BLOG CONNECT

A MINI PROJECT REPORT

Submitted by

MUNEESH P (220701175)

in partial fulfillment for the course

CS19611 – MOBILE APPLICATION DEVELOPMENT LABORATORY

of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE
RAJALAKSHMI NAGAR
THANDALAM
CHENNAI – 602 105
MAY 2025

RAJALAKSHMI ENGINEERING COLLEGE
CHENNAI - 602105

BONAFIDE CERTIFICATE

Certified that this project report **"BLOG CONNECT"** is the bonafide work of **" MUNEESH P (220701175)"** who carried out the project work (CS19611-Mobile Application Development Laboratory) under my supervision.

Dr. P.Kumar

HEAD OF THE DEPARTMENT

Professor and Head

Department of

Computer Science and Engineering

Rajalakshmi Engineering College

Rajalakshmi Nagar

Thandalam

Chennai - 602105

Mr.G.Saravana Gokul

SUPERVISOR

Assistant Professor (SG)

Department of

Computer Science and Engineering

Rajalakshmi Engineering College

Rajalakshmi Nagar

Thandalam

Chennai - 602105

TABLE OF CONTENTS

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S.Meganathan, B.E, F.I.E.,** our Vice Chairman **Mr. Abhay Meganathan, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) Thangam Meganathan, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution. Our sincere thanks to **Dr. S.N.Murugesan, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P.Kumar, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Mr. G.Saravana Gokul, M.E.,** Assistant Professor (SG), Department of Computer Science and Engineering, Rajalakshmi Engineering College for their valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Mr. B.Bhuvaneswaran, M.E.,** Assistant Professor (SG), Department of Computer Science and Engineering for his useful tips during our review to build our project.

MUNEESH P(220701175)

ABSTRACT

This project, titled BLOG CONNECT, is a mobile application developed using Android Studio with Kotlin as the programming language. It is designed to provide a seamless blogging experience for users by allowing them to register, log in, and manage their personal blog posts efficiently. The app features a secure login and signup system with credentials stored locally using SQLite. Once logged in, users can view a list of all blog posts, demonstrating an organized and user-friendly interface.

The core functionality of the application revolves around CRUD (Create, Read, Update, Delete) operations for blog posts. Users can easily create new posts, view all saved entries, and perform actions like editing and deleting existing posts. All blog data is stored in a local SQLite database, ensuring that user-generated content is persistently and reliably maintained within the app. The navigation between activities (Login, Sign Up, Blog List, Create Post, and Edit Post) is designed to be intuitive and responsive, providing a smooth user experience.

The application includes a logout option for security and session management. The interface is designed using both Compose and XML layouts to optimize performance and visual clarity. This project showcases a practical implementation of user authentication, data persistence using SQLite, and dynamic UI rendering.

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
CRUD	Create, Read, Update, Delete
IDE	Integrated Development Environment
UI	User Interface
API	Application Programming Interface
SDK	Software Development Kit
SQL	Structured Query Language
APK	Android Package

CHAPTER 1

INTRODUCTION

1.1 GENERAL

In the current era of digital communication, blogging has become an essential medium for individuals to share their ideas, thoughts, and experiences with the world. From personal journaling to professional content creation, blogs are widely used across the internet. With the increasing reliance on mobile applications for content consumption and creation, there is a growing need for user-friendly blogging platforms that are accessible on-the-go. BLOG CONNECT is a mobile application developed using Android Studio and Kotlin that empowers users to sign up, log in, create, view, edit, and delete blog posts efficiently through a clean and interactive interface. It ensures secure user management and local data storage using SQLite. BLOG CONNECT capitalizes on this trend by delivering a blogging experience tailored to mobile devices without compromising on functionality or user experience.

1.2 OBJECTIVE

The primary objective of this project is to design and develop a mobile blogging application that simplifies the process of blog creation and management for users. It aims to provide an intuitive and responsive interface that allows users to seamlessly interact with the app to perform all essential blogging activities. Key goals include secure user authentication, efficient data handling through SQLite, and the ability for users to manage their blog posts with options to create, read, update, and delete content. The app also emphasizes smooth navigation, logout functionality, and reliable local storage, ensuring a complete offline experience.

1.3 EXISTING SYSTEM

Most existing blogging platforms like WordPress or Blogger are web-based and often require internet connectivity and complex backend infrastructure. Mobile apps for such platforms usually rely heavily on cloud storage and may not function offline. Additionally, setting up or maintaining such systems often demands technical knowledge and server-side resources. These systems may also be overwhelming for users who need a lightweight and local blogging solution for simple journaling or offline content drafting purposes. Furthermore, many existing platforms require an internet connection even for basic tasks like viewing or editing blog drafts, making them less reliable in offline scenarios.

1.4 PROPOSED SYSTEM

The proposed system, BLOG CONNECT, overcomes the limitations of the existing systems by offering a standalone Android application with local storage capabilities. Built with Kotlin and SQLite, the app ensures all user data, including blog content and login credentials, are stored securely on the device. It offers core functionalities such as user registration, login/logout, and full CRUD operations for blog posts. The interface is optimized for mobile devices, making it simple for users to manage their content efficiently.

In addition to core features, BLOG CONNECT emphasizes user experience with a minimalistic and clean user interface, ensuring ease of use for individuals of all age groups. It also includes a logout option to enhance session control and security. Since all functionalities are integrated into a single compact app, users can create and manage blog content without distractions. The proposed system thus serves not only as a personal content management tool but also lays the foundation for future expansions, such as multimedia support or synchronization.

CHAPTER 2

LITERATURE REVIEW

2.1 GENERAL

The evolution of digital communication has led to an increased demand for user-generated content platforms such as blogs. Blogging platforms have become vital tools for sharing knowledge, opinions, experiences, and professional insights. With the proliferation of smartphones, mobile-based blog applications have gained prominence, offering users the flexibility to create, edit, and publish content on the go. Literature in the field highlights the importance of user-friendly interfaces, secure user authentication, and efficient data storage in mobile applications to enhance the blogging experience.

Several studies have explored the role of mobile applications in improving digital content management, emphasizing intuitive UI/UX design, integration with local databases like SQLite, and performance optimization. The need for offline capabilities in blog apps is also well documented, especially for users with inconsistent internet access. This review sets the foundation for understanding how BLOG CONNECT contributes to addressing these gaps and improving user interaction in blog-based mobile applications.

Although many web-based systems provide rich functionality, lightweight mobile alternatives that incorporate essential features like user authentication, blog CRUD operations and intuitive navigation are still developing. BLOG CONNECT aims to bridge this gap by offering a streamlined mobile experience with a focus on usability and offline storage capabilities via SQLite.

CHAPTER 3
SYSTEM DESIGN

3.1 GENERAL

System design is a critical phase in software development that translates requirements into a structured solution architecture. For the BLOG CONNECT application, the system design ensures that each module from user authentication to blog post management works seamlessly together while maintaining clarity, efficiency, and scalability. The design follows a modular and layered approach, separating concerns between user interface, logic, and data storage, which enhances maintainability and supports future updates or feature integrations.

The architecture of BLOG CONNECT is primarily based on the Model-View-Controller (MVC) pattern. The Model represents the SQLite database where all user and blog data are stored. The View includes XML layouts for different activities such as login, signup, and blog management screens. The Controller is composed of Kotlin activity files that handle user interactions, validate inputs, manage navigation, and communicate with the database. This separation allows independent development and testing of components while promoting code reuse and simplicity.

Additionally, user session management and database operations are abstracted into helper classes such as DatabaseHelper, ensuring that data manipulation and logic are not tightly coupled with the UI. The application also ensures smooth transitions between activities using explicit intents and lifecycle-aware components, allowing users to navigate without losing data. This system design also allows for scalability and enhancement.

3.1.1 SYSTEM FLOW DIAGRAM

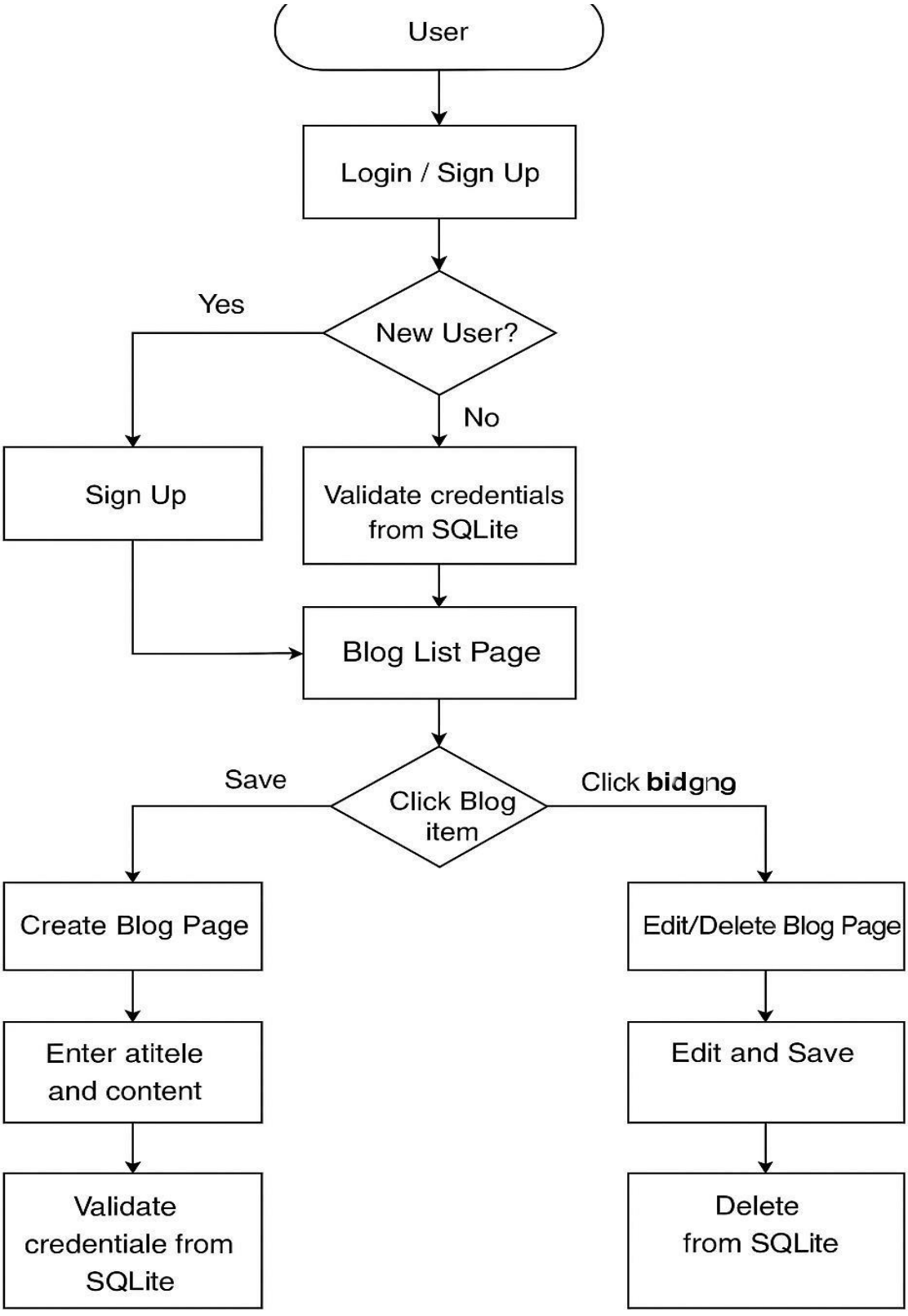

Fig 3.1

3.1.2 ARCHITECTURE DIAGRAM

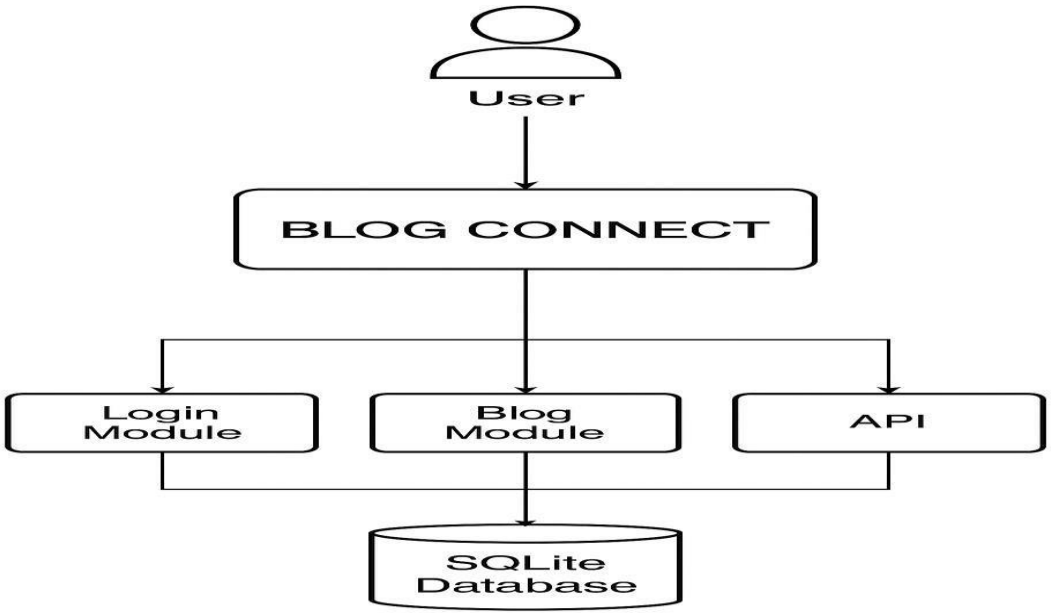

Fig 3.2

3.1.3 USECASE DIAGRAM

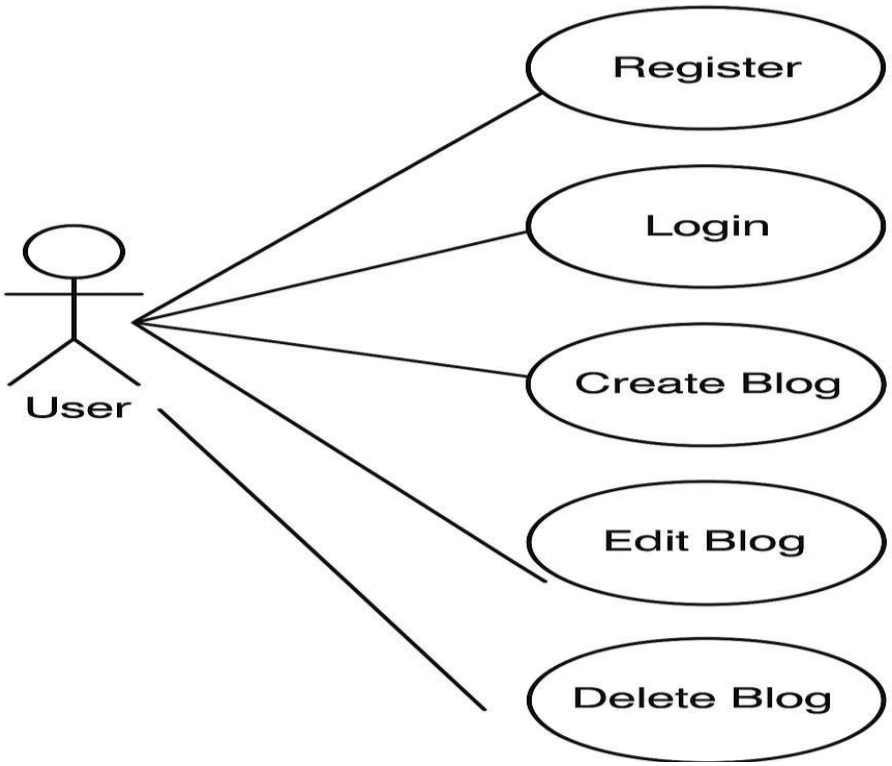

Fig3.3

CHAPTER 4

PROJECT DESCRIPTION

4.1 METHODOLOGY

The methodology adopted for the BLOG CONNECT application development is based on the Incremental Development Model, which allows the project to be built in small, functional parts and tested thoroughly at each step. This approach is well-suited for mobile app development where features can be progressively integrated, tested, and refined. Each stage of development focuses on a specific module such as user authentication or blog management ensuring that functionality is stable before moving to the next phase.

This model enhances flexibility, allowing modifications and improvements based on user feedback or testing outcomes. It also supports early detection of issues and simplifies debugging. During development, emphasis was placed on modular programming, reusable components, and adherence to Android development best practices using Kotlin and SQLite. The design was kept user-centric, focusing on ease of navigation and a clean interface for an optimal user experience.

To ensure robust functionality, the application was developed and tested in phases. Each phase consisted of coding, unit testing, and UI testing. The use of Android Studio IDE provided essential tools such as emulators, logcat for debugging, and layout designers, all of which helped accelerate the development process. SQLite was chosen for local data persistence due to its simplicity, lightweight nature, and integration with Android. Kotlin, being the officially recommended language for Android development, was used for its modern syntax, null safety, and concise code structure, which contributed to faster and more maintainable development.

4.1.1 MODULES

The BLOG CONNECT app is divided into the following key modules:

1. **User Authentication Module**

   This module handles the sign-up and login processes. It stores user credentials securely in an SQLite database and validates user login attempts. It also includes a logout feature that redirects users safely to the login screen.

2. **Create Blog Post Module**

   This module allows logged-in users to compose new blog posts by entering a title and content. The data is saved locally in the SQLite database and timestamped for future management.

3. **View Blog Posts Module**

   This module fetches and displays all existing blog posts from the database using a ListView. Each item is clickable, allowing the user to access detailed options to edit or delete the post.

4. **Edit/Delete Blog Post Module**

   This module enables users to update or remove their previously created posts. It ensures data integrity by using the unique post ID for database operations, thereby avoiding accidental overwrites or deletions.

5. **Logout and Navigation Module**

   This module manages navigation between activities and provides logout functionality to terminate user sessions securely. It ensures users are directed appropriately based on login status and actions performed within the app.

CHAPTER 5
CONCLUSION

5.1 GENERAL

The development of BLOG CONNECT has successfully achieved its intended objectives by delivering a lightweight, intuitive, and fully functional blogging platform for Android users. Through the integration of Kotlin and SQLite, the application provides essential blogging functionalities such as user authentication, post creation, editing, viewing, and deletion — all while operating without the need for internet connectivity. The modular architecture and clear navigation enhance the user experience and allow for seamless content management.

This project not only demonstrates the feasibility of building a secure, offline-capable mobile blogging application but also highlights the importance of user-centric design and efficient local data storage in mobile development. With its robust yet minimal framework, BLOG CONNECT serves as a useful tool for personal journaling or drafting content in environments where internet access is limited or unavailable.

The project also opens doors for future enhancements such as media attachments, cloud synchronization, theme customization, and multi-user support. Overall, BLOG CONNECT reflects a practical implementation of Android app development best practices, offering valuable insights into mobile UI/UX design, local database management, and activity-based navigation structures.

REFERENCES

1. Android Developers. (n.d.). *Developer Guides: Android Developer*. Retrieved from https://developer.android.com/guide
2. SQLite. (n.d.). *SQLite Documentation*. Retrieved from https://www.sqlite.org/docs.html
3. Kotlin Language Documentation. (n.d.). *Kotlin by JetBrains*. Retrieved from https://kotlinlang.org/docs/home.html
4. MindOrks. (n.d.). *Android SQLite Database Tutorial*. Retrieved from https://mindorks.com/android-tutorial
5. Vogella. (n.d.). *Android SQLite Database Tutorial*. Retrieved from https://www.vogella.com/tutorials/AndroidSQLite/article.html
6. GeeksforGeeks. (n.d.). *Android – SQLite Database*. Retrieved from https://www.geeksforgeeks.org/android-sqlite-database-in-kotlin/
7. Stack Overflow. (n.d.). *Android Development Questions and Answers*. Retrieved from https://stackoverflow.com/

APPENDICES

Appendix A: Source Code Snippets

1. User Login Validation (LoginActivity.kt)

kotlin

```
btnLogin.setOnClickListener {
    val username = etUsername.text.toString()
    val password = etPassword.text.toString()
    val isValid = db.checkUser(username, password)
    if (isValid) {
        val intent = Intent(this, BlogListActivity::class.java)
        startActivity(intent)
        finish()
    } else {
        Toast.makeText(this, "Invalid credentials",
Toast.LENGTH_SHORT).show()
    }
}
```

2. Insert Blog Post (CreateBlogPostActivity.kt)

kotlin

```
val stmt = dbWritable.compileStatement("INSERT INTO blog_posts (title,
content) VALUES (?, ?)")
stmt.bindString(1, title)
stmt.bindString(2, content)
stmt.executeInsert()
```

3. Blog Post List Display (BlogListActivity.kt)

kotlin

```
val cursor = dbReadable.rawQuery("SELECT id, title, content FROM
blog_posts", null)
val posts = mutableListOf<String>()
if (cursor.moveToFirst()) {
    do {
        val title = cursor.getString(1)
        val content = cursor.getString(2)
        posts.add("Title: $title\n$content")
    } while (cursor.moveToNext())
}
```

4. Edit and Delete Functionality (EditBlogPostActivity.kt)

kotlin

```
btnDelete.setOnClickListener {
    dbWritable.execSQL("DELETE FROM blog_posts WHERE id = ?",
arrayOf(postId.toString()))
    finish()
}
```

Appendix B: Tools and Technologies Used

Android Studio - IDE used for Android app development

Kotlin - Programming language used for app logic

SQLite - Local database for storing users and blog posts

XML - Used for designing UI layouts

AVD (Android Virtual Device) - Used for testing the app on virtual devices.

Appendix C: Application Screenshots

Login Module

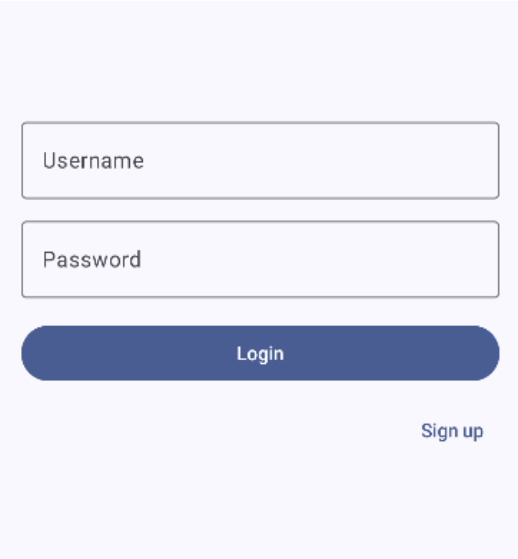

Fig 3.4

Blog Creation Module

Fig 3.5

Blog List Module

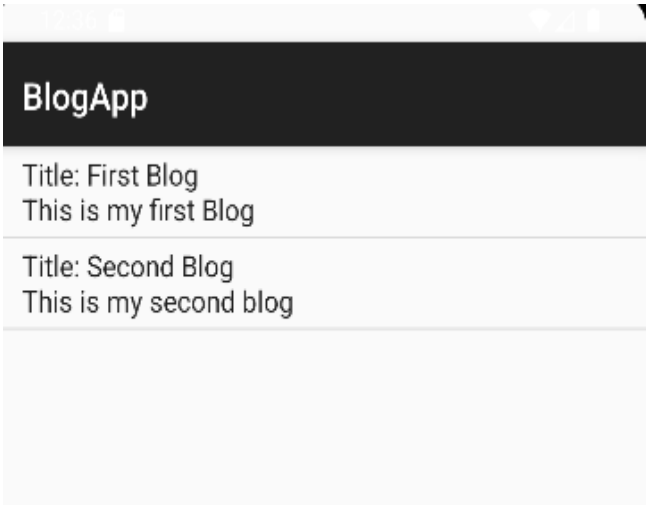

Fig 3.6

Blog Editing Module

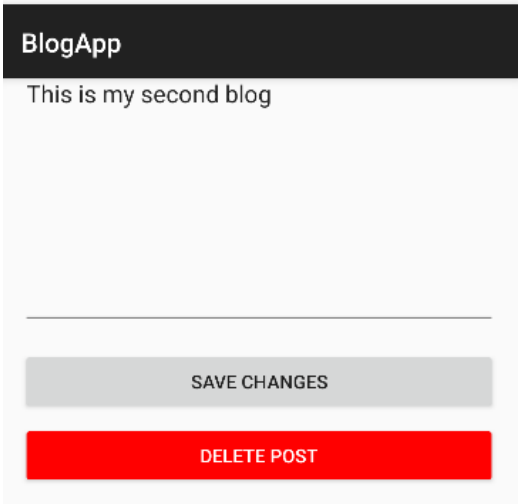

Fig 3.7