for innovative solutions.

### 1.2 PROBLEM STATEMENT

- **Problem**
  - The problem being addressed is the need for efficient and seamless communication in today's digital landscape. Traditional communication methods often lack the immediacy and versatility required for modern interactions. People want to communicate in real-time, share various types of multimedia content, and do so across different devices and platforms effortlessly.



- **Importance of the Problem**

- In today's fast-paced world, effective communication is essential for personal, professional, and social interactions. Whether it's coordinating work projects, staying connected with friends and family, or accessing customer support services, communication plays a vital role in daily life. However, existing communication tools may be limited in their capabilities or may not offer the level of privacy and security users require.

- **Solution**

- By developing a chat application that addresses these challenges, the aim is to enhance the way people communicate. Importance of the real-time communication has been recognized, multimedia integration, and cross-platform compatibility in meeting users' evolving needs. The goal is to provide a solution that offers a seamless and enjoyable communication experience while prioritizing user privacy and data security. By solving these issues, the project contribute to improving productivity, fostering connections, and enriching people's lives in an increasingly connected world.

### **1.3 AIMS AND OBJECTIVES**

- **Real-time Communication:**

- Real-time communication in a chat application means that messages are sent and received with minimal delay, mimicking face-to-face conversations. This not only includes text messages but also real-time voice and video calls.

- Immediate message delivery is critical for user satisfaction and usability in communication apps. It enhances the interaction experience, making it feel more natural and engaging. It's especially crucial for business communications, customer support, and personal chats where timely responses are expected.
- **Integrating a Wide Array of Multimedia:**
  - Multimedia integration includes the ability to share various types of content such as images, videos, audio files, and potentially even interactive elements like (Graphics Interchange Format) GIFs. This feature requires robust file upload and download capabilities, efficient data handling, and possibly transcoding features to adapt media for different devices.
  - Offering multimedia sharing capabilities enhances the expressiveness and functionality of the chat application. Users expect to be able to share more than just text in today's digital environment—they want a rich media experience that can help convey emotions, share information more effectively, and keep the conversation engaging.
- **Cross-Platform Application:**
  - A cross-platform application functions across multiple operating systems and device types (such as (iPhone Operating System) iOS, Android, Windows, macOS) with a consistent user interface and experience. Developing such applications often involves using frameworks that support multiple platforms or developing

separate apps for each platform with shared backend infrastructure.

- Cross-platform compatibility ensures that the app can reach a wider audience. It allows users on different devices and operating systems to interact without barriers, which is particularly important in a globally connected world. It reduces development and maintenance costs compared to building and supporting multiple native apps for each platform.

## **1.4 CHAPTER SUMMERY**

In the subsequent chapters, this report will articulate the journey of the chat application project from conception to realization. Chapter 2 delves into the methodology, detailing the strategies and procedures followed during the development process. Chapter 3 provides a design for the application. In Chapter 4, the results and discussions shed light on the effectiveness of the application and its reception by users. Finally, Chapter 5 encapsulates the conclusions drawn from the project and offers recommendations for future enhancements.

# **CHAPTER TWO**

## **2 METHODOLOGY**

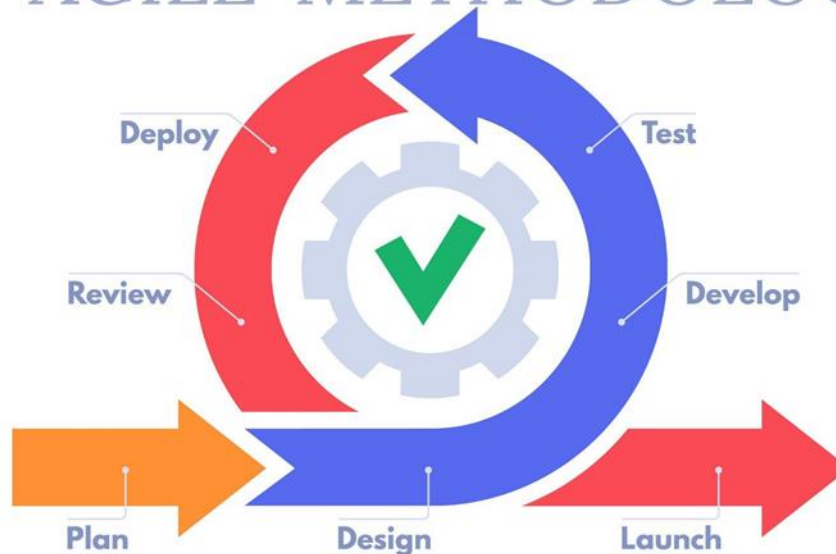
This chapter outlines the methodology employed in the development of the chat application project. It discusses the strategies and work phases undertaken throughout the implementation process. While software engineering methodologies often include diagrams and technical details, this chapter focuses solely on describing the actions and tasks performed during the project.

### **2.1 SOFTWARE MODEL**

Agile software model has been used due to its agility and flexibility. Unlike the waterfall model, agile allows for gathering main requirements at the beginning while accommodating additional requirements throughout the project phases. (Layton, 2020)

The agile software model consists of six phases: requirements, design, development, testing, deployment, and review. After the review phase, we return to the requirements gathering phase, creating a cyclic process that continues until the project is completed. The number of cycles may vary depending on the project and team's needs, typically averaging around three cycles. (Layton, 2020)

# AGILE METHODOLOGY



*Figure 2.1 Agile Methodology*

## **2.1.1 REQUIREMENT GATHERING AND ANALYSIS**

Requirements analysis is a set of operations that helps define user's expectations of the application you are building or modifying, Software engineering professionals sometimes call it requirement engineering, requirements capturing or requirement gathering. (Indeed Editorial Team, 2023).

The process involves analyzing, documenting, validating and managing system or software requirements, Requirements analysis involves various tasks that help engineers understand stakeholder demands and explain them in simple and visual ways. It is essential to a software or system project's success. (Indeed Editorial Team, 2023).

Based on gathered requirement and analysis, the following requirements have emerged as critical focal points for the project:

1. **Preference for NoSQL Databases:** (Not only SQL) NoSQL databases are increasingly favored over (Structured Query Language) SQL

databases for applications requiring real-time communication. This preference is attributed to the flexible data models, scalability, and faster data retrieval capabilities that NoSQL databases offer. These features are essential for handling large volumes of unstructured data and for supporting the high concurrency levels required by real-time applications.

2. **Utilization of Agora for Real-Time Communication:** In the realm of real-time communication, frameworks such as Agora are extensively utilized. Agora's framework is chosen for its ability to provide high-quality voice and video communication solutions that are both reliable and easy to integrate. The use of such a framework allows for the implementation of advanced communication features while maintaining low latency and high levels of performance.
3. **Employment of Google's Firebase:** Google's Firebase is employed as the foundational technology for the chat application. Known for its robustness and ease of use, Firebase offers a comprehensive suite of tools that support the development of high-quality, interactive chat applications. Firebase is particularly renowned for features like real-time database updates, user authentication, and hosting, which are integral to developing modern chat applications.
4. **Implementation of a User-Friendly Interface:** A user-friendly interface is implemented to ensure that the application is accessible and easy to navigate for all users. This involves the design of intuitive layouts, clear visual elements, and responsive controls that enhance user

interaction without overwhelming or confusing the user. The goal of such an interface design is to improve user satisfaction and engagement by making the technology accessible to users of varying technical proficiency.

### **2.1.2 SOFTWARE DESIGN**

Software design is a method that converts user requirements into a suitable form for the programmer to employ in software coding and implementation. It is concerned with converting the client's requirements as defined in the SRS (Software Requirement Specification) document into a form that can be easily implemented using a programming language. (Sushant, 2024)

Following tool has been used for Software design:

- 1. Use of Figma for Designing User Interface and User Experience:** Figma has been utilized for designing the user interface and enhancing user experience. This design tool is selected due to its collaborative capabilities and robust feature set, which allow for real-time cooperation among design teams. The use of Figma facilitates the creation of interactive and visually appealing designs, which can be dynamically tested and refined to meet the specific needs of the user base.
- 2. Employment of Draw.io for Use Case Diagrams:** The Draw.io website has been employed for drawing use case diagrams for the chat application. Draw.io is chosen for its accessibility and ease of use, providing designers and developers with a versatile tool for creating

detailed diagrams that help visualize complex interactions. This aids in the systematic representation of functional requirements, thereby enhancing clarity and understanding throughout the development process.

### **2.1.3 IMPLEMENTATION**

During the implementation phase, the following tools has been utilized:

1. **Flutter:** Flutter has been adopted as the primary framework for the development of the user interface. This choice is due to Flutter's ability to enable cross-platform development while maintaining native performance and visual consistency across different operating systems.
2. **Dart Programming Language:** The Dart programming language has been utilized predominantly in the development process. Dart's compatibility with Flutter and its features for building structured, scalable applications make it suitable for this project.
3. **Utilization of Go Language:** A limited amount of Go language has been incorporated primarily for leveraging Agora's capabilities. Go is utilized due to its efficiency in handling concurrent operations and networked applications, enhancing the real-time communication features provided by Agora.
4. **Android Studio and Physical Devices:** Android Studio has been used as an emulator to test the application, alongside the use of physical



devices. This dual approach allows for thorough testing in simulated environments as well as real-world conditions, ensuring broader compatibility and usability.

5. **Object-Oriented and Functional Paradigms:** Both Object-Oriented Programming (Object Oriented Programming) OOP and functional paradigms have been employed to structure the software development. This mixed approach leverages the strengths of both paradigms, facilitating robust code organization and effective handling of the application state.
6. **Firestore NoSQL Database:** The Firestore NoSQL database has been used due to its real-time data synchronization capabilities, automatic scaling, and ease of integration with other Google services.
7. **Git and GitHub:** Git, along with GitHub, has been utilized for version control and source code management. This toolset supports collaborative development efforts, enabling tracking of changes, branching, and merging, which are essential for managing the developmental lifecycle of the application.
8. **Visual Studio Code:** Visual Studio Code has been chosen as the primary code editor for the project. Its versatility and support for multiple programming languages, along with powerful debugging tools and extensions, make it an ideal choice for a development environment.

#### 2.1.4 TESTING

**Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements. (Thomas, 2024)

Manual testing has been adopted as a primary testing method, conducted both on emulators and real devices. This approach ensures that various aspects of the application, such as performance, security, and user interface, are thoroughly examined. Emulators have been used to simulate different device environments and operating conditions, enabling developers to identify potential issues under controlled scenarios. Additionally, testing on real devices has been carried out to validate the application's functionality in real-world usage conditions. This dual testing strategy helps in uncovering (User Interface) UI flaws and security vulnerabilities, ensuring a robust and user-friendly application.

# CHAPTER THREE

## 3 ANALYSIS AND DESIGN

This chapter presents detailed designs and descriptions of different system components of the chat application project. It includes figures describing various aspects of the design components. The following are listed and described:

### 3.1 USE CASES

The use cases outlined below provide a detailed view of user activities within the application. They serve as practical examples that illustrate the typical interactions and behaviors of users as they navigate through various features of the app.

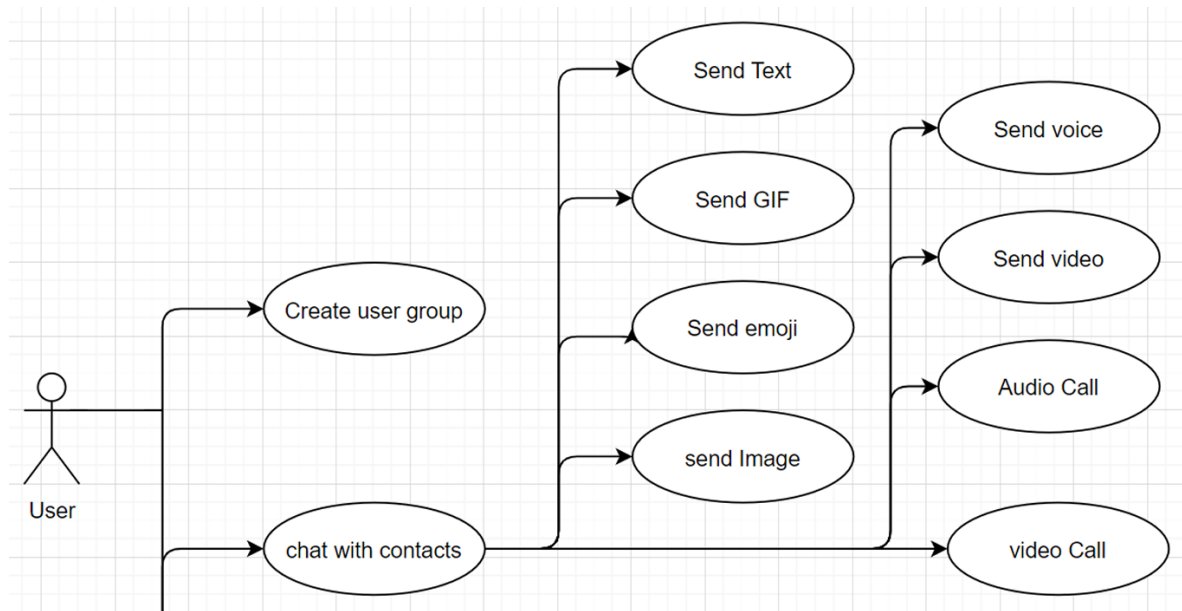


Figure 3.1 Use Case diagram

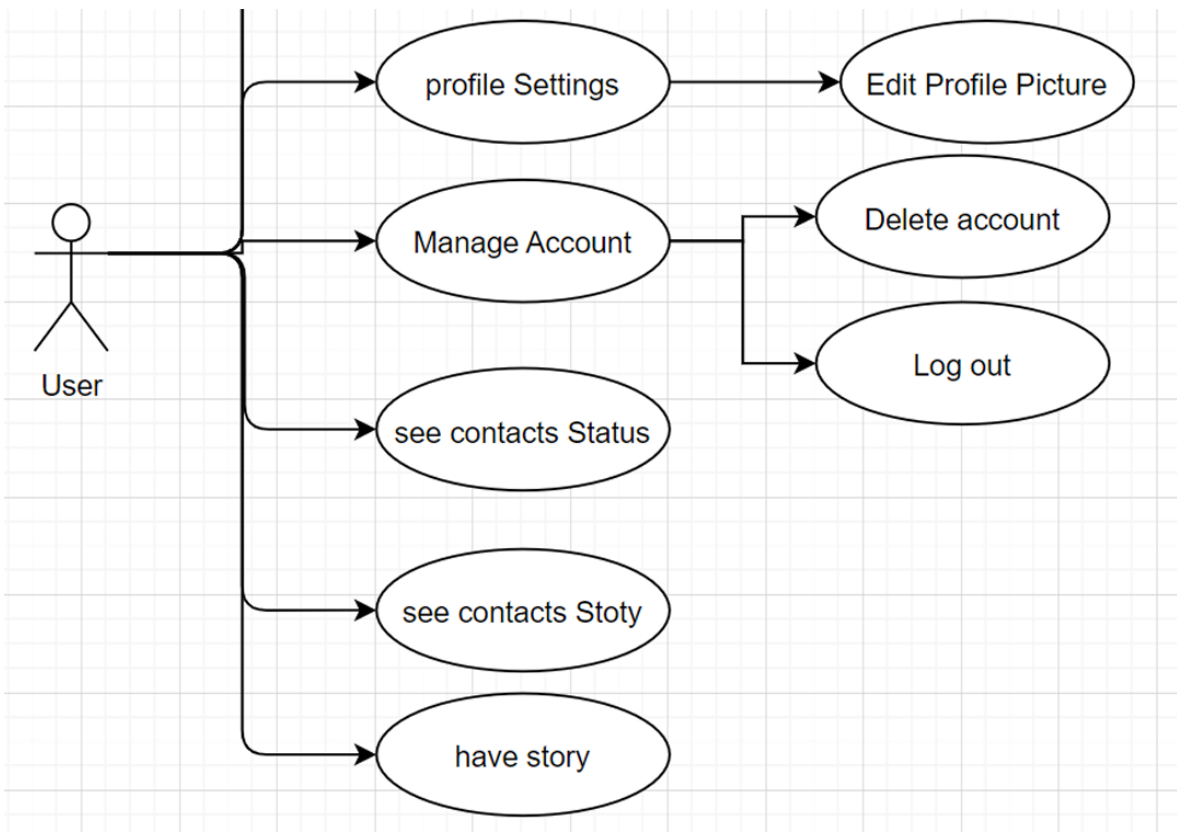


Figure 3.2 Use Case Diagram

### 3.2 NOSQL COLLECTIVES DIAGRAM

The diagrams provided herein illustrate the structure of NoSQL database collectives. These visual representations aim to clarify how data is organized and managed within NoSQL environments

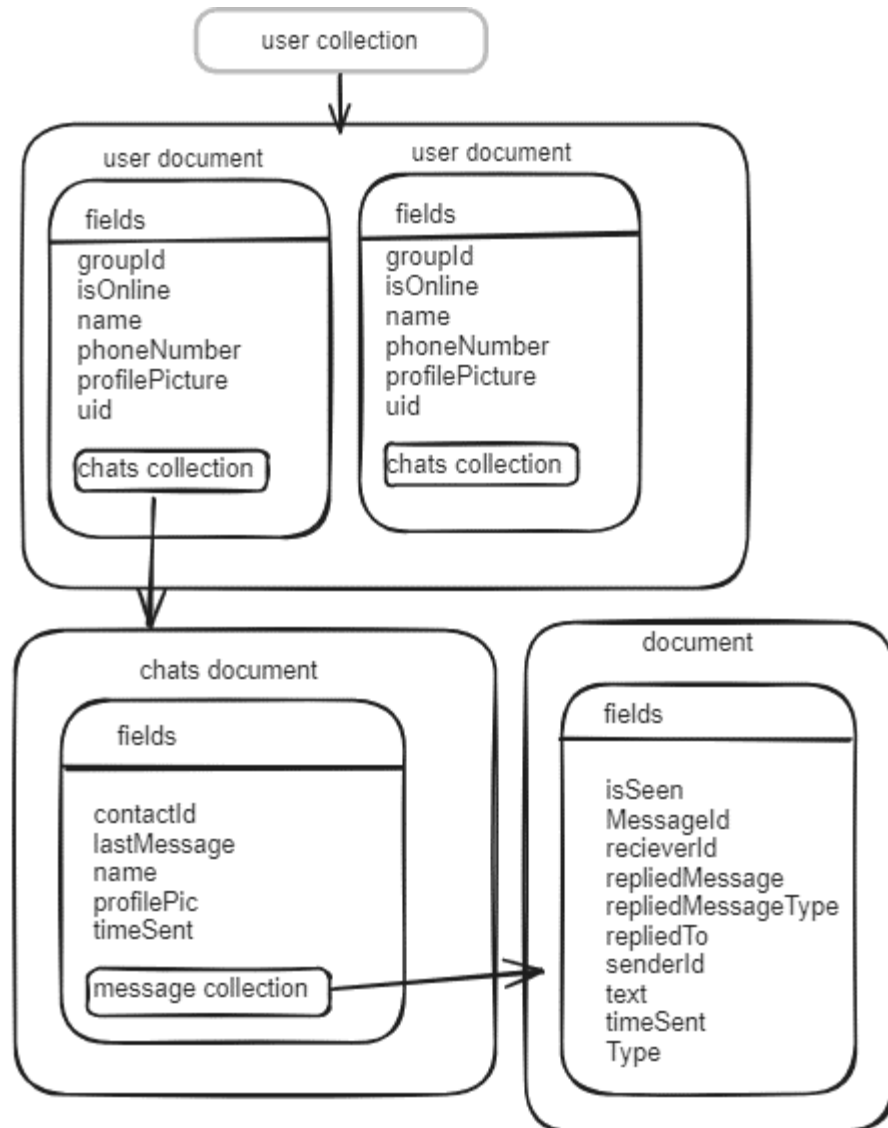


Figure 3.3 User Collection Diagram

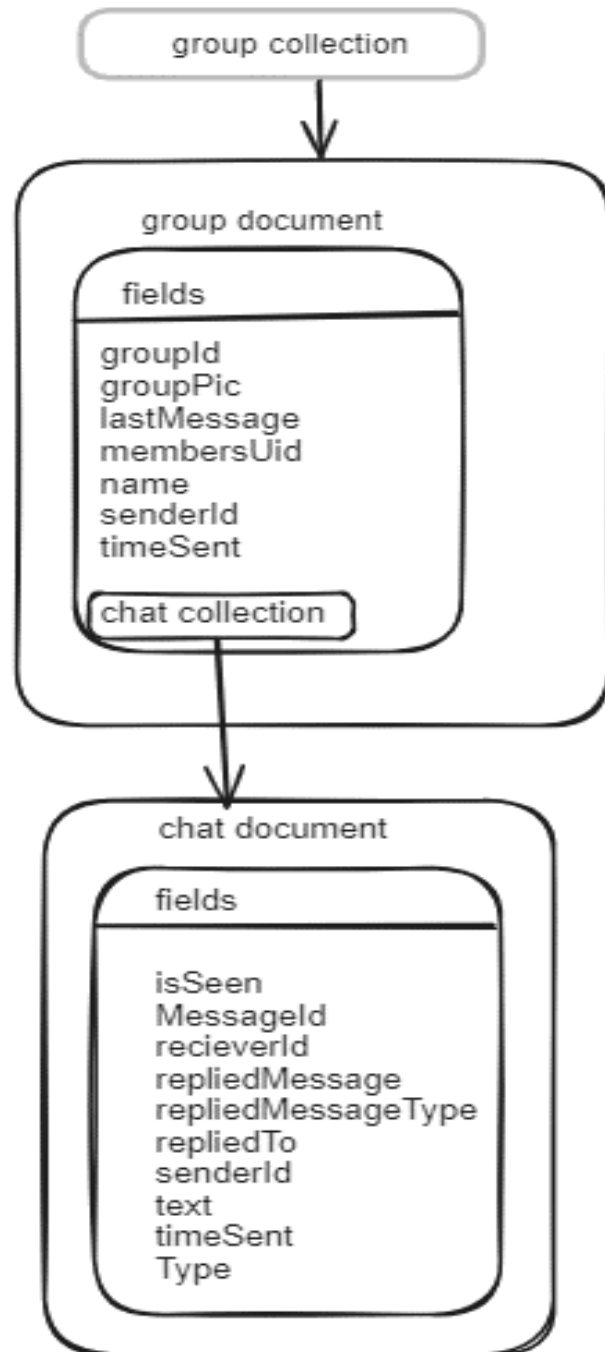


Figure 3.4 Group Collection Diagram

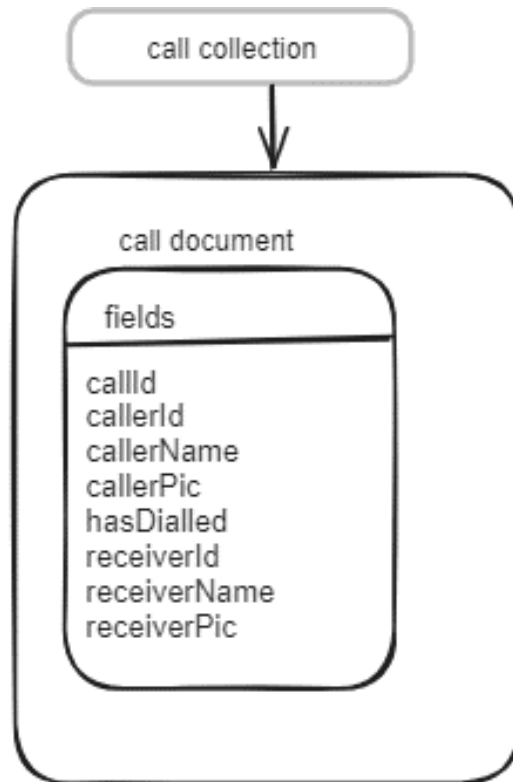


Figure 3.5 Call Collection

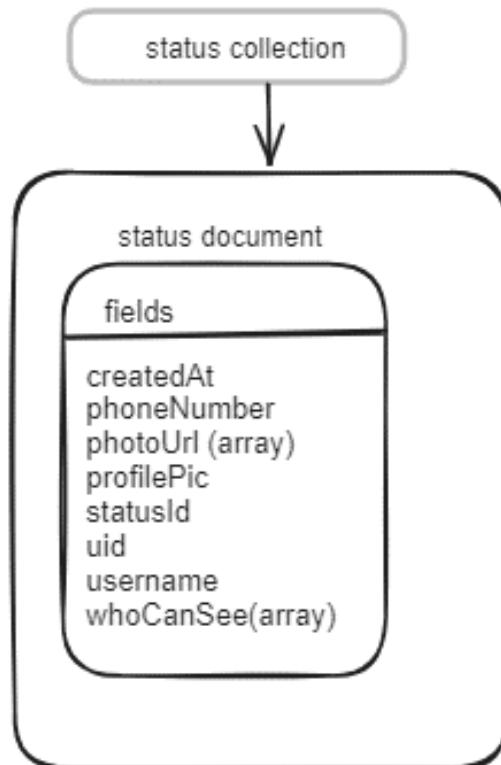


Figure 3.6 Status Collection

# CHAPTER FOUR

## 4 RESULTS AND DISCUSSIONS

This chapter presents the results obtained from the chat application project, organized in figures. The results are briefly summarized.

### 4.1 CREATING ACCOUNT



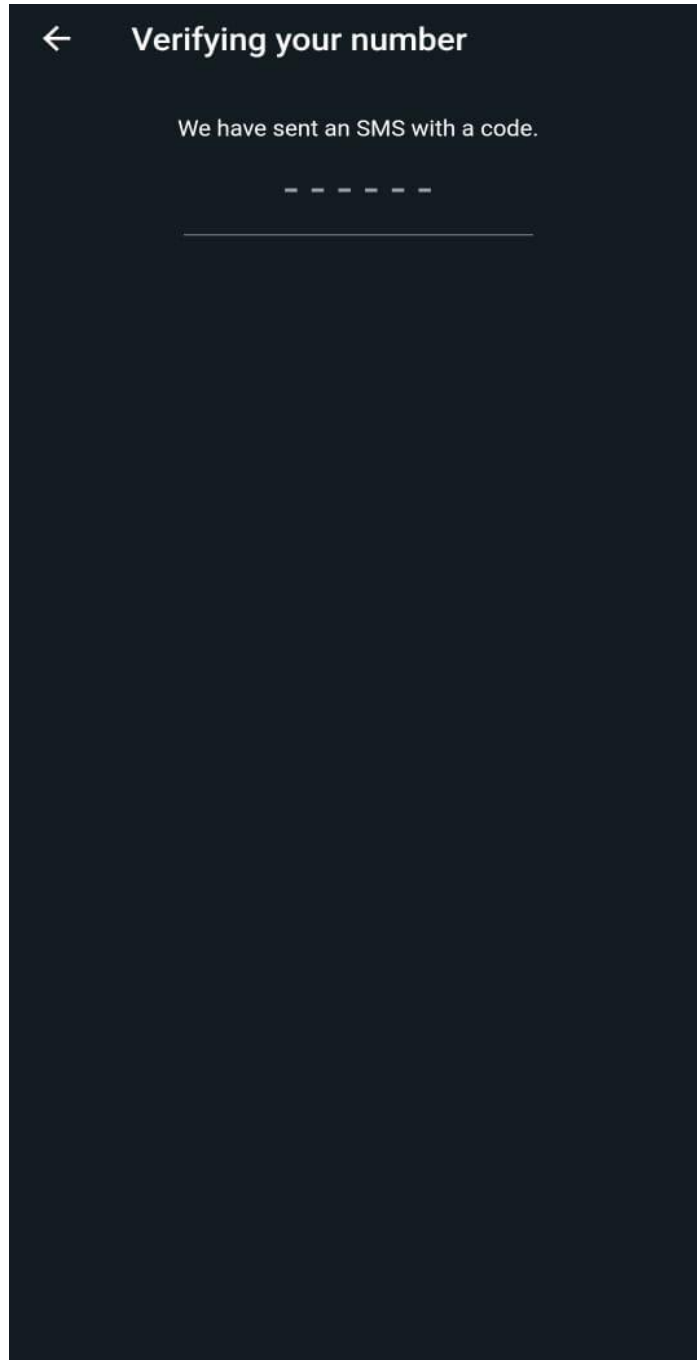
Figure 4.1 Create Account Image 1



Figure 4.2 Create Account Image 2



## 4.2 (SHORT MESSAGE/MESSAGING SERVICE) SMS VERIFICATION



*Figure 4.3 Verification*

## 4.3 HOME SCREEN

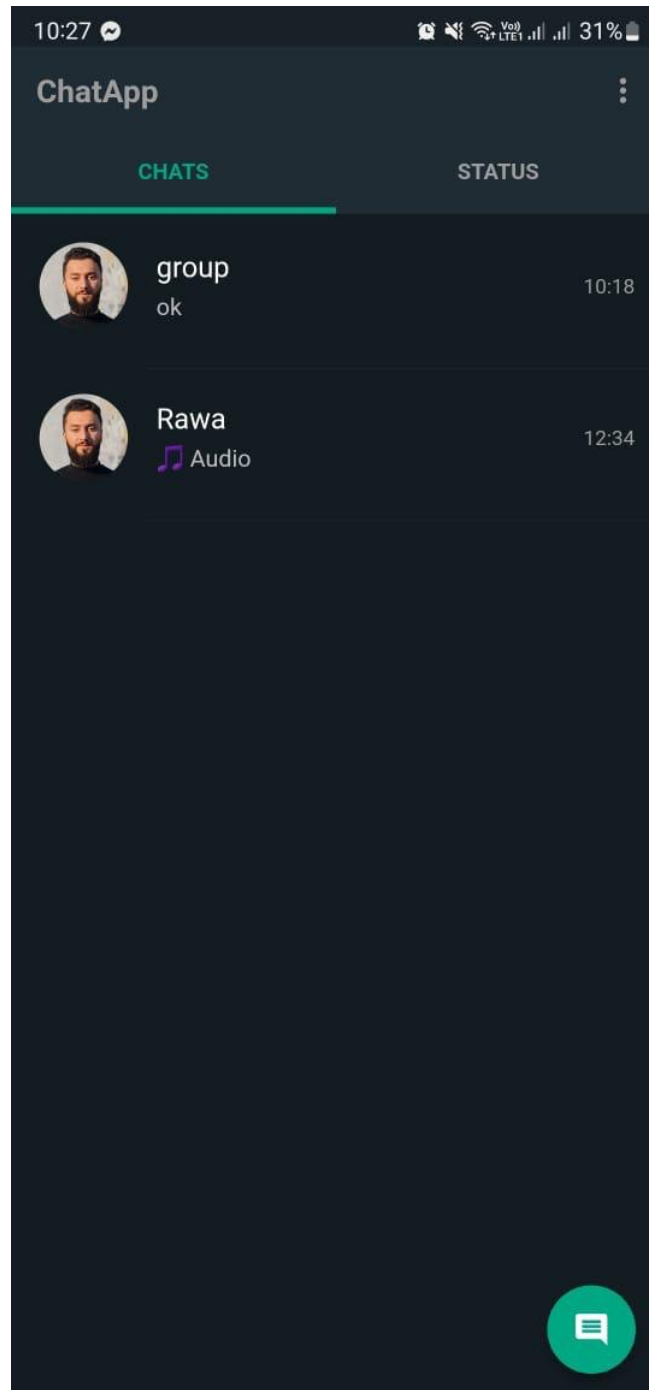


Figure 4.4 Home Screen

## 4.4 CREATING GROUPS

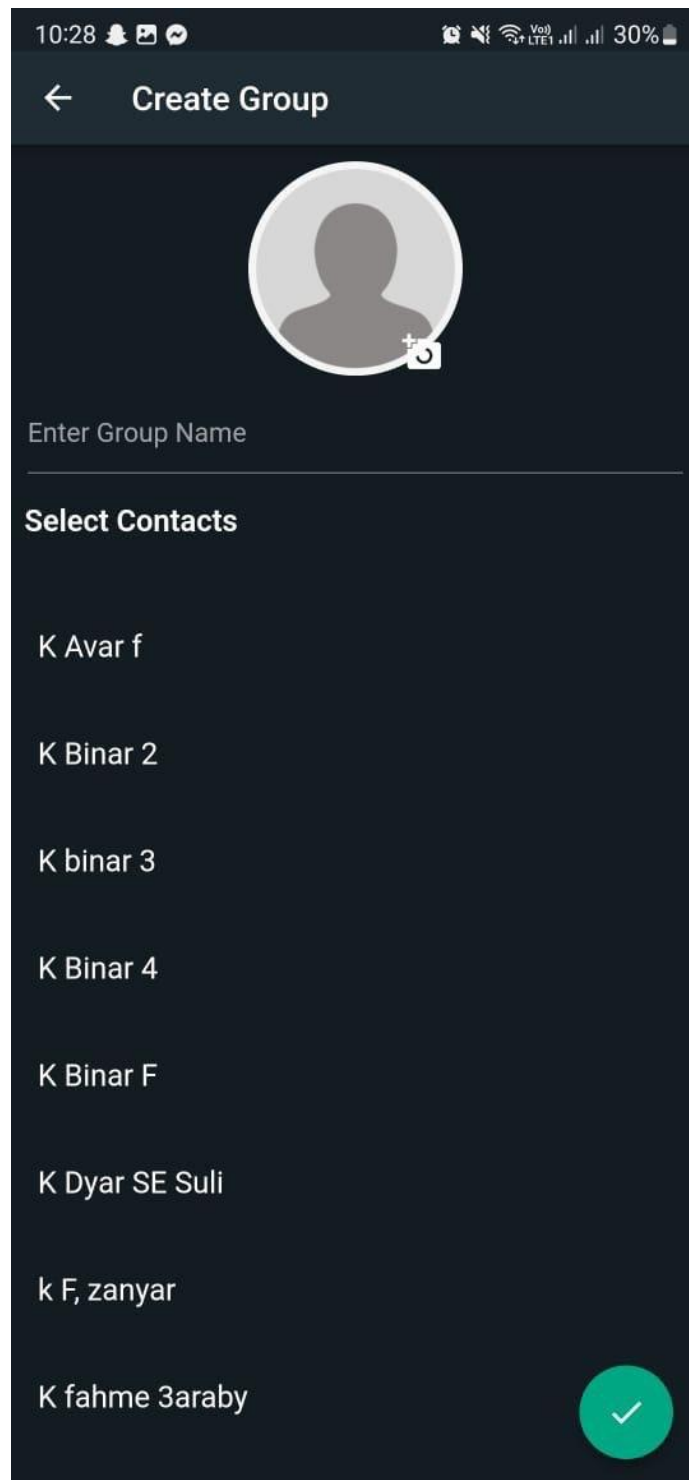


Figure 4.5 Creating Group

## CHAPTER FIVE

### 5 CONCLUSION AND RECOMMENDATIONS

#### 5.1 CONCLUSION:

- The chat application project has successfully achieved its objectives of developing a modern and user-friendly communication platform.
- Through the implementation of cutting-edge technologies such as Flutter, Firebase, and Agora, the application offers seamless real-time communication, robust security features, and an intuitive user interface.
- The project has demonstrated the feasibility and effectiveness of utilizing agile methodologies for software development, allowing for flexibility, adaptability, and iterative improvements throughout the development process.
- User feedback and testing results have provided valuable insights into usability, functionality, and performance, informing future enhancements and optimizations.

#### 5.2 RECOMMENDATIONS:

1. **Inclusion of the Kurdish Language:** Adding support for the Kurdish language would make the application more accessible to Kurdish-speaking users, thereby expanding the user base and enhancing linguistic inclusivity. This involves not only translating the user interface but also ensuring that input and output functionalities are fully adapted to handle the Kurdish script and dialectal variations. Future

work could explore the integration of language-specific features that cater to the cultural and social norms of Kurdish speakers.

2. **Integration of In-App Video Multiplayer Capabilities:** To further enhance the interactive capabilities of the application, the introduction of in-app video multiplayer features is recommended. This would allow users to engage in video interactions within the chat application, supporting real-time visual communication alongside text and voice messaging. Implementing this feature would require careful consideration of bandwidth management, user interface adjustments, and possibly more robust server support to handle increased data loads.

## REFERENCES

- Layton, M. C., Ostermiller, S. J., & Kynaston, D. J. (2020). Agile Project Management For Dummies (3rd ed.). For Dummies.
- Indeed Editorial Team (2023). What Is Requirements Analysis? Process and Techniques. Available at: <https://www.indeed.com/career-advice/career-development/requirements-analysis> (Accessed 1st May 2024)
- Sushant Kumar (2024). Basic of Software Design. Available At: <https://www.scaler.com/topics/software-engineering/what-is-software-design/> (Accessed 1st May 2024)
- Thomas Hamilton (2024). What is Software Testing? Available at: <https://www.guru99.com/software-testing-introduction-importance.html> (Accessed 1st May 2024)