

# **Entrepreneur Growth Guide**

## **Final Year Project Report**

**B.S. in Software Engineering**

**By**

Abdullah Ansari (GL)

2021F-SE-073

Syed Husain Ahmed

2021F-SE-381

Muneeb Ur Rehman

2021F-SE-052

Syed Akber Hasan

2021F-SE-079

### **Supervised By**

Engr. Syed Haris Mehboob

Assistant Professor

SSUET



**2024-2025**

**DEPARTMENT OF SOFTWARE ENGINEERING**

**Sir Syed University of Engineering and Technology**

# Preface

This report entails the climax of our Final Year Project, The Entrepreneur Growth Guide App. Its purpose is to address a major gap in the entrepreneurial ecosystem by offering a solution through the mobile platform that helps to bring structure to the growth of entrepreneurs and small business owners through up-to-date tool sets, strategic frameworks, and practical pieces of information.

This project concept was inspired by the fact that more and more aspiring entrepreneurs, especially in the developing world, lack the right guidance (mentorship), planning resources, and readily available materials on how to start a business. We identified the opportunity of mobile technology to enable such people, and hence we developed the solution which would serve as a digital mentor, easy to use, actionable, and was based on the needs of real-life businesses.

This report provides a discussion of our path, which starts with conceptualization and culminates in deployment, including technical design, literature insight, development, testing, and performance assessment. By doing this, we have not only practiced what we have learned in the software engineering field, but have also ventured into fields under business strategy, user experience, and product design.

This project is a very fruitful endeavor in learning. Our biggest challenges were to comprehend the user requirement, the appropriate tool selection, scope handling within deadlines, and user-centric design. However, conquering these obstacles improved our capabilities in terms of problem solving and provided us with excellent insight into what software development was all about in the real world.

## **Acknowledgments**

First and foremost, we express our sincere gratitude to Allah Almighty, who blessed us with the strength, knowledge, and patience to successfully complete our Final Year Project.

We would like to extend our heartfelt appreciation to our respected supervisor, Sir Haris Mehboob, for his unwavering guidance, continuous support, and invaluable feedback throughout the course of this project. His mentorship played a pivotal role in steering our work toward a successful outcome.

We are also deeply thankful to the academic and technical staff of the Department of Software Engineering at Sir Syed University of Engineering Technology, whose consistent support and cooperation enriched our learning experience throughout our academic journey.

Our sincere thanks also go to the former students, whose documented work and shared experiences provided us with meaningful insights into project expectations and challenges, helping us better navigate our own path.

Lastly, we would like to acknowledge the love, patience, and support of our families and friends. Their encouragement, emotional backing, and understanding played a significant role in helping us stay motivated during challenging times, ultimately enabling us to complete this project with dedication and efficiency.

# Introduction to Group Members



Abdullah Ansari (GL) (2021F-SE-073)  
System Programmer / System Quality Analyzer / Group Leader  
Contact: 0317-3883712, [abdullahansari7@gmail.com](mailto:abdullahansari7@gmail.com)



Syed Husain Ahmed (2021F-SE-381)  
UI/UX Designer (Figma)  
Contact: 0309-9932685, [syedhusainahmed.1@gmail.com](mailto:syedhusainahmed.1@gmail.com)



Muneeb Ur Rehman (2021F-SE-052)  
Developer / 3D Animator  
Contact: 0302-2881409, [muneebr811@gmail.com](mailto:muneebr811@gmail.com)



Syed Akber Hasan (2021F-SE-079)  
Concept Designer / 3D Model Designer  
Contact: 0306-289339, [akberrizvi80@gmail.com](mailto:akberrizvi80@gmail.com)



## SIR SYED UNIVERSITY OF ENGINEERING & TECHNOLOGY

**University Road, Karachi-75300, Pakistan**

Tel. : 4988000-2, 4982393-474583, Fax: (92-21)-4982393

## CERTIFICATE OF COMPLETION

This is to certify that the following students

|                   |             |
|-------------------|-------------|
| Abdullah Ansari   | 2021-SE-073 |
| Syed Husain Ahmed | 2021-SE-381 |
| Muneeb Ur Rehman  | 2021-SE-052 |
| Syed Akber Hasan  | 2021-SE-079 |

Have successfully completed the requirements for Final Year Project titled

## PROJECT TITLE

In the report submission for the Degree of Bachelor of Science in Software Engineering

---

Engr. Syed Haris Mehboob  
Assistant Professor  
SSUET

# Abstract

In today's highly competitive and digitally driven landscape, aspiring entrepreneurs often lack timely access to proper mentorship, personalized guidance, and reliable resources essential to transforming their ideas into viable businesses. This project aims to address these challenges through the design and development of a mobile-based mentorship platform, specifically targeted at students, recent graduates, and early-stage entrepreneurs located in Karachi. The application connects mentees with experienced mentors across various business domains, offering personalized support through text messaging, voice and video calls, and media-sharing capabilities.

The core functionality of the app includes user authentication (sign up and login), mentor discovery and profile viewing, connection request handling, and real-time communication features such as chat, voice calls, and video calls. Mentors can manage incoming requests, accept or reject connections, and communicate directly with mentees using a familiar and responsive interface. The app also integrates a chatbot feature, trained to answer basic business-related queries and capable of handling media files like images and videos, enhancing usability and support.

Emphasis has been placed on creating a clean, user-friendly interface, while maintaining responsiveness across devices. Key technical features include network handling with auto-reconnect on disconnection, secure media transfer, and real-time session management. The backend architecture ensures scalability, low latency communication, and efficient user session tracking.

Furthermore, the application was thoroughly tested through both manual and automated testing, validating functionalities like login, connection management, media transfer, and network resilience. Extensive unit and integration test cases were defined to verify each feature's stability and interaction reliability.

By bridging the gap between aspiring entrepreneurs and seasoned mentors, this project contributes a scalable and adaptable platform that not only supports business mentorship but also fosters entrepreneurship, skill development, and knowledge sharing within underserved communities. The project demonstrates how mobile technology can be leveraged to create meaningful, socially impactful digital solutions in emerging economies.

# Contents

|  |           |
|--|-----------|
| PREFACE . . . . .  | i         |
| ACKNOWLEDGMENTS . . . . .                                    | ii        |
| ABSTRACT . . . . .   | v         |
| LIST OF FIGURES . . . . .                                    | ix        |
| LIST OF TABLES . . . . .                                     | ix        |
| <b>1 Introduction</b>  | <b>1</b>  |
| 1.1 Overview . . . . .                                       | 1         |
| 1.2 Problem Statement . . . . .                              | 1         |
| 1.3 Objectives . . . . .                                     | 2         |
| 1.4 System Features . . . . .                                | 2         |
| 1.5 Project Scope . . . . .                                  | 2         |
| 1.6 Chapter Summary . . . . .                                | 2         |
| <b>2 Literature Review</b>                                   | <b>4</b>  |
| <b>3 Design</b>  | <b>7</b>  |
| 3.1 Design Methodology and Software Process Model . . . . .  | 7         |
| 3.2 Architectural design / Design Patterns . . . . .         | 8         |
| 3.2.1 Architectural Design . . . . .                         | 8         |
| 3.3 Process flow / Representation . . . . .                  | 9         |
| 3.4 Design models [along with description] . . . . .         | 11        |
| 3.4.1 Class Diagram . . . . .                                | 11        |
| 3.4.2 Use Case Diagram . . . . .                             | 12        |
| 3.5 Data Design . . . . .                                    | 12        |
| 3.5.1 Firestore Collections . . . . .                        | 13        |
| 3.6 Data Dictionary . . . . .                                | 15        |
| 3.6.1 Collection: users . . . . .                            | 15        |
| 3.6.2 Collection: appointments . . . . .                     | 15        |
| 3.6.3 Collection: calls . . . . .                            | 16        |
| 3.6.4 ZIMMessage (Message Model) . . . . .                   | 16        |
| 3.6.5 Media Message Fields (Images, Videos, Audio) . . . . . | 17        |
| 3.7 Chapter Summary . . . . .                                | 17        |
| <b>4 System Development</b>                                  | <b>18</b> |
| 4.1 System Architecture . . . . .                            | 18        |
| 4.2 Development Tools and Technologies . . . . .             | 18        |
| 4.3 Authentication and User Management . . . . .             | 18        |
| 4.4 Chat and Call Integration . . . . .                      | 19        |
| 4.5 Database Design and Structure . . . . .                  | 19        |

|  |           |
|--|-----------|
| 4.6 Chatbot Feature . . . . .                              | 19        |
| 4.7 Media and File Upload Handling . . . . .               | 19        |
| 4.8 Functional Testing and Verification Strategy . . . . . | 19        |
| 4.9 Final Build Deployment and Maintenance Plan . . . . .  | 20        |
| <b>5 Testing</b>   | <b>21</b> |
| 5.1 Manual Testing . . . . .                               | 21        |
| 5.1.1 Unit Testing . . . . .                               | 23        |
| 5.1.2 Integration Testing . . . . .                        | 25        |
| 5.2 Automation Testing . . . . .                           | 27        |
| <b>6 Performance Evaluation</b>                            | <b>28</b> |
| 6.1 Evaluation Objectives . . . . .                        | 28        |
| 6.2 Tools and Environment . . . . .                        | 28        |
| 6.3 Performance Metrics and Benchmarks . . . . .           | 28        |
| 6.4 Key Results . . . . .                                  | 29        |
| 6.4.1 App Startup and Navigation . . . . .                 | 29        |
| 6.4.2 Chat and Messaging . . . . .                         | 29        |
| 6.4.3 Audio and Video Calling . . . . .                    | 29        |
| 6.4.4 UI Performance . . . . .                             | 29        |
| 6.5 Error Handling and Resilience . . . . .                | 30        |
| 6.6 Limitations Identified . . . . .                       | 30        |
| 6.7 Optimization Actions Taken . . . . .                   | 30        |
| <b>7 Business Model</b>                                    | <b>31</b> |
| 7.1 Market Research . . . . .                              | 31        |
| 7.2 Unique Value Proposition (UVP) . . . . .               | 31        |
| 7.3 Targeted Audience . . . . .                            | 31        |
| 7.4 Monetization Strategy . . . . .                        | 32        |
| 7.5 Scalability and Internationalization . . . . .         | 32        |
| 7.6 Propose Business Plan . . . . .                        | 32        |
| 7.7 Integration with Final Year Project . . . . .          | 32        |
| 7.7.1 Project Alignment . . . . .                          | 33        |
| 7.7.2 Implementation Details . . . . .                     | 33        |
| 7.7.3 Market Experience: . . . . .                         | 33        |
| 7.8 Future Road map . . . . .                              | 33        |
| 7.8.1 Future Development . . . . .                         | 33        |
| 7.8.2 Adaptability . . . . .                               | 33        |
| 7.9 International Expansion . . . . .                      | 34        |
| 7.10 Legal Considerations . . . . .                        | 34        |
| 7.11 Automation Testing . . . . .                          | 34        |
| <b>8 Future Directions and Conclusion</b>                  | <b>35</b> |
| 8.1 Future Directions: . . . . .                           | 35        |
| 8.2 Conclusion: . . . . .                                  | 36        |
| 8.3 Getting Started . . . . .                              | 37        |
| 8.3.1 Installation: . . . . .                              | 37        |
| 8.3.2 Login / Signup: . . . . .                            | 37        |
| 8.4 Navigating the App . . . . .                           | 37        |

|                    |                                    |           |
|--------------------|------------------------------------|-----------|
| 8.4.1              | User Authentication . . . . .      | 37        |
| 8.4.2              | Choosing a Mentor . . . . .        | 37        |
| 8.4.3              | Send Connection Request . . . . .  | 37        |
| 8.4.4              | Mentor Accepting Request . . . . . | 38        |
| 8.5                | Communication Features . . . . .   | 38        |
| 8.5.1              | Text Messaging . . . . .           | 38        |
| 8.5.2              | Send Media Files . . . . .         | 38        |
| 8.5.3              | Voice Messages . . . . .           | 38        |
| 8.5.4              | Voice Call . . . . .               | 38        |
| 8.5.5              | Video Call . . . . .               | 38        |
| 8.5.6              | Call Controls . . . . .            | 38        |
| 8.6                | Chatbot Feature . . . . .          | 39        |
| 8.6.1              | Smart Chatbot Assistant . . . . .  | 39        |
| <b>Appendicies</b> |                                    | <b>37</b> |
| Appendix A         | . . . . .                          | 37        |
| Appendix B         | . . . . .                          | 40        |
| Appendix C         | . . . . .                          | 41        |
| Appendix D         | . . . . .                          | 42        |
| 8.7                | Poster . . . . .                   | 45        |
| 8.8                | Standee . . . . .                  | 46        |
| 8.9                | Brochure . . . . .                 | 47        |
| Appendix E         | . . . . .                          | 45        |
| <b>References</b>  |                                    | <b>49</b> |

# List of Figures

|      |                                 |    |
|------|---------------------------------|----|
| 3.1  | Enter Caption . . . . .         | 10 |
| 3.2  | Enter Caption . . . . .         | 11 |
| 3.3  | Enter Caption . . . . .         | 12 |
| 3.4  | Users . . . . .                 | 15 |
| 3.5  | Appointments . . . . .          | 15 |
| 3.6  | Calls . . . . .                 | 16 |
| 3.7  | Message . . . . .               | 16 |
| 3.8  | Media . . . . .                 | 17 |
| 5.1  | Manual Testing1 . . . . .       | 21 |
| 5.2  | Manual Testing 2 . . . . .      | 22 |
| 5.3  | Manual Testing 3 . . . . .      | 22 |
| 5.4  | Manual Testing 4 . . . . .      | 23 |
| 5.5  | Unit testing 1 . . . . .        | 23 |
| 5.6  | Unit testing 2 . . . . .        | 24 |
| 5.7  | Integration Testing 1 . . . . . | 25 |
| 5.8  | Integration Testing 2 . . . . . | 25 |
| 5.9  | Integration Testing 3 . . . . . | 26 |
| 5.10 | Automation Testing . . . . .    | 27 |
| 8.1  | Research paper 1 . . . . .      | 43 |
| 8.2  | research paper 2 . . . . .      | 44 |
| 8.3  | Poster . . . . .                | 45 |
| 8.4  | Standee . . . . .               | 46 |
| 8.5  | Brochure Front . . . . .        | 47 |
| 8.6  | Brochure Back . . . . .         | 48 |

# Chapter 1

## Introduction

### 1.1 Overview

Karachi has always been a city of ambitions, creativity, and untapped potential and this pandemic will only add to their undoing. Home to thousands of schools, colleges, and universities, it is also a place of the rapidly increasing number of tech-savvy young people, and hence a cauldron of entrepreneurial spirit. However, a substantial part of students and even those who want to start their business in the city do not have stern knowledge, guidance, and exposure towards starting or expanding a business. This project proposes the business oriented mentorship mobile application that will cater to the needs of the students and the inexperienced people in Karachi who are interested in business and entrepreneurship, however lack the guidance, experience or resources in this field. The platform fills this gap because it helps entry-level business builders get in touch with experienced business coaches, startup leaders, business instructors, and other business experts in real-time. The application also includes other features that are advanced in terms of communication like the one-on-one chat, audio and video calls, and an AI-based chatbot which knows the variants of basic business advice, startup support, and motivational advice. It is a virtual incubator of businesses available to those who have a smartphone and the inspiration to learn.

### 1.2 Problem Statement

Although Karachi has developed an entrepreneurial ecosystem with increasing startups, business incubators and business schools, majority of the students and early-stage startup founders are left behind within the ecosystem because they lack access to the right mentorship. High levels of business education cost a lot of money, they are sometimes not applicable and often not available to other students in low resource protected regions. A lot of people also possess incredible ideas but lack knowledge in market analysis, pitching, investment, scalability, and customer acquisition amongst others. Also, students do not always succeed in identifying a reputable professional who may want to mentor them, and preferably in an individual and customized manner. Such networks are too general and unfiltered such as LinkedIn or social media to be good mentorship tools due to their lack of relevance in the process of mentorship except to the individual with the most network connection or business experience. Consequently, most of the good ideas remain untapped and some of the potential entrepreneurs quit before they start because of being confused, afraid or losing direction.

## 1.3 Objectives

The main goals of this project are the following ones:

- To foster entrepreneurship thinking and enable the users to have a practical insight into business-related matters through the expert instructions.
- To offer business mentorship that is to be accessible to Karachi students in underserved regions of the city, an initiative that will nurture a lack of exposure to formal business networks.
- To support real-time chat, one on one audio/video call and meeting scheduling.
- Integration of an AI powered chatbot that provides some rudimentary instructions on starting, answering commonly asked questions and assisting mentees in shaping their idea.
- To inspire a set of mentees and teach them to build the courage and vision to mentee using sustainable, one on one mentorship.

## 1.4 System Features

- **User Signup and Role Selection:**

One will be allowed to register as a mentor or as a mentee using basic credentials.

- **Profile Management:**

Individuals profile themselves by outlining their backgrounds and interests at the personal and professional levels.

- **Mentor Categorization:**

Mentees are able to search mentors according to their respective fields.

- **One-on-One Chat:**

Most forms of private messaging take the form of text-based exchanges between mentors and their mentees.

- **Audio Calling:**

Mentorship communication can be done directly through voice in order to have a more effective conversation.

- **Video Calling:**

Real life mentorship via video conferencing in the app.

- **AI-Powered Chatbot:**

A virtual assistant that delivers business advice and support, advice about and help with starting up a business.

## 1.5 Project Scope

This project will be focused on the students, recent graduates and early entrepreneurs located in Karachi looking to get business mentorship. The application is mobile-based, which means residents of the city, both in terms of physical location and the level of income, can benefit by receiving individual advice available. Although the app is business and entrepreneurship-oriented, new capabilities will involve professional development in work as a freelancer, digital, and startup financing.

## 1.6 Chapter Summary

This chapter proposed a business mentoring application that can help the youth of Karachi with some good hands-on guidance in business and expert advice to those beginner and amateur

individuals who are keen to start a business. The app offers the essential features of messaging, video call, and AI support to bridge the most necessary gap in the startup and academic environment of Karachi. In the following chapter, we shall go through extant platforms and studies on mentorship models as applied to the concept of entrepreneurship and young people development.

# Chapter 2

## Literature Review

### **Introduction:**

The present chapter discusses the main works, platforms and frameworks that relate to the topics of business mentorship, entrepreneurship learning and the mobile technology. It points out the lack of resources in the current systems and reasons out why localized and digital mentorship platform is required among students and potential business owners in Karachi.

### **Sensitivity of Mentorship:**

Mentorship has been shown to help in building confidence, skills and the long term success in terms of a career. Mentors can offer a lot of expertise to entrepreneurs on things such as testing out an idea, marketing, raising funds, and business planning.

- Research (Crisp Cruz, 2009) indicates that a mentored person performs better in his or her professional life.
- The ILO finds that business survival gets enhanced greatly when it is trained by mentorships in the developing nations.
- Young people who cannot find a family or institutional support of their business activities can also be the most influenced by mentorship in such cities as Karachi.

### **Pakistani Entrepreneurship Education:**

Although major universities in Pakistan have business courses, most students especially those in government colleges have not been exposed to entrepreneurship training that can be applied in their business enterprise.

- According to UNDP (2018), Pakistani young people have a great entrepreneurial capability but lack support and access to resources due to lack of direction.
- Most curricula are theory-oriented and not about real world experience and mentorship.
- Business incubators (e.g., The Nest I/O, NIC Karachi) are open and accessible to startups but are not open and accessible to the general group of students.
- There is a wish to have a more inclusive and easier-to-access mentorship model.

### Significance of Mobile Technology:

Mobile learning and mentoring is an effective means to provide learning and mentoring in an environment like Karachi, where smartphone penetration is dominating.

- Pakistan has more than 80 million internet users with majority of them accessing through the mobile.
- International websites such as MentorCloud and PushFar are out there yet costly and not built to serve the local emphasis.
- Other mentorship applications only support corporate mentoring while leaving behind the early stage or the student entrepreneur.
- Volunteers can become accessible to underserved youth by providing mentorship on a localized, mobile-first platform that uses Urdu and English.

### Gaps in Current Platforms

- LinkedIn: Has networking aspects very effective, not as to mentorship.
- Coursera/EdX: Runs an excellent course but the problem is that you do not have access to a personal mentor.
- Local incubators: Have small capacities and are only available to advanced startups.

### Karachi Local Difficulties:

The early business people and students in Karachi usually:

- They do not have access to experienced mentors.
- Know little of the real world.
- Cultural/family pressure not to start a business
- Want simple and low cost to utilize tools of common languages.

An app to mentor the residents of Karachi so that they can eliminate these obstacles since it will be targeted at the setting where the needs exist.

### AI Mentorship:

- Education and business platforms are also widely using AI chatbots as tools that can help round the clock.
- They have the potential to provide some business low-level advice, definitions, and instructions and continue to do so even when mentors are offline.

- They alleviate the burden to human mentors and act as a frontline of support to the users.

Since the AI is added to the app, it makes the app more useful and available to new users.

## **Summary**

Although there are mentorship and business learning platforms in the world some of them are inaccessible, unaffordable, and irrelevant to the youth of Karachi. Startup environments in the northwestern regions of the U.S. lack a mobile first, business centered mentorship platform that supports interactivity through real time communication, leverages AI to increase the scale of support and connections, and deploys an element of localization cultural familiarity and accessibility to the student economy; that is, students and aspiring entrepreneurs regardless of their socioeconomic background.

# Chapter 3

## Design

### 3.1 Design Methodology and Software Process Model

The Entrepreneur Growth Guide app employs a User-Centered Design (UCD) methodology, focusing on the distinct needs of two user roles Mentors and Mentees.

Key aspects include:

#### User Role Differentiation:

The design caters to Mentors (who manage connections, initiate calls, and schedule appointments) and Mentees (who browse profiles, send requests, and receive calls). This is implemented through role-based UI customization on the home and profile screens.

#### Iterative Prototyping:

The app uses iterative design cycles, starting with on boarding and signup screens, followed by role-specific home screens with four sections (Profiles, Chats, Calls, Chatbot).

#### Data-Driven Design:

Firestore's schema drives the app's data interactions, such as storing user profiles, connections, and appointments, influencing UI components like chat bubbles and call history.

#### Modular Integration:

External services (Gemini API for chatbot, Zegocloud for real-time communication, Cloudinary for media storage) are integrated modularly, allowing independent updates and scalability.

#### Feedback-Driven Refinement:

Features like email verification and notification scheduling (via FCM) incorporate user feedback to enhance usability and engagement.

This methodology ensures a responsive, role-specific, and scalable application tailored to entrepreneurial mentorship needs.

## 3.2 Architectural design / Design Patterns

### 3.2.1 Architectural Design

The app adopts a Client-Server Architecture with a Microservices-like structure, integrating multiple services

#### Client Layer

- Built with Flutter, handling UI and user interactions across onboarding, signup/signin, and home screen sections (Profiles, Chats, Calls, Chatbot).
- Role-based UI rendering for Mentors (connection management, call initiation) and Mentees (profile browsing, request sending).

#### Server Layer

- Firebase Authentication: Manages user authentication and email verification.
- Firestore: Stores user profiles, connections, appointments, and FCM tokens in a NoSQL database.
- Gemini API: Powers the chatbot for mentorship guidance.
- Zegocloud: Enables real-time text, voice, video, and file sharing.
- Cloudinary: Stores media (images, videos, audio) for chat attachments.

#### Layered Architecture in Flutter

- Presentation Layer: Widgets like Chats Screen, One On One Chat Screen, and One On One ChatBubble.
- Business Logic Layer: Stateful widget logic (e.g., send Message, schedule Appointment) and role-based restrictions.
- Data Layer: Interfaces with Firestore, Zegocloud, and Cloudinary via APIs.

#### Design Patterns

##### Singleton Pattern:

- Ensures single instances of FirebaseAuth, FirebaseFirestore, and ZIM (Zegocloud) for consistent access.

##### MVC (Model-View-Controller):

- Model: Firestore collections (users, appointments, Mentors, Mentees).–
- View: Flutter widgets rendering data (e.g., ChatTile, OneOnOneChatBubble).
- Controller: State management in One On One Chat Screen State for message sending, appointment scheduling.

**Observer Pattern:**

- Handles real-time updates via ZIMEEventHandler.onReceivePeerMessage for messages and FirebaseMessaging.onMessage for notifications.

**Factory Pattern:**

- Dynamically renders chat bubbles based on message type (text, image, video, audio, appointment) in One On One Chat Bubble.

**Repository Pattern:**

- Firestore serves as a centralized data repository, abstracting data access for profiles, connections, and appointments

### 3.3 Process flow / Representation

The process flow outlines the sequence of user interactions and system operations, represented as a sequence diagram.

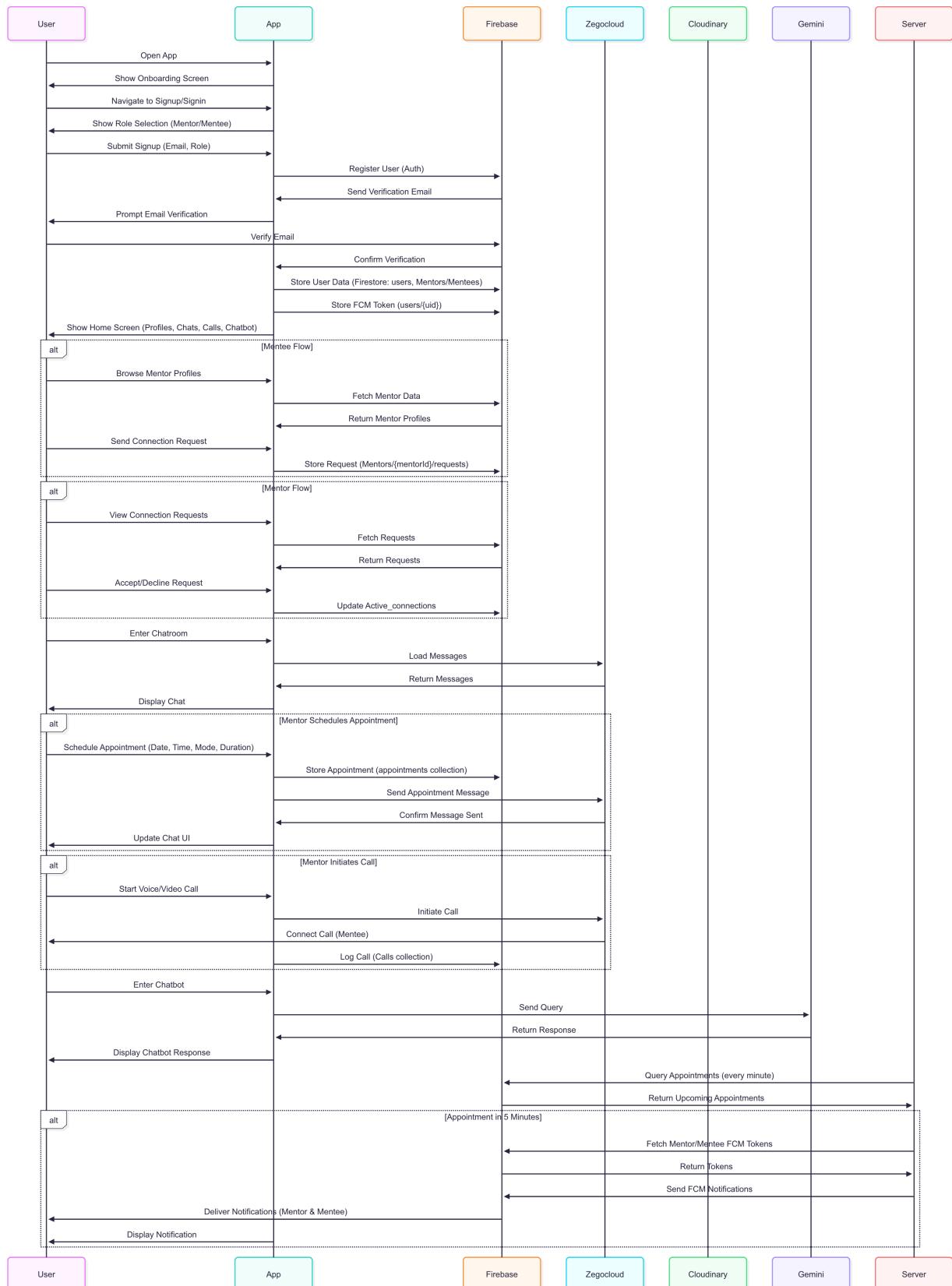


Figure 3.1: Enter Caption

## 3.4 Design models [along with description]

Design models represent the system's structure and behavior using UML diagrams.

### 3.4.1 Class Diagram

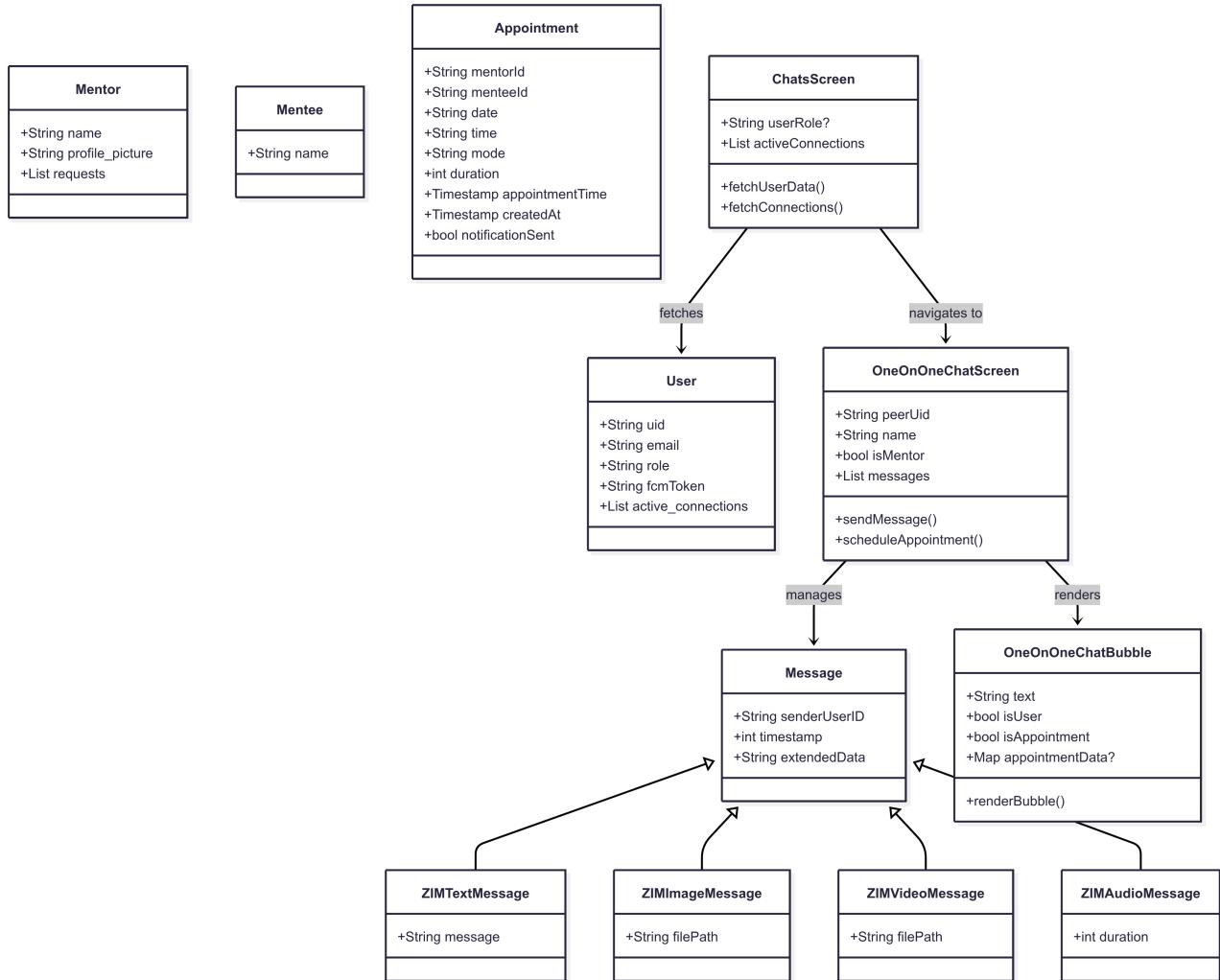


Figure 3.2: Enter Caption

### 3.4.2 Use Case Diagram

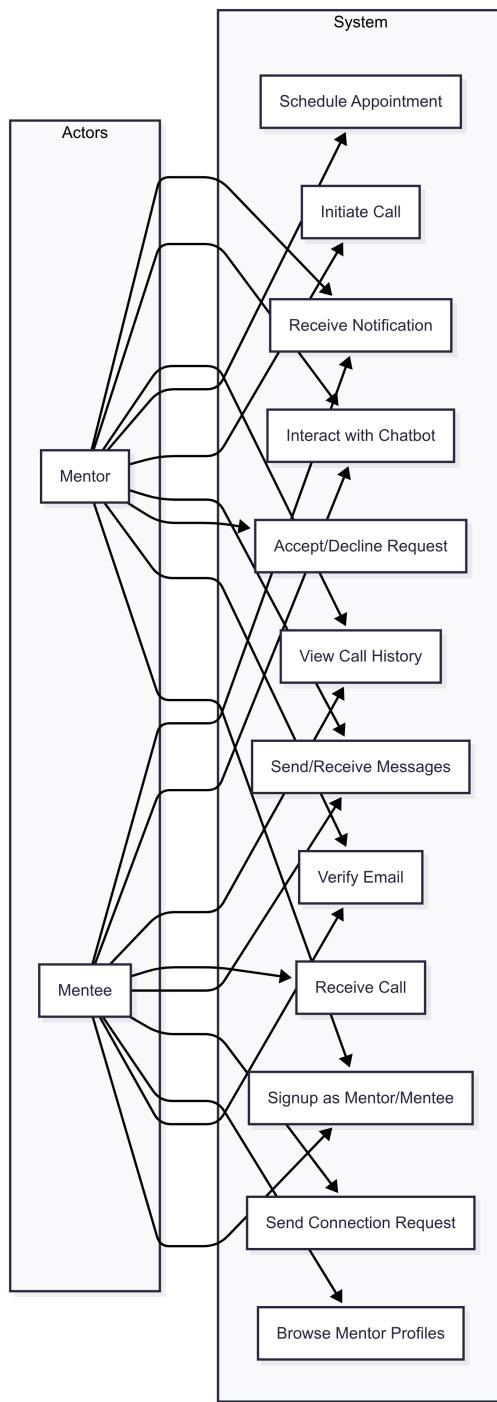


Figure 3.3: Enter Caption

## 3.5 Data Design

The app uses Firestore (NoSQL) for primary data storage, Cloudinary for media, and Zegocloud for real-time message data. The schema is designed for role-based access and real-time updates.

### 3.5.1 Firestore Collections

#### Users:

- Document ID: User UID

#### Fields:

- Email: String (user's email)
- Role: String ("Mentor" or "Mentee")
- fcmToken: String (FCM registration token)
- Active connections: Array (list of connected user UIDs)
- Purpose: Stores user authentication and notification data.

#### Mentors:

- Document ID: Mentor UID

#### Fields:

- Name: String
- Profile picture: String (Cloudinary URL)
- Requests: Array (list of mentee UIDs requesting connection)
- Purpose: Stores mentor specific profile data and connection requests.

#### Mentees

- Document ID: Mentee UID– Fields:
  - Name: String
  - Purpose: Stores mentee-specific profile data.

#### Appointments

- Document ID: Auto-generated

#### Fields:

- MentorId: String (Mentor UID)
- MenteeId: String (Mentee UID)
- Date: String (e.g., "2025-08-10")
- Time: String (e.g., "14:00")
- Mode: String ("video" or "voice")

- Duration: Integer (minutes)
- AppointmentTime: Timestamp (full date-time)
- CreatedAt: Timestamp
- NotificationSent: Boolean
- Purpose: Stores appointment details for scheduling and notifications.

## Calls

- Document ID: Auto-generated

### Fields:

- Call id: String
- Caller id: String (Mentor UID)
- Caller id: String (Mentee UID)
- Call type: String (“voice” or “video”)
- Status: String (“initiated”, “missed”, etc.)
- Timestamp: Timestamp
- Accept time: Timestamp (nullable)
- End time: Timestamp (nullable)
- Duration: Integer (seconds)
- Purpose: Logs call history.

## 3.6 Data Dictionary

### 3.6.1 Collection: users

| Field              | Type   | Description   |
|--------------------|--------|---|
| uid                | String | Unique identifier for the user (Firebase Auth UID).                     |
| email              | String | User's email address used for authentication and verification.          |
| role               | String | Defines the user role: "Mentor" or "Mentee".                            |
| fcmToken           | String | Firebase Cloud Messaging token for sending push notifications.          |
| active_connections | Array  | List of user UIDs (Mentors/Mentees) the user is currently connected to. |
| name               | String | User's full display name.   |
| profile_picture    | String | URL of the mentor's profile picture (stored in Cloudinary).             |
| requests           | Array  | List of mentee UIDs who have sent connection requests to a mentor.      |

Figure 3.4: Users

### 3.6.2 Collection: appointments

| Field            | Type      | Description   |
|------------------|-----------|---|
| mentorId         | String    | UID of the mentor involved in the appointment.                        |
| menteeId         | String    | UID of the mentee who booked the appointment.                         |
| date             | String    | Date of the appointment in "YYYY-MM-DD" format.                       |
| time             | String    | Time of the appointment in "HH:mm" format.                            |
| mode             | String    | Type of session: "video" or "voice".                                  |
| duration         | Integer   | Duration of the session in minutes (e.g., 15, 30, 60).                |
| appointmentTime  | Timestamp | Full timestamp of the appointment (used for sorting or filtering).    |
| createdAt        | Timestamp | Timestamp of when the appointment was created.                        |
| notificationSent | Boolean   | Indicates whether a 5-minute prior notification was sent to the user. |

Figure 3.5: Appointments

### 3.6.3 Collection: calls

| Field       | Type      | Description                                     |
|-------------|-----------|---|
| call_id     | String    | Unique identifier for each call.                |
| caller_id   | String    | UID of the user who initiated the call.         |
| callee_id   | String    | UID of the recipient user.                      |
| call_type   | String    | Call mode: "voice" or "video".                  |
| status      | String    | Call status such as "initiated", "missed", etc. |
| timestamp   | Timestamp | Time when the call was initiated.               |
| accept_time | Timestamp | Time when the call was accepted (nullable).     |
| end_time    | Timestamp | Time when the call ended (nullable).            |
| duration    | Integer   | Total call duration in seconds.                 |

Figure 3.6: Calls

### 3.6.4 ZIMMessage (Message Model)

| Field        | Type                       | Description  |
|--------------|----------------------------|--|
| senderUserID | String                     | UID of the sender of the message.                                |
| timestamp    | Integer (ms)               | Time when the message was created (milliseconds since epoch).    |
| extendedData | JSON String                | Metadata in JSON format (e.g., appointment details, media URLs). |
| message      | String<br>(ZIMTextMessage) | The actual text content of the message.                          |

Figure 3.7: Message

### 3.6.5 Media Message Fields (Images, Videos, Audio)

| Field    | Type                 | Description   |
|----------|----------------------|---|
| filePath | String               | Local path or Cloudinary URL of the media file.                           |
| duration | Integer (ms)         | Duration of audio/video messages (in milliseconds).                       |
| type     | ZIMImage/Video/Audio | Message type determines how media is rendered and interpreted in the app. |

Figure 3.8: Media

## 3.7 Chapter Summary

This design document provides a comprehensive overview of the Entrepreneur Growth Guide app, detailing its user-centered design methodology, Agile (Scrum) process model, client-server architecture with design patterns, process flows, UML design models, Firestore-based data design, and a detailed data dictionary. The app leverages Firebase, Zegocloud, Cloudinary, and Gemini API to deliver a robust mentorship platform with role based functionality, real-time communication, and push notifications.

# Chapter 4

# System Development

This chapter outlines the overall development of the Entrepreneur Growth Guide application, including the architecture, technologies used, database structure, core functionalities, and integration of third-party services. The purpose of this chapter is to give a comprehensive understanding of how each component of the system was designed, implemented, and optimized during the development process.

## 4.1 System Architecture

The system is based on a client-server architecture. The mobile application, developed using Flutter, interacts with cloud-based services like Firebase and ZEGOCLOUD to provide real-time communication, authentication, database access, and media management. The front-end runs on user devices, while all back-end operations are managed on cloud servers. This architecture supports scalability, modularity, and ease of deployment.

## 4.2 Development Tools and Technologies

Flutter was chosen as the front-end development framework due to its cross-platform capabilities, expressive UI components, and strong community support. Firebase was selected for backend services, including authentication, real-time databases, cloud messaging, and performance monitoring. For video and voice communication, ZEGOCLOUD SDK was integrated. Media files are uploaded and managed via Cloudinary, which provides efficient storage and fast delivery.

## 4.3 Authentication and User Management

User authentication is handled via Firebase Authentication, supporting email and password login. Upon successful registration, users are assigned roles (either mentor or mentee), which determine their permissions and accessible features. The user data is stored in Firestore, including profile information, active connections, and FCM tokens for notifications.

Security rules are implemented to ensure that data access is limited based on user roles and session validation. For example, mentees cannot access mentor chat histories unless a valid connection is established.

## 4.4 Chat and Call Integration

The communication system was a core feature of the application. Real-time chat was implemented using ZEGOCLOUD's messaging SDK, which supports text, images, voice notes, and document sharing. This makes it easier for mentees to share media files during discussions with mentors.

Voice and video call features were integrated through ZEGOCLOUD's call SDK. Users can initiate or receive calls based on appointment status. All calls are tracked for duration and status (accepted, rejected, missed). Proper permissions and lifecycle handling were managed using Flutter plugins for camera, microphone, and notifications.

## 4.5 Database Design and Structure

The backend is powered by Firebase Firestore, a NoSQL real-time database. The structure is document-based, where each collection stores user-specific data, appointments, messages, and call logs.

User profiles are stored in a "users" collection, and connections between mentors and mentees are handled through user references. Appointments have their own records with fields for date, time, type of call, and participants. Messages and call records are stored separately to improve scalability and performance.

## 4.6 Chatbot Feature

An additional chatbot was integrated into the application to answer business-related queries. The chatbot is not connected to application guidance or support features but is capable of providing responses to general business questions like "How to register a startup in Pakistan?" or "What is an entrepreneur?"

The chatbot also supports receiving and sending media content, such as images or documents shared by the user. This makes the conversation more dynamic and helpful, especially in explaining business models or frameworks.

## 4.7 Media and File Upload Handling

For managing media files like images, voice notes, and documents, Cloudinary was integrated. It provides efficient media upload and access. Users can share these files in chat, and mentors can review shared resources asynchronously.

All uploaded files are securely stored and accessed via public URLs. Media upload progress and error handling are managed on the front-end to provide feedback to the user during transmission.

## 4.8 Functional Testing and Verification Strategy

A comprehensive testing strategy was followed to ensure system correctness and reliability. Unit testing was conducted for core functions such as authentication, chat messaging, and appointment scheduling. Integration testing verified the proper interaction between modules like the user database, chat interface, and video call SDK. Real-device testing was also carried

out on various Android smartphones to identify UI compatibility and performance variations. Bugs discovered during testing were logged, categorized, and fixed according to priority before the final deployment.

## 4.9 Final Build Deployment and Maintenance Plan

Once development was complete and testing was successful, the application was built and deployed as a signed APK for Android distribution. The APK was shared with the target users. Post-deployment, a maintenance plan was established which includes regular updates for SDK dependencies, bug fixes, UI enhancements, and adding new features based on user feedback. Future plans include hosting on the Google Play Store and enabling iOS deployment.

# Chapter 5

## Testing

### 5.1 Manual Testing

| Manual Test cases for Entrepreneur Growth Guide App |             |          |   |   |  |        |
|---|-------------|----------|---|---|--|--------|
| Test Case ID  | Feature     | Type     | Test Steps  | Expected Result   | Actual Behavior                                  | Status |
| TC01-P  | App Launch  | Positive | Open the app from home screen or app drawer.          | Splash screen appears and redirects to login/signup page. | App launches correctly and transitions smoothly. | Pass   |
| TC01-N1   | App Launch  | Negative | Launch the app with airplane mode or no internet.     | App should show network error or handle gracefully.       | Error shown; app handled offline state.          | Pass   |
| TC01-N2   | App Launch  | Negative | Launch on unsupported OS version or corrupted APK.    | App should restrict launch and show compatibility error.  | App crashed or showed OS incompatibility.        | Fail   |
| TC02-P  | User Login  | Positive | Enter valid email/password and press 'Login'.         | User is authenticated and navigated to home screen.       | Login successful.                                | Pass   |
| TC02-N1   | User Login  | Negative | Enter wrong password and try to login.                | System should deny access and show error message.         | Error message shown: 'Invalid credentials'.      | Pass   |
| TC02-N2   | User Login  | Negative | Leave fields empty and press 'Login'.                 | App should prompt to fill all fields.                     | Validation message appeared.                     | Pass   |
| TC03-P  | User Signup | Positive | Enter valid name, email, password and press 'Signup'. | User account is created and redirected to home.           | Signup successful and user redirected.           | Pass   |
| TC03-N1   | User Signup | Negative | Enter already registered email.                       | App should notify about duplicate email.                  | Duplicate email message shown.                   | Pass   |
| TC03-N2   | User Signup | Negative | Enter invalid email format.                           | System should reject and show email format error.         | Validation error shown.                          | Pass   |

Figure 5.1: Manual Testing1

|         |                   |          |   |   |                                  |      |
|---------|-------------------|----------|---|---|----------------------------------|------|
| TC04-P  | Choose Mentor     | Positive | Scroll and select a mentor profile.       | Mentor details shown; user can send connection request. | Mentor profile loaded.           | Pass |
| TC04-N1 | Choose Mentor     | Negative | Tap a mentor card without internet.       | System should show loading error.                       | Network error message displayed. | Pass |
| TC04-N2 | Choose Mentor     | Negative | Select a non-existent mentor (edge case). | App should not crash and handle gracefully.             | Handled with fallback response.  | Pass |
| TC05-P  | View Mentor       | Positive | Tap on mentor profile after search.       | Mentor's full profile and availability displayed.       | Detailed view shown correctly.   | Pass |
| TC05-N1 | View Mentor       | Negative | Try viewing mentor without logging in.    | System should redirect to login.                        | Redirected to login page.        | Pass |
| TC05-N2 | View Mentor       | Negative | Click profile during network switch.      | App should handle delay and retry.                      | Slight delay but handled.        | Pass |
| TC06-P  | Send Connection   | Positive | Click 'Connect' on mentor profile.        | Request sent and status updated.                        | Request sent successfully.       | Pass |
| TC06-N1 | Send Connection   | Negative | Send request to already connected mentor. | System should notify duplicate.                         | Already connected message shown. | Pass |
| TC06-N2 | Send Connection   | Negative | Rapidly tap 'Connect' repeatedly.         | App should debounce the requests.                       | Only one request sent.           | Pass |
| TC07-P  | Accept Connection | Positive | Tap 'Accept' on mentee request.           | Request accepted and connection established.            | Connection successful.           | Pass |
| TC07-N1 | Accept Connection | Negative | Try to accept after user cancels.         | App should notify request no longer valid.              | Error: Request expired.          | Pass |
| TC07-N2 | Accept Connection | Negative | Tap accept with poor network.             | System should retry or show failure message.            | Handled with retry.              | Pass |
| TC08-P  | Text Chat         | Positive | Open chat, type message and send.         | Message should appear in chat instantly.                | Message sent successfully.       | Pass |
| TC08-N1 | Text Chat         | Negative | Send empty message.                       | Send button should remain disabled.                     | No action triggered.             | Pass |

Figure 5.2: Manual Testing 2

|         |                   |          |  |  |                                   |      |
|---------|-------------------|----------|--|--|-----------------------------------|------|
| TC08-N2 | Text Chat         | Negative | Send message with network off.         | Message should be queued or fail gracefully.     | Pending or failed to send.        | Pass |
| TC09-P  | Media Sharing     | Positive | Send an image file in chat.            | Image appears in chat with preview.              | Image sent and rendered.          | Pass |
| TC09-N1 | Media Sharing     | Negative | Send unsupported file type.            | App should reject with message.                  | Unsupported format message shown. | Pass |
| TC09-N2 | Media Sharing     | Negative | Send large video file.                 | App should compress or show limit error.         | File size error shown.            | Pass |
| TC10-P  | Voice Call        | Positive | Initiate voice call and speak.         | Call connects with clear audio.                  | Call connected and clear.         | Pass |
| TC10-N1 | Voice Call        | Negative | Call when recipient is offline.        | Call should not connect and notify user.         | Offline status shown.             | Pass |
| TC10-N2 | Voice Call        | Negative | Mute/unmute repeatedly.                | App should toggle states correctly.              | Mute toggle worked fine.          | Pass |
| TC11-P  | Video Call        | Positive | Initiate video call and enable camera. | Call connects and video streams properly.        | Video call connected smoothly.    | Pass |
| TC11-N1 | Video Call        | Negative | Cover camera or poor lighting.         | Video should still stream or show dim.           | Low quality but maintained.       | Pass |
| TC11-N2 | Video Call        | Negative | Switch apps mid-call.                  | App should run in background or resume properly. | Resumed successfully.             | Pass |
| TC12-P  | Call Controls     | Positive | Use mute, camera toggle, and end call. | All controls work as intended.                   | Controls responded correctly.     | Pass |
| TC12-N1 | Call Controls     | Negative | Press multiple controls rapidly.       | App should debounce and respond orderly.         | Slight lag but handled.           | Fail |
| TC12-N2 | Call Controls     | Negative | Try controls with screen off.          | Should be blocked or disabled.                   | Not accessible when screen off.   | Pass |
| TC13-P  | Network Stability | Positive | Use app with stable Wi-Fi.             | All services work without issue.                 | Smooth operation.                 | Pass |
| TC13-N1 | Network Stability | Negative | Switch from Wi-Fi to 4G mid-use.       | App should reconnect gracefully.                 | Minor delay, auto-reconnected.    | Pass |

Figure 5.3: Manual Testing 3

|         |                   |          |                                |  |                        |      |
|---------|-------------------|----------|--------------------------------|--|------------------------|------|
| TC13-N2 | Network Stability | Negative | Disable data completely.       | System should show offline state.        | Network error shown.   | Pass |
| TC14-P  | UI Responsiveness | Positive | Navigate menus, open features. | All transitions are smooth.              | Minor lag, acceptable. | Fail |
| TC14-N1 | UI Responsiveness | Negative | Spam button presses.           | App should throttle and stay responsive. | Lagged slightly.       | Fail |
| TC14-N2 | UI Responsiveness | Negative | Scroll heavy lists rapidly.    | App should handle smoothly.              | Slow response.         | Fail |

Figure 5.4: Manual Testing 4

### 5.1.1 Unit Testing

| Test Case ID | Module           | Test Scenario                     | Input                 | Expected Output   | Status   |
|--------------|------------------|-----------------------------------|-----------------------|-------------------|----------|
| UT01         | Login            | Login - Valid scenario            | Valid input           | Expected output   | Positive |
| UT02         | Login            | Login - Invalid input 1           | Invalid input         | Error message     | Negative |
| UT03         | Login            | Login - Invalid input 2           | Empty/Incorrect input | Validation failed | Negative |
| UT04         | Signup           | Signup - Valid scenario           | Valid input           | Expected output   | Positive |
| UT05         | Signup           | Signup - Invalid input 1          | Invalid input         | Error message     | Negative |
| UT06         | Signup           | Signup - Invalid input 2          | Empty/Incorrect input | Validation failed | Negative |
| UT07         | View Mentor      | View Mentor - Valid scenario      | Valid input           | Expected output   | Positive |
| UT08         | View Mentor      | View Mentor - Invalid input 1     | Invalid input         | Error message     | Negative |
| UT09         | View Mentor      | View Mentor - Invalid input 2     | Empty/Incorrect input | Validation failed | Negative |
| UT10         | Send Connection  | Send Connection - Valid scenario  | Valid input           | Expected output   | Positive |
| UT11         | Send Connection  | Send Connection - Invalid input 1 | Invalid input         | Error message     | Negative |
| UT12         | Send Connection  | Send Connection - Invalid input 2 | Empty/Incorrect input | Validation failed | Negative |
| UT13         | Chat             | Chat - Valid scenario             | Valid input           | Expected output   | Positive |
| UT14         | Chat             | Chat - Invalid input 1            | Invalid input         | Error message     | Negative |
| UT15         | Chat             | Chat - Invalid input 2            | Empty/Incorrect input | Validation failed | Negative |
| UT16         | Send Media       | Send Media - Valid scenario       | Valid input           | Expected output   | Positive |
| UT17         | Send Media       | Send Media - Invalid input 1      | Invalid input         | Error message     | Negative |
| UT18         | Send Media       | Send Media - Invalid input 2      | Empty/Incorrect input | Validation failed | Negative |
| UT19         | Voice Call       | Voice Call - Valid scenario       | Valid input           | Expected output   | Positive |
| UT20         | Voice Call       | Voice Call - Invalid input 1      | Invalid input         | Error message     | Negative |
| UT21         | Voice Call       | Voice Call - Invalid input 2      | Empty/Incorrect input | Validation failed | Negative |
| UT22         | Video Call       | Video Call - Valid scenario       | Valid input           | Expected output   | Positive |
| UT23         | Video Call       | Video Call - Invalid input 1      | Invalid input         | Error message     | Negative |
| UT24         | Video Call       | Video Call - Invalid input 2      | Empty/Incorrect input | Validation failed | Negative |
| UT25         | Call Controls    | Call Controls - Valid scenario    | Valid input           | Expected output   | Positive |
| UT26         | Call Controls    | Call Controls - Invalid input 1   | Invalid input         | Error message     | Negative |
| UT27         | Call Controls    | Call Controls - Invalid input 2   | Empty/Incorrect input | Validation failed | Negative |
| UT28         | Network Handling | Network Handling - Valid scenario | Valid input           | Expected output   | Positive |

Figure 5.5: Unit testing 1

|      |                   |                                     |                       |                   |          |
|------|-------------------|-------------------------------------|-----------------------|-------------------|----------|
| UT29 | Network Handling  | Network Handling - Invalid input 1  | Invalid input         | Error message     | Negative |
| UT30 | Network Handling  | Network Handling - Invalid input 2  | Empty/Incorrect input | Validation failed | Negative |
| UT31 | UI Responsiveness | UI Responsiveness - Valid scenario  | Valid input           | Expected output   | Positive |
| UT32 | UI Responsiveness | UI Responsiveness - Invalid input 1 | Invalid input         | Error message     | Negative |
| UT33 | UI Responsiveness | UI Responsiveness - Invalid input 2 | Empty/Incorrect input | Validation failed | Negative |
| UT34 | Logout            | Logout - Valid scenario             | Valid input           | Expected output   | Positive |
| UT35 | Logout            | Logout - Invalid input 1            | Invalid input         | Error message     | Negative |
| UT36 | Logout            | Logout - Invalid input 2            | Empty/Incorrect input | Validation failed | Negative |
| UT37 | Forgot Password   | Forgot Password - Valid scenario    | Valid input           | Expected output   | Positive |
| UT38 | Forgot Password   | Forgot Password - Invalid input 1   | Invalid input         | Error message     | Negative |
| UT39 | Forgot Password   | Forgot Password - Invalid input 2   | Empty/Incorrect input | Validation failed | Negative |
| UT40 | Profile Update    | Profile Update - Valid scenario     | Valid input           | Expected output   | Positive |
| UT41 | Profile Update    | Profile Update - Invalid input 1    | Invalid input         | Error message     | Negative |
| UT42 | Profile Update    | Profile Update - Invalid input 2    | Empty/Incorrect input | Validation failed | Negative |
| UT43 | Search Mentor     | Search Mentor - Valid scenario      | Valid input           | Expected output   | Positive |
| UT44 | Search Mentor     | Search Mentor - Invalid input 1     | Invalid input         | Error message     | Negative |
| UT45 | Search Mentor     | Search Mentor - Invalid input 2     | Empty/Incorrect input | Validation failed | Negative |

Figure 5.6: Unit testing 2

### 5.1.2 Integration Testing

| Test Case ID | Module             | Test Scenario                        | Input          | Expected Output  | Status   |
|--------------|--------------------|--------------------------------------|----------------|------------------|----------|
| IT01         | Login → Home       | Login → Home - Valid flow            | Valid flow     | Expected outcome | Positive |
| IT02         | Login → Home       | Login → Home - Error in step 1       | Step 1 failure | Flow blocked     | Negative |
| IT03         | Login → Home       | Login → Home - Error in step 2       | Step 2 failure | Error message    | Negative |
| IT04         | Signup → Login     | Signup → Login - Valid flow          | Valid flow     | Expected outcome | Positive |
| IT05         | Signup → Login     | Signup → Login - Error in step 1     | Step 1 failure | Flow blocked     | Negative |
| IT06         | Signup → Login     | Signup → Login - Error in step 2     | Step 2 failure | Error message    | Negative |
| IT07         | Home → Mentor List | Home → Mentor List - Valid flow      | Valid flow     | Expected outcome | Positive |
| IT08         | Home → Mentor List | Home → Mentor List - Error in step 1 | Step 1 failure | Flow blocked     | Negative |
| IT09         | Home → Mentor List | Home → Mentor List - Error in step 2 | Step 2 failure | Error message    | Negative |
| IT10         | Mentor → Connect   | Mentor → Connect - Valid flow        | Valid flow     | Expected outcome | Positive |
| IT11         | Mentor → Connect   | Mentor → Connect - Error in step 1   | Step 1 failure | Flow blocked     | Negative |
| IT12         | Mentor → Connect   | Mentor → Connect - Error in step 2   | Step 2 failure | Error message    | Negative |
| IT13         | Mentor → Chat      | Mentor → Chat - Valid flow           | Valid flow     | Expected outcome | Positive |
| IT14         | Mentor → Chat      | Mentor → Chat - Error in step 1      | Step 1 failure | Flow blocked     | Negative |
| IT15         | Mentor → Chat      | Mentor → Chat - Error in step 2      | Step 2 failure | Error message    | Negative |
| IT16         | Chat → Call        | Chat → Call - Valid flow             | Valid flow     | Expected outcome | Positive |
| IT17         | Chat → Call        | Chat → Call - Error in step 1        | Step 1 failure | Flow blocked     | Negative |
| IT18         | Chat → Call        | Chat → Call - Error in step 2        | Step 2 failure | Error message    | Negative |
| IT19         | Chat → Media       | Chat → Media - Valid flow            | Valid flow     | Expected outcome | Positive |

Figure 5.7: Integration Testing 1

|      |                    |                                  |                |                  |          |
|------|--------------------|----------------------------------|----------------|------------------|----------|
| IT20 | Chat → Media       | Chat → Media - Error in step 1   | Step 1 failure | Flow blocked     | Negative |
| IT21 | Chat → Media       | Chat → Media - Error in step 2   | Step 2 failure | Error message    | Negative |
| IT22 | Call → End         | Call → End - Valid flow          | Valid flow     | Expected outcome | Positive |
| IT23 | Call → End         | Call → End - Error in step 1     | Step 1 failure | Flow blocked     | Negative |
| IT24 | Call → End         | Call → End - Error in step 2     | Step 2 failure | Error message    | Negative |
| IT25 | Profile → Edit     | Profile → Edit - Valid flow      | Valid flow     | Expected outcome | Positive |
| IT26 | Profile → Edit     | Profile → Edit - Error in step 1 | Step 1 failure | Flow blocked     | Negative |
| IT27 | Profile → Edit     | Profile → Edit - Error in step 2 | Step 2 failure | Error message    | Negative |
| IT28 | App Launch → Login | App Launch → Login - Valid flow  | Valid flow     | Expected outcome | Positive |

Figure 5.8: Integration Testing 2

|      |                      |  |                |                  |          |
|------|----------------------|--|----------------|------------------|----------|
| IT29 | App Launch → Login   | App Launch → Login - Error in step 1   | Step 1 failure | Flow blocked     | Negative |
| IT30 | App Launch → Login   | App Launch → Login - Error in step 2   | Step 2 failure | Error message    | Negative |
| IT31 | Search → View Mentor | Search → View Mentor - Valid flow      | Valid flow     | Expected outcome | Positive |
| IT32 | Search → View Mentor | Search → View Mentor - Error in step 1 | Step 1 failure | Flow blocked     | Negative |
| IT33 | Search → View Mentor | Search → View Mentor - Error in step 2 | Step 2 failure | Error message    | Negative |
| IT34 | Mentor → Reviews     | Mentor → Reviews - Valid flow          | Valid flow     | Expected outcome | Positive |
| IT35 | Mentor → Reviews     | Mentor → Reviews - Error in step 1     | Step 1 failure | Flow blocked     | Negative |
| IT36 | Mentor → Reviews     | Mentor → Reviews - Error in step 2     | Step 2 failure | Error message    | Negative |
| IT37 | Connect → Accept     | Connect → Accept - Valid flow          | Valid flow     | Expected outcome | Positive |
| IT38 | Connect → Accept     | Connect → Accept - Error in step 1     | Step 1 failure | Flow blocked     | Negative |
| IT39 | Connect → Accept     | Connect → Accept - Error in step 2     | Step 2 failure | Error message    | Negative |
| IT40 | Home → Notifications | Home → Notifications - Valid flow      | Valid flow     | Expected outcome | Positive |
| IT41 | Home → Notifications | Home → Notifications - Error in step 1 | Step 1 failure | Flow blocked     | Negative |
| IT42 | Home → Notifications | Home → Notifications - Error in step 2 | Step 2 failure | Error message    | Negative |
| IT43 | Mentor List → Filter | Mentor List → Filter - Valid flow      | Valid flow     | Expected outcome | Positive |
| IT44 | Mentor List → Filter | Mentor List → Filter - Error in step 1 | Step 1 failure | Flow blocked     | Negative |
| IT45 | Mentor List → Filter | Mentor List → Filter - Error in step 2 | Step 2 failure | Error message    | Negative |

Figure 5.9: Integration Testing 3

## 5.2 Automation Testing

| No | Tool Used | Test Description                                   | Module Automated       | Test Script Reference                        | Framework Used  | Test Result                 |
|----|-----------|--|------------------------|--|-----------------|-----------------------------|
| 1  | Appium    | Validate login with valid/invalid credentials      | Login                  | test_login.py – test_valid_login()           | PyTest + Appium | Pass                        |
| 2  | Appium    | Automate user sign-up with form validation         | Signup                 | test_signup.py – test_valid_signup()         | PyTest + Appium | Pass                        |
| 3  | Appium    | Navigate mentor profiles and view details          | Mentor Profile View    | test_mentor_profile.py – test_view_profile() | PyTest + Appium | Pass                        |
| 4  | Appium    | Send/receive chat messages and images              | Chat                   | test_chat.py – test_send_receive_chat()      | PyTest + Appium | Pass                        |
| 5  | Appium    | Start and end voice/video calls                    | Video/Audio Call       | test_call.py – test_start_call()             | PyTest + Appium | Pass                        |
| 6  | Appium    | Upload and update profile image                    | Profile Management     | test_profile.py – test_upload_picture()      | PyTest + Appium | Pass                        |
| 7  | Appium    | Search and filter mentors by category              | Mentor Search & Filter | test_search.py – test_filter_by_field()      | PyTest + Appium | Pass                        |
| 8  | Appium    | Test AI chatbot response to user queries           | AI Chatbot             | test_chatbot.py – test_bot_response()        | PyTest + Appium | Pass                        |
| 9  | Appium    | Test app behavior while switching networks         | Network & Connectivity | test_network.py – test_wifi_to_data()        | PyTest + Appium | Pass                        |
| 10 | Appium    | Test layout responsiveness on various screen sizes | UI Responsiveness      | test_ui.py – test_responsive_layout()        | PyTest + Appium | Fail (lag on small devices) |
| 11 | Appium    | Test mute/unmute/camera during active call         | Call Controls          | test_controls.py – test_toggle_buttons()     | PyTest + Appium | Fail (toggle delay)         |

Figure 5.10: Automation Testing

# Chapter 6

## Performance Evaluation

### 6.1 Evaluation Objectives

The primary objectives of the performance evaluation were:

- To measure responsiveness and speed of core functionalities (chat, call, and scheduling).
- To assess system behavior under poor network conditions.
- To identify and reduce UI lags and loading delays.
- To evaluate the effectiveness of exception handling and error recovery mechanisms.
- To verify database query response times and backend efficiency.

### 6.2 Tools and Environment

Several tools and platforms were used to carry out performance testing, including:

- Firebase Performance Monitoring: Used to track response time of APIs, screen rendering times, and app startup latency.
- Android Profiler (Android Studio): Used to monitor memory usage, CPU consumption, and network activity.
- ZEGOCLOUD Logs: Helped measure audio/video call stability, delay, and connection quality.
- Cloudinary Reports: Analyzed image and video load speeds from the cloud.
- 

### 6.3 Performance Metrics and Benchmarks

The app was evaluated against the following performance metrics:

- App Launch Time: Time taken from tapping the icon to home screen load.
- Message Delivery Time: Delay between sending and receiving chat messages.

- Call Connection Time: Time taken to initiate and connect video/voice calls.
- Image/Video Load Time: How quickly mentor profile images and shared media load.
- UI Lag/Frame Drops: How frequently the interface freezes or drops frames during interactions.
- Error Recovery: App's response to network disconnection and API failures.

## 6.4 Key Results

### 6.4.1 App Startup and Navigation

- Average cold start time: 2.8 seconds on mid-range devices.
- Smooth navigation observed between screens.
- Initial lag noted on low-end devices during first-time media load.

### 6.4.2 Chat and Messaging

- Messages were delivered in  $\pm 1$  second on stable internet.
- Message timestamps, typing indicators, and media attachments worked as expected.
- Image sharing had a 3-5 second delay depending on file size and network speed.

### 6.4.3 Audio and Video Calling

- Voice calls established in 1.5 seconds on average.
- Video call connection took 2.5 to 4 seconds, slightly delayed on low bandwidth.
- Lag observed during first 2 seconds of video on slow networks, but recovered quickly.
- Call dropped gracefully on disconnect and resumed upon reconnection (ZEGOCLOUD auto-reconnect feature).

### Appointment Scheduling

- Appointment creation took  $\pm 500$ ms on stable networks.
- Input validation prevented duplicate or invalid bookings.
- Firebase Firestore update time was consistently under 1 second.

### 6.4.4 UI Performance

- Minor UI stutter observed on scrolling mentor cards on older devices.
- UI responsiveness was optimal on 60Hz+ screens.
- Average screen rendering time: 14ms, within acceptable mobile range.

## 6.5 Error Handling and Resilience

- Network disconnection was successfully detected.
- Once internet resumed, chat auto-refreshed and active sessions resumed smoothly.
- Calls failed gracefully with appropriate messages (“User unavailable”, “Call failed”).

## 6.6 Limitations Identified

Despite generally strong performance, some limitations were noted:

- UI delays on older Android devices, especially during image rendering.
- Audio/video sync issues in first few seconds of weak network calls.
- No in-app retry button for failed media uploads.
- App still lacks complete background notification support for chat.

## 6.7 Optimization Actions Taken

In response to performance evaluations, the following improvements were implemented:

- Implemented lazy loading for mentor images.
- Reduced image resolution for previews, loading full version only on tap.
- Used Firebase indexes to optimize queries on large appointment datasets.
- Added retry logic and loading indicators for image/video sharing.
- Implemented caching for recent chat messages and user profiles.

# Chapter 7

## Business Model

### 7.1 Market Research

#### **Pinpointing a Niche:**

Direct attention toward students and youth in Karachi keen on business yet lacking direction. Pinpoint the underserved segments without established structured mentorship platforms.

#### **Business Clients:**

Institutions of higher education and vocational training centers. Business mentors, trainers, and incubators that operate locally.

#### **Competitor Analysis:**

Scout out existing resources, including platforms such as Coursera, YouTube channels, and local training centers. Carry out SWOT analyses on each of those platforms within the framework of the Karachi market.

### 7.2 Unique Value Proposition (UVP)

- Provide AI-driven business assistance around the clock.
- Offer personalized video mentoring through sessions with local business professionals.
- A seamless user experience for users of low-end as well as mid-range smartphones.
- The content is carefully curated to mirror local culture, conventional business practices, and tangible real-world case studies.

### 7.3 Targeted Audience

#### **Customer Segmentation:**

- Students of high school, college, and early university level.
- Young people out of work or at the beginning of their careers desiring to know more about startups.

**User Personas:**

- Professionals between the age of 16-25 years.
- Karachi commerce and business stream students.
- Women users who want to make business at home.

## 7.4 Monetization Strategy

- Freemium model: Free content or premium mentorship packages.
- Subscription for AI-powered chatbot with advanced features.
- Institutional licensing for schools and training centers.
- Brand partnerships with local business communities.

## 7.5 Scalability and Internationalization

**Scalability:**

- Built on Flutter and Firebase to create a cross-platform and scale it.
- With Cloudinary, media content will be handled efficiently.

**Internationalization:**

- There are also future plans of including multiple languages.
- Gemini has AI that allows flexibility in international business inquiries.

## 7.6 Propose Business Plan

**Phase 1:**

Start the app in Karachi with a small group of mentors.

**Phase 2:**

Collect feedback from users and add more mentors.

**Phase 3:**

Add paid plans and promote the app in colleges and on social media.

## 7.7 Integration with Final Year Project

This mentorship mobile application is designed as part of the Final Year Project, aligning both academic objectives and real-world impact. The project addresses a genuine need in society by guiding students and inexperienced individuals who are interested in business but lack access to proper mentorship.

### 7.7.1 Project Alignment

The idea supports educational and entrepreneurial development, which directly fits the theme of innovation and social impact encouraged by final year projects. It showcases how technology can solve local problems effectively.

### 7.7.2 Implementation Details

The project is developed using:

- Flutter for mobile app development
- Firebase for backend, authentication, and database
- ZEGOCLOUD for video/audio calling
- Cloudinary for image and media handling
- Gemini (AI) for chatbot integration
- Figma for UI/UX design and prototyping

These tools were chosen to ensure real-time performance, scalability, and a good user experience.

### 7.7.3 Market Experience:

To gain practical understanding, surveys and interviews were conducted with local students and early-stage entrepreneurs in Karachi. Their feedback helped shape the app's features, especially the one-on-one calling and AI guidance system. This hands-on exposure added valuable real-world learning to the academic project.

## 7.8 Future Road map

This project aims to grow step-by-step, with planned updates and improvements based on user needs and market trends.

### 7.8.1 Future Development

- Add features like in-app payments and appointment scheduling.
- Build a mentor rating and review system.
- Improve the AI chatbot with more advanced business knowledge.
- Introduce discussion forums for peer learning.

### 7.8.2 Adaptability

- The app will be tested and adjusted based on feedback from different cities.
- UI and features will be adapted for different user groups like college students, early entrepreneurs, and women-led startups.
- Designed to be compatible with both Android and iOS in future versions.

## 7.9 International Expansion

After successful implementation in Karachi, the app will expand to other cities in Pakistan. Later, it will target other developing countries with similar needs, starting with South Asia and the Middle East.

## 7.10 Legal Considerations

- Terms and conditions, privacy policy, and mentorship disclaimers will be drafted.
- User data will be protected under local data protection regulations.
- The app will comply with digital platform and e-commerce laws in Pakistan.

## 7.11 Automation Testing

- Automated testing will be used to check key features like login, video calls, and chatbot replies.
- Tools like Firebase Test Lab and integration testing frameworks in Flutter will be used.
- This ensures that each update runs smoothly without breaking existing features.

# Chapter 8

## Future Directions and Conclusion

### 8.1 Future Directions:

The current version of our mobile application provides a functional and scalable platform for entrepreneurs and students in Karachi to connect with business mentors through voice/video calls, messaging, and multimedia communication. However, several key enhancements are planned to expand the capabilities and impact of the system:

#### Expanded User Base Location Support:

- Extend access beyond Karachi to other cities in Pakistan and internationally.
- Enable multilingual support (Urdu, English, regional languages).

#### In-App Payments Subscription Plans:

- Add secure in-app payment gateway for paid mentorship sessions.
- Introduce freemium models and tiered subscription plans.

#### Performance Optimization:

- Improve responsiveness and reduce UI lag observed during manual testing.
- Optimize video and voice call quality for low-bandwidth users.

#### AI-Powered Features:

- Smart mentor recommendations using machine learning.
- Chat summarization and session transcription using NLP.

#### Feedback Ratings:

- Let mentees rate mentors after calls.
- Provide analytics dashboards for mentors to track impact.

**Admin Panel Development:**

- Full-featured backend portal for admin control, reporting, user management, and analytics.

## 8.2 Conclusion:

This project successfully delivers a mobile-based mentorship platform tailored to early-stage entrepreneurs and students in Karachi. It effectively bridges the gap between experienced mentors and aspiring business individuals through easy-to-use features like voice and video calling, chat, and media sharing. The manual and automated testing efforts confirmed that the application's core functionalities are working correctly, with some minor UI performance areas to be optimized in future versions.

The project not only aligns with the local entrepreneurship ecosystem's needs but also provides a scalable model for broader impact. With planned improvements and user-centered innovation, this platform can become a critical support system for the entrepreneurial journey in Pakistan and beyond.

# Appendix A - User Manual

## 8.3 Getting Started

### 8.3.1 Installation:

- Download the APK from the provided link or QR code.
- Install and launch the app on your Android device.

### 8.3.2 Login / Signup:

- **New users:** Tap “Signup” and fill in required details.
- **Existing users:** Use “Login” with valid credentials.
- Incorrect or empty fields will show an error prompt.

## 8.4 Navigating the App

### 8.4.1 User Authentication

- **Login:** Enter email and password to access the homepage.
- **Signup:** Fill in Name, Email, Phone Number, and Password to register.
- Displays errors for wrong or empty fields.

### 8.4.2 Choosing a Mentor

- View available mentors from the homepage.
- Tap on any mentor card to view their profile and details.

### 8.4.3 Send Connection Request

- Tap ”Connect” button on mentor’s profile.
- Mentor receives the request in their dashboard.

#### 8.4.4 Mentor Accepting Request

- Mentor can approve the connection request.
- Once accepted, a chat screen is automatically enabled between both parties.

### 8.5 Communication Features

#### 8.5.1 Text Messaging

- Type and send real-time messages through the chat screen.
- Connection errors or empty messages trigger alerts.

#### 8.5.2 Send Media Files

Use gallery options to send:

- Images
- Videos

File size limits apply.

#### 8.5.3 Voice Messages

- Long press the mic icon to record a message.
- Requires microphone permission.

#### 8.5.4 Voice Call

- Initiate voice call from chat screen.
- If accepted, voice call interface appears.
- Call status is displayed for success or failure.

#### 8.5.5 Video Call

- Start video call using the camera icon in chat.
- Permission required for camera and microphone.
- If the other user accepts, live video begins.

#### 8.5.6 Call Controls

- Call screen includes mute, speaker toggle, camera toggle, and end call button.
- Call duration and status are displayed in real-time.

## **8.6 Chatbot Feature**

### **8.6.1 Smart Chatbot Assistant**

- Accessible from home or help screen.
- Ask questions about basic business topics such as startup guidance, marketing, funding, etc.
- Supports sending media files (images/videos) to the chatbot.
- Chatbot responds with relevant business insights or suggests mentor categories accordingly.

# **Appendix B- Software Requirement Specification (SRS)**

# **Appendix C- Software Design Description (SDD)**



# Appendix D- Dissemination Activity

## *Entrepreneur Growth Guide:*

### *An AI-Driven Mobile Mentorship Platform for Empowering Karachi's Young Innovators*

Syed Husain Ahmed

Department of Software Engineering  
Sir Syed University of Engineering &  
Technology  
Karachi, Pakistan  
[syedhusainahmed1@gmail.com](mailto:syedhusainahmed1@gmail.com)

Abdullah Ansari

Department of Software Engineering  
Sir Syed University of Engineering &  
Technology  
Karachi, Pakistan  
[abdullahansari7@gmail.com](mailto:abdullahansari7@gmail.com)

**Abstract—** In today's rapidly evolving business landscape, aspiring entrepreneurs, particularly in underserved urban regions like Karachi, face critical challenges due to the lack of structured mentorship and accessible resources. The *Entrepreneur Growth Guide* is a mobile application designed to address this gap by offering a virtual mentorship ecosystem. The platform connects students, freelancers, and early-stage entrepreneurs with experienced business mentors through features such as one-on-one messaging, voice and video calling, and an AI-powered chatbot for instant guidance.

This paper presents the complete development lifecycle of the project, from problem identification and user requirement analysis to design, implementation, testing, and performance evaluation. The application leverages Flutter for cross-platform development, Firebase for real-time backend support, and ZEGOCLOUD for seamless audio-video communication. Furthermore, the integration of AI enhances user interaction by delivering 24/7 automated business advice and mentorship suggestions.

Market research tailored to the local context highlights the needs of Karachi's entrepreneurial youth, leading to a scalable, culturally relevant solution. A business model is incorporated to ensure sustainability through freemium plans and institutional licensing. Rigorous manual and automated testing confirmed functionality and responsiveness.

By combining mobile technology with strategic business insights, the *Entrepreneur Growth Guide* offers a socially impactful and technically robust solution, with the potential for expansion across other regions in Pakistan and beyond, empowering the next generation of entrepreneurs through digital mentorship.

**Keywords —** Business mentorship, mobile application, AI chatbot, Flutter, Firebase, entrepreneurship, Karachi.

#### Introduction

Entrepreneurship in developing economies is often hindered by limited access to mentorship, financial literacy, and practical business knowledge. While Karachi has a growing entrepreneurial ecosystem, many students and early-stage founders lack the structured support necessary for sustainable business growth. High costs, limited mentor availability, and fragmented networking opportunities exacerbate this problem.

The *Entrepreneur Growth Guide* addresses these challenges by providing a mobile-based mentorship platform. Through accessible features such as chat, video conferencing, and AI-powered guidance, the system ensures that users can connect with mentors anytime, anywhere, using a smartphone.

#### Problem Statement

Despite the presence of incubators and training centers, a large proportion of youth in Karachi lack access to experienced mentors due to cost, exclusivity, and logistical constraints. This project aims to bridge the gap by offering a low-cost, mobile-first mentorship ecosystem tailored to local needs.

#### 1. Objectives

- Provide an accessible mentorship network for students and early entrepreneurs.
- Enable real-time communication via text, audio, and video channels.
- Integrate AI-powered chatbots for instant guidance.
- Ensure scalability and cultural relevance for Pakistan's entrepreneurial ecosystem.

#### 2. Literature Review

Mentorship has been identified as a key factor in entrepreneurial success, providing guidance in market analysis, funding, and growth strategies [1]. Global platforms such as MentorCloud and PushFar demonstrate the potential of online mentorship, but their high costs and lack of localization limit accessibility for developing nations.

Studies in Pakistan reveal a gap between business education and practical entrepreneurship training [2]. Government colleges often focus on theoretical concepts with limited exposure to industry practices, leaving many students underprepared for business challenges.

Mobile-first solutions are gaining traction in education and entrepreneurship due to high smartphone penetration [3]. Localized mobile platforms with real-time engagement capabilities are shown to significantly improve mentorship reach and effectiveness.

### 3. Methodology

The project adopted an Agile Scrum approach to ensure iterative development and stakeholder feedback integration. The main tools and technologies include:

- Flutter for cross-platform mobile application development.
- Firebase for authentication, cloud storage, and database management.
- ZEGOCLOUD for high-quality audio and video communication.
- Cloudinary for optimized media storage.
- Gemini AI for chatbot implementation.

The application architecture follows a modular MVC pattern for maintainability and scalability.

### 4. System Development

The platform supports two primary user roles: *Mentor* and *Mentee*. Key functionalities include:

- User Registration & Role Selection
- Profile Management & Mentor Categorization
- One-on-One Messaging
- Audio & Video Calling
- AI Chatbot Support

Backend integration ensures secure authentication, while the chatbot provides instant answers to common business queries, reducing dependency on live mentors for basic guidance.

### 5. Testing & Evaluation

Testing comprised both manual and automated procedures. Manual testing validated functional flows, including registration, login, messaging, and video calling. Automated testing using Firebase Test Lab ensured compatibility across devices and stability after updates.

Performance evaluation confirmed low latency in communication and efficient media handling under varying network conditions.

### 6. Business Model

Market research identified students, young professionals, and women entrepreneurs as the primary target segments. The monetization strategy includes:

- Freemium Access - Basic mentorship for free, premium packages for advanced services.
- Institutional Licensing - For universities and training centers.
- Brand Partnerships - Collaborations with local business communities.

The Unique Value Proposition (UVP) lies in localized content, AI-driven support, and compatibility with low-end smartphones.

### 7. Future Work

Planned enhancements include:

- In-app payment integration for paid mentorship sessions.
- Multilingual support for Urdu and regional languages.
- AI-based mentor recommendation systems.
- Discussion forums for peer learning.
- Expansion to other Pakistani cities and developing nations.

### 8. Conclusion

The *Entrepreneur Growth Guide* successfully demonstrates how mobile technology can bridge mentorship gaps in Karachi's entrepreneurial ecosystem. By combining human expertise with AI-powered support, the platform offers a sustainable, scalable, and culturally relevant solution. With continued development, it holds potential for national and international impact in fostering entrepreneurship.

### References

- [1] C. M. Christensen and M. E. Raynor, *The Innovator's Solution: Creating and Sustaining Successful Growth*. Boston, MA: Harvard Business Review Press, 2013.
- [2] M. U. Hassan, S. Shaukat, M. A. Nawaz, and S. Naz, "Exploring entrepreneurial intentions among university students in Pakistan," *Asia Pacific Journal of Innovation and Entrepreneurship*, vol. 14, no. 3, pp. 325–342, 2020.
- [3] World Bank, *Pakistan Jobs Diagnostic: Creating More and Better Jobs*, Washington, DC, USA, 2020.
- [4] Google Developers, "Flutter Documentation," 2024. [Online]. Available: <https://flutter.dev>
- [5] Firebase, "Firebase Documentation – App Development Platform," 2024. [Online]. Available: <https://firebase.google.com/docs>
- [6] ZegoCloud, "Real-time Engagement for Mobile Apps," 2024. [Online]. Available: <https://www.zegocloud.com>
- [7] E. Devaney, "How AI is Changing the Future of Entrepreneurship," HubSpot Blog, 2023. [Online]. Available: <https://blog.hubspot.com>
- [8] P. A. Lemoine and M. D. Richardson, "Artificial intelligence in higher education: Challenges and opportunities," *International Journal of Education and Practice*, vol. 7, no. 4, pp. 411–420, 2019.

Figure 8.2: research paper 2

# Appendix E- Marketing / Promotional Material

## 8.7 Poster

**Project Scope**  
The Entrepreneur Growth Guide App is a mobile platform that supports entrepreneurs through every stage of their business journey. It offers strategic guidance, mentorship, and educational resources to help users start, manage, and grow their businesses with practical, expert-backed insights.

**Project Features**

- One-on-One Chat: Private text messaging between mentors and mentees.
- Audio Calling: Voice communication for more personal and effective conversations.
- Video Calling: Face-to-face mentorship through in-app video conferencing.
- AI-powered chatbot: Provides instant support and answers user queries.
- Seamless scheduling feature: Allows users to book mentorship sessions with experts easily.

**Overall System Model**

The system model illustrates the architecture of the Entrepreneur Growth Guide App. It includes the following components:

- Entrepreneur Resources:** Browse mentors, Request support, Expert Guidance.
- Mentorship & Communication:** Connect with mentors/experts, Chat with mentor, Call with mentor.
- AI Support:** Mentor suggestion from Chatbot, Personalized business advices, Constant availability.
- Entrepreneur Growth Guide App:** The central application connecting all components.
- Database:** Stores user data, Secure Access, Retrieves profiles.
- User Authentication:** Sign In / Sign Up, Email verification, Secure access.

**Flow chart Diagram**

```

    graph TD
        Start((USER SIGN UP / LOGIN)) --> SignUp[SIGN UP]
        Start --> LogIn[LOG IN]
        
        SignUp -- NOT VERIFIED --> VerifyEmail[EMAIL VERIFICATION]
        VerifyEmail -- NOT REGISTERED --> Register[REGISTER]
        VerifyEmail -- VERIFIED --> Dashboard[ACCESS DASHBOARD]
        
        LogIn -- NOT REGISTERED --> Register
        LogIn --> Dashboard
        
        Dashboard --> ViewMentors[VIEW MENTOR PROFILES]
        
        ViewMentors --> UseAI[USE AI CHATBOT FOR MENTOR RECOMMENDATIONS?]
        
        UseAI -- YES --> GetRecommendations[GET RECOMMENDATIONS]
        UseAI -- NO --> ChooseMentor[CHOOSE A MENTOR]
        
        GetRecommendations --> StartCall[START ONE-ON-ONE CHAT OR SCHEDULE CALL]
        ChooseMentor --> StartCall
        
        StartCall -- CHAT --> ChatSession[MENTORSHIP SESSION (CHAT)]
        StartCall -- CALL --> CallSession[MENTORSHIP SESSION (CALL)]
        
        ChatSession -- PROVIDE FEEDBACK --> ThankFeedback[THANK YOU FOR YOUR FEEDBACK!]
        CallSession -- PROVIDE FEEDBACK --> ThankFeedback
        
        ThankFeedback -- YES --> EndSession[END SESSION]
        ThankFeedback -- NO --> EndSession
    
```

**Project Goals & Objectives**

- Provide Actionable Insights
- Facilitate Business Development
- Empower Decision Making
- Promote Continuous Learning
- Build a Supportive Community

**Tools Used**

- ZEGOCLOUD
- Gemini
- Figma
- Flutter
- Cloudinary
- Firebase

Figure 8.3: Poster

## 8.8 Standee



Figure 8.4: Standee

## 8.9 Brochure



Figure 8.5: Brochure Front

### 📌 System Diagram

```

graph TD
    ER[Entrepreneur Resources] --> EGGA[Entrepreneur Growth Guide App]
    DB[Database] --> EGGA
    AI[AI Support] --> EGGA
    UA[User Authentication] --> EGGA
    MC[Mentorship & Communication] --> EGGA
    EGGA --> ER
    EGGA --> DB
    EGGA --> AI
    EGGA --> UA
    EGGA --> MC
  
```

### 📌 Project Scope

The Entrepreneur Growth Guide App is a mobile platform that supports entrepreneurs through every stage of their business journey. It offers strategic guidance, mentorship, and educational resources to help users start, manage, and grow their businesses with practical, expert-backed insights.

*The Power Tool for Passionate Founders.*

### 📌 App View

### 📌 Project Concept

The Entrepreneur Growth Guide app is a user-friendly platform that helps aspiring and early-stage entrepreneurs plan, launch, and grow their businesses. It provides essential tools, goal tracking, strategic insights, and curated resources to turn ideas into successful ventures with confidence and clarity.

### 📌 App Features

- ◆ **User Authentication :** Simple and secure user registration with email verification.
- ◆ **One-on-One Chat & Audio Calling:** Private text and voice communication for effective mentor-mentee interaction.
- ◆ **AI-powered chatbot:** Provides instant support and answers user queries efficiently.
- ◆ **Seamless scheduling feature :** Allows users to book mentorship sessions with experts easily.

### 📌 Why Choose Our App?

- ◆ **Actionable Insights** Gain practical business knowledge.
- ◆ **Expert Mentorship** Connect with industry professionals.
- ◆ **AI-Powered Support** Get instant assistance anytime.
- ◆ **Seamless Scheduling** Book mentorship sessions with ease.

Figure 8.6: Brochure Back

# References

- [1] L. Pittaway and J. Cope, "Entrepreneurship Education: A Systematic Review of the Evidence," *International Small Business Journal*, vol. 25, no. 5, pp. 479–510, 2007. Available: <https://doi.org/10.1177/0266242607080656>
- [2] S. R. Isaacs, R. Visser, and C. Friedrich, "Mentorship for Entrepreneurial Development: A Critical Review," *Journal of Entrepreneurship Education*, vol. 21, no. 4, pp. 1–12, 2018.
- [3] K. Moberg, "Two Approaches to Entrepreneurship Education: The Different Effects of Education for and through Entrepreneurship," *The International Journal of Management Education*, vol. 12, no. 3, pp. 512–528, 2014. Available: <https://doi.org/10.1016/j.ijme.2014.05.002>
- [4] R. McKenzie and D. Woodruff, "Business Training and Female Enterprise Start-Up, Growth, and Dynamics: Experimental Evidence from Sri Lanka," *Journal of Development Economics*, vol. 106, pp. 199–210, 2014. Available: <https://doi.org/10.1016/j.jdeveco.2013.09.005>
- [5] J. G. Greve and M. Salaff, "Social Networks and Entrepreneurship," *Entrepreneurship Theory and Practice*, vol. 28, no. 1, pp. 1–22, 2003. Available: <https://doi.org/10.1111/1540-8520.t01-1-00029>
- [6] A. Goyal, S. Mishra and K. Sharma, "AI Chatbots in Learning Platforms: Adoption and Impact in Entrepreneurship Education," *2021 IEEE Global Engineering Education Conference (EDUCON)*, Vienna, 2021, pp. 175–180. Available: <https://doi.org/10.1109/EDUCON46332.2021.9453971>
- [7] M. Fatima and M. Ramzan, "Mobile Learning Trends and Their Role in Enhancing Entrepreneurial Skills Among University Students in Pakistan," *Journal of Educational Research*, vol. 23, no. 2, pp. 45–60, 2021.
- [8] A. Mohapatra and K. R. Patra, "A Study on the Role of Digital Mentoring in Supporting Young Entrepreneurs in Developing Countries," *2020 International Conference on Digital Futures (ICDF)*, Singapore, 2020. Available: <https://doi.org/10.1109/ICDF50959.2020.9342163>
- [9] K. Shabbir and M. Ghani, "A Systematic Review of AI Chatbot Integration in Mobile Apps for Personalized Learning," *2022 IEEE International Conference on Artificial Intelligence (ICAIS)*, Lahore, 2022. Available: <https://doi.org/10.1109/ICAIS55815.2022.9752123>
- [10] M. Saleem and U. Ahsan, "Design and Evaluation of a Business Mentorship Mobile Application for Pakistani Youth," *2023 IEEE Conference on Emerging Technologies for Education and Entrepreneurship (CETEE)*, Karachi, Pakistan, 2023.