DREAMHOUSE – A Real Estate Application in Kotlin

CS19611 – MOBILE APPLICATION DEVELOPMENT LAB

Submitted by

SURANJANA S (2116220701293)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

THANDALAM, CHENNAI - 602105

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Project titled **“DREAMHOUSE – A REAL ESTATE APP”** is the Bonafide work of **“SURANJANA S (2116220701293),** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr. Duraimurugan N,., M.Tech., Ph.D.,
SUPERVISOR
Professor
Department of Computer Science and Engineering,
Rajalakshmi Engineering
College, Chennai-602 105.

Submitted to Project Viva-Voce Examination held on ___________________

Internal Examiner **External Examiner**

ABSTRACT

DreamHouse is a modern, feature-rich real estate mobile application developed using Android Studio and Kotlin. The primary goal of the app is to simplify and digitize the process of renting and selling properties by offering a user-friendly platform for both property owners and potential buyers or tenants. Users can register and log in to the app, browse properties based on location using a real-time search feature, add properties for rent or sale, and bookmark their favourite listings through a wish list system.

One of the key highlights of DreamHouse is its ability to directly connect potential buyers with property owners via integrated call functionality and automated booking notifications. When a user books a property, the owner instantly receives a message with the user's email, streamlining communication and improving trust. The app uses Firebase for backend services including real-time database management, user authentication, and data persistence.

With its intuitive interface, real-time updates, and communication-driven approach, DreamHouse bridges the gap between property seekers and owners, providing a scalable and practical digital solution for today’s real estate market.

ACKNOWLEDGMENT

ACKNOWLEDGMENT Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Mr. S. MEGANATHAN, B.E, F.I.E., our Vice Chairman Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S., and our respected Chairperson Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D., for providing us with the requisite infrastructure and sincere endeavoring 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, Dr. Duraimurugan N , M.Tech., Ph.D., Professor of the Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, Mr. Duraimurugan N Professor Department of Computer Science and Engineering for his useful tips during our review to build our project.

SURANJANA S (2116220701293)

TABLE OF CONTENT

CHAPTER-1

1.INTRODUCTION

In the age of digital transformation, the demand for smart, efficient, and user-centric applications has risen significantly across various sectors, including the real estate industry. Traditionally, property renting and selling have involved manual processes, limited reach, and communication barriers between owners and buyers. Recognizing this gap, **DreamHouse** was developed as a mobile application that aims to digitize and streamline the real estate experience for both property owners and prospective tenants or buyers.

DreamHouse provides an all-in-one platform where users can register, browse, list, book, and manage properties directly from their smartphones. The app eliminates the dependency on middlemen and newspaper listings, making the process more transparent, accessible, and real-time. Users can search for properties using keywords (e.g., location names), bookmark preferred listings through a wish list, and communicate directly with property owners via built-in call functionality. Additionally, when a booking is made, the property owner automatically receives an SMS notification, ensuring timely communication and increased efficiency.

The application is built using **Android Studio** with **Kotlin** as the primary language and leverages **Firebase** for authentication and real-time data storage. DreamHouse not only demonstrates effective use of mobile development technologies but also showcases a practical solution to a real-world problem in the real estate domain.

CHAPTER-2

LITERATURE SURVEY

This section reviews existing systems, mobile applications, and technologies in the real estate domain and highlights the gaps that DreamHouse addresses.

3.1 Existing System

Numerous real estate platforms such as 99acres, MagicBricks, and Housing.com offer users the ability to browse, rent, or purchase properties. These platforms are web-based and often feature complex interfaces, which may overwhelm users looking for a simpler experience. Additionally, these systems often involve middlemen or brokers, reducing transparency in owner-buyer communication.

Mobile apps developed by such platforms usually require high bandwidth and have limited offline functionality. Moreover, they rarely support instant owner notifications upon booking, and typically lack direct call integration between buyer and seller.

3.2 Limitations of Existing Systems

- Lack of direct communication between buyers and property owners.
- Absence of instant notifications to property owners upon booking confirmation.
- Overdependence on third-party agents or brokers.
- Complex user interfaces that may not be accessible for all users.
- High dependency on continuous internet access for basic operations.

3.3 Proposed System – DreamHouse

DreamHouse overcomes these limitations by offering:

- A clean and intuitive mobile-first interface.

- Real-time search and filtering of properties by location.
- Integrated wishlist to mark favorite properties.
- Direct phone call functionality to connect with property owners.
- Automatic SMS notifications to owners when a house is booked.
- Use of Firebase backend to ensure secure, scalable, and real-time data management.

DreamHouse provides a more streamlined, interactive, and owner-centric experience, addressing the key challenges faced in traditional and even some modern real estate applications.

CHAPTER-3

METHODOLOGY

The methodology adopted for the development of the **DreamHouse** application is based on a modular, iterative software development lifecycle, ensuring each feature is built, tested, and integrated effectively. The app is designed with scalability, user experience, and real-time responsiveness in mind. It uses Android Studio with Kotlin as the primary development language and Firebase as the backend service.

4.1 System Architecture

The system is divided into two main components:

- **Frontend (Mobile App)**:
  Developed in Kotlin using Android Studio. It handles all user interactions, screen navigations, form submissions, and real-time data display.

- **Backend (Firebase)**:
  Includes Firebase Authentication for login/signup, Firebase Realtime Database for storing house data and bookings, and Firebase Storage (optional) for property images.

4.2 Development Methodology

The project was developed following these structured steps:

1. **Requirement Analysis**:
   Identifying features such as user registration/login, house posting, search, wish list, booking, and call/SMS integration.

2. **Design**:
   - UI mockups created using Android layouts (XML).
   - Data models designed (House, User, Booking).
   - Navigation flow planned.

3. **Implementation**:
   - Login and registration screens using Firebase Authentication.
   - Property listing screen with add/view capabilities using Firebase Realtime Database.
   - Wishlist and bookings stored under user-specific Firebase nodes.
   - Call button implemented using Intent.ACTION_DIAL.
   - SMS functionality integrated to notify owners on booking via SMS Intent.

4. **Testing**:
   - Unit testing for input validation.
   - Manual testing for navigation, SMS/call functionality, Firebase data syncing, and search filtering.

5. **Deployment**:
   - APK generated for mobile installation.
   - Debug and release builds created using Android Studio's build system.

4.3 Tools and Technologies Used

Tool/Platform	**Purpose**
Android Studio	IDE for development
Kotlin	Primary programming language
Firebase Auth	User authentication
Firebase Realtime DB	Data storage (houses, bookings, wishlist)
Firebase Storage	(Optional) Image storage
XML	UI layout design
Google Intents	For calling and SMS functionality

Backend Infrastructure

The backend infrastructure of **DreamHouse** is designed to provide real-time responsiveness, scalability, and secure data handling. The system relies heavily on **Firebase**, a cloud-based Backend-as-a-Service (BaaS) provided by Google, which integrates seamlessly with Android applications and supports core backend functionalities such as authentication, database management, and cloud messaging.

5.1 Firebase Authentication

Firebase Authentication is used to manage user login and registration. It supports email/password-based login for secure and reliable access control. All authenticated users are uniquely identified via their Firebase UID, which is then used to personalize and associate data like bookings and wish lists.

5.2 Firebase Realtime Database

The Realtime Database is a NoSQL cloud database that stores data in JSON format. It enables real-time syncing between users and the backend, ensuring that all users view updated data instantly. Key nodes in the database include:

- /users: Stores basic user info (optional).
- /houses: Contains all property listings including title, price, address, type, and owner UID & phone number.
- /bookings: Contains booking records, linking user ID and house ID.
- /wishlist: Stores favourite houses per user.

5.3 Firebase Storage (Optional)

If the user wishes to upload images of the house, Firebase Storage can be integrated to handle image uploads. Images are stored in a secure, scalable manner and are accessed via download URLs which are linked in the Realtime Database.

5.4 Real-Time Capabilities

- The Realtime Database listener is implemented in the frontend to **dynamically update the house listings** as new entries are added or removed.
- This eliminates the need for manual refresh and provides a seamless user experience.

5.5 Data Flow

1. When a user **adds a new house**, the data is pushed to /houses in Firebase.
2. When a house is **booked**, a record is added to /bookings and the owner's phone number is used to trigger an **SMS intent** with the booking message.
3. When a user **adds a house to wishlist**, it is stored under /wishlist/{userId}/{houseId}.

5.6 Security

Firebase Authentication ensures that each user's data (wishlist, bookings) is isolated. Rules can be configured in Firebase Realtime Database to restrict read/write access based on authenticated user IDs, ensuring data integrity and privacy.

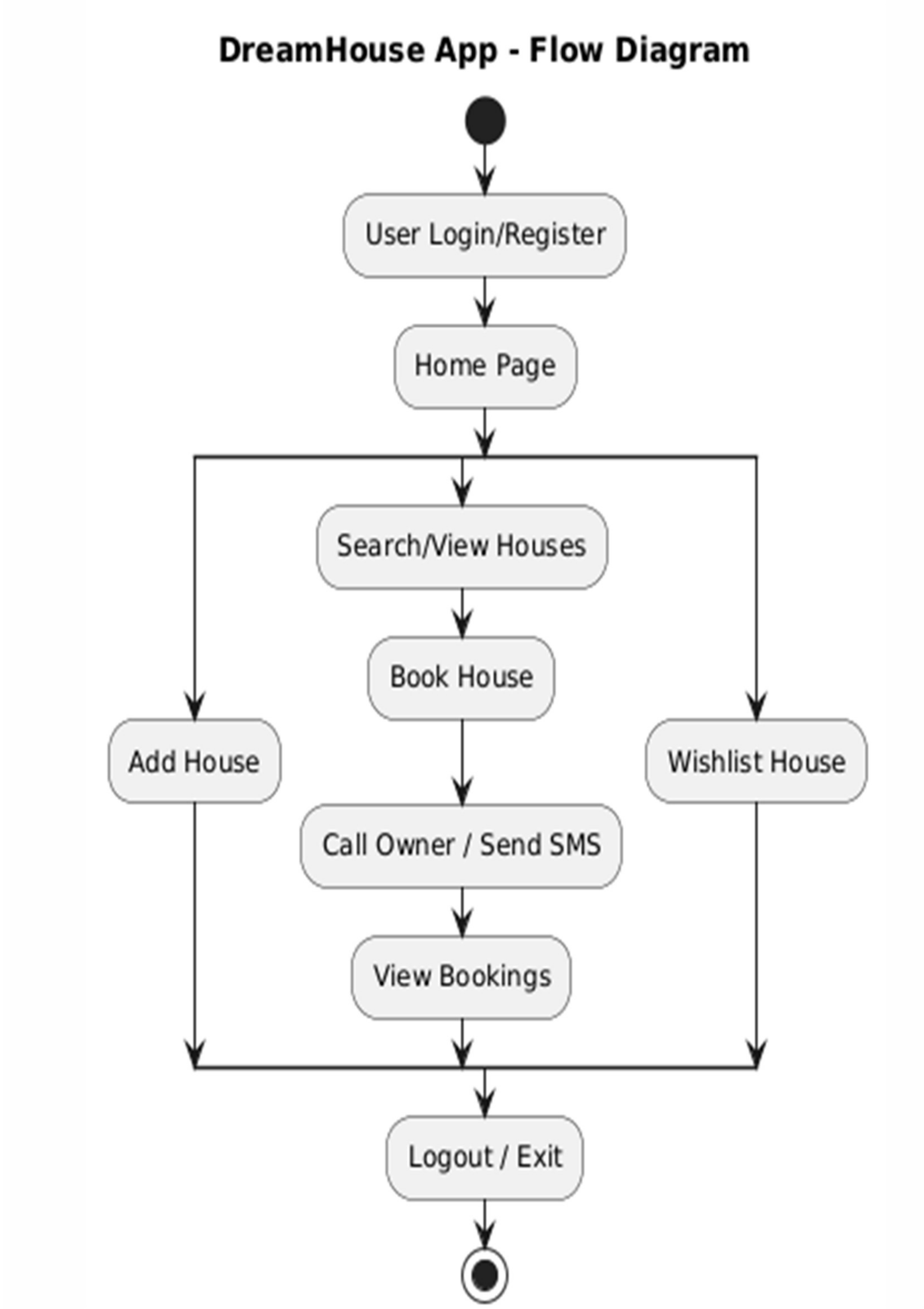
OBJECTIVES

The primary goal of the DreamHouse application is to create a user-friendly, scalable, and efficient mobile platform for renting and selling properties with enhanced communication between property owners and potential tenants or buyers. The specific objectives of the project are as follows:

1. To provide a centralized real estate mobile platform
   Enable users to browse, add, book, and manage properties from a single, intuitive mobile application.
2. To ensure real-time property updates and user synchronization
   Use Firebase Realtime Database to deliver up-to-date listings and seamless user experience across devices.
3. To eliminate the dependency on intermediaries or brokers
   Facilitate direct communication between property owners and customers via call functionality.
4. To introduce an automated booking alert system
   Notify property owners instantly when their house is booked via SMS, enhancing communication speed and engagement.
5. To simplify the search and decision-making process
   Allow users to search properties based on location keywords and manage preferred listings using a wishlist.
6. To ensure secure access through user authentication
   Use Firebase Authentication for secure login, personalized data storage, and preventing unauthorized access.
7. To develop a modular, extendable app architecture
   Build a clean, maintainable codebase that supports future scalability and feature additions.

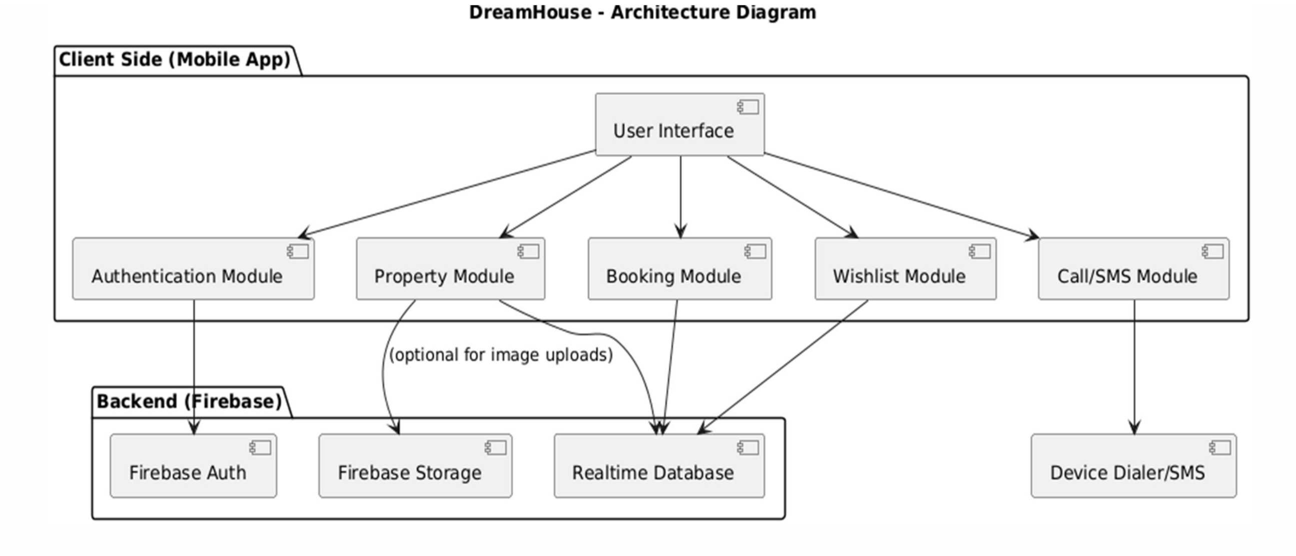
CHAPTER 4

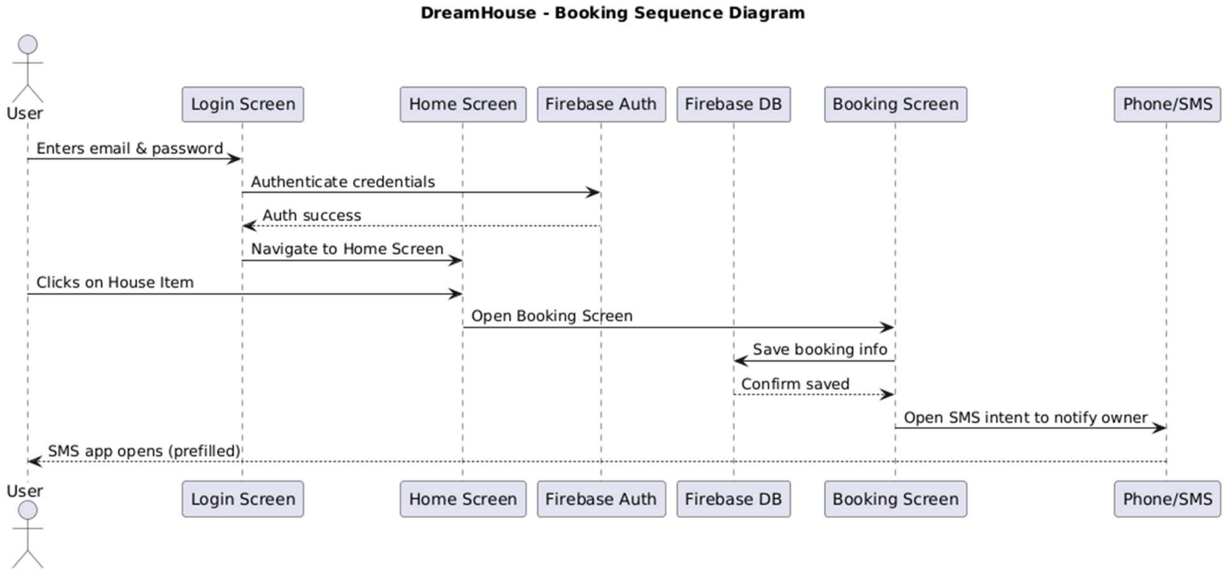
FLOW DIAGRAM



CHAPTER-5

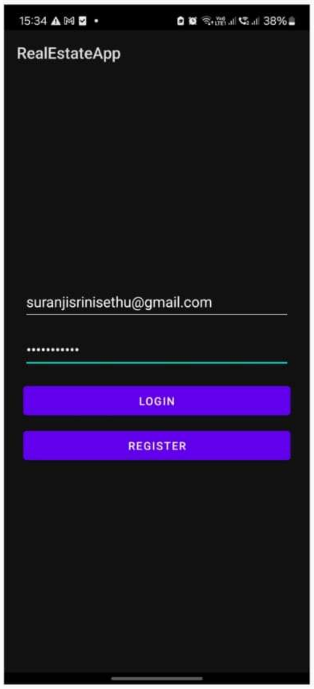
ARCHITECTURE DIAGRAM





CHAPTER-6

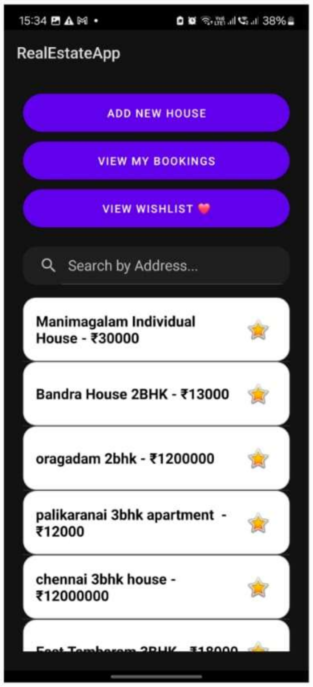
OUTPUT SCREENSHOT


15:34 38%

RealEstateApp

suranjisrinisethu@gmail.com

•••••••••••


15:34 38%

RealEstateApp

Manimagalam Individual House - ₹30000 ⭐

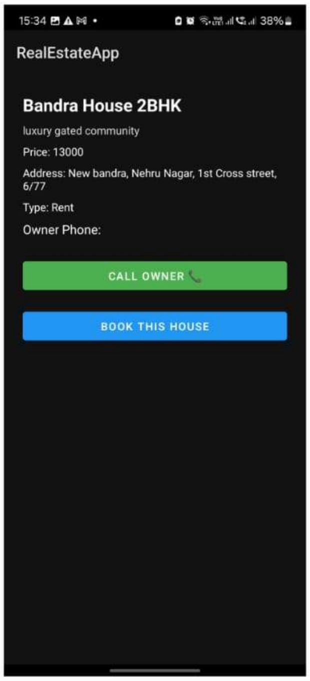
Bandra House 2BHK - ₹13000 ⭐

oragadam 2bhk - ₹1200000 ⭐

palikaranai 3bhk apartment - ₹12000 ⭐

chennai 3bhk house - ₹12000000 ⭐

East Tambaram 2BHK - ₹18000 ⭐


15:34 38%

RealEstateApp

Bandra House 2BHK

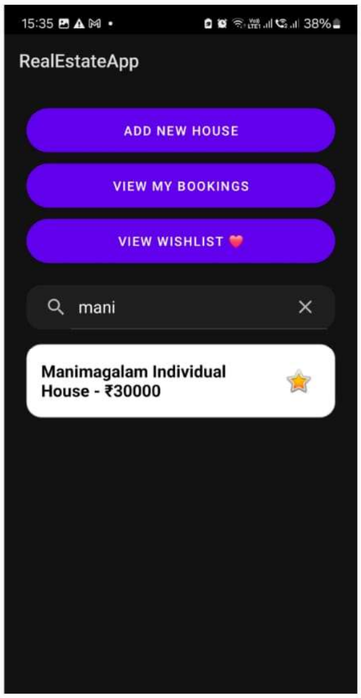
luxury gated community

Price: 13000

Address: New bandra, Nehru Nagar, 1st Cross street, 6/77

Type: Rent

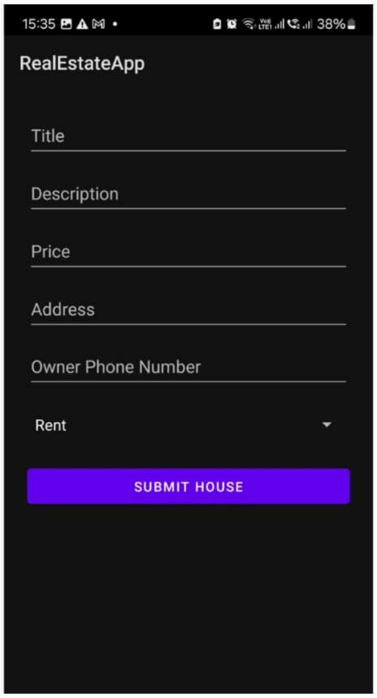
Owner Phone:


15:35 38%

RealEstateApp

mani

Manimagalam Individual House - ₹30000 ⭐


15:35 38%

RealEstateApp

Title

Description

Price

Address

Owner Phone Number

Rent

CHAPTER-7

RESULTS AND DISCUSSION

The DreamHouse mobile application was successfully developed and tested, meeting all the core objectives of the project. The app provides a seamless experience for users to list, search, book, and interact with property listings, all through a clean and intuitive mobile interface. Integration with Firebase ensures real-time updates and secure data management, while additional features like wishlist, call owner functionality, and SMS booking alerts enhance user engagement and operational transparency.

9.1 Results Achieved

- User Authentication: Secure login and registration using Firebase Authentication was implemented successfully. Only authenticated users can post or book properties.
- Property Listing: Owners can list properties with essential details such as title, description, address, type, price, and phone number.
- Search and Filter: Users can dynamically search for properties based on location (e.g., “Chennai”) using real-time filtering.
- Wishlist Feature: Users can add houses to a wish list for future reference, with all data saved in user-specific Firebase nodes.
- Booking System: Users can book a house, and booking details are stored and displayed in the "My Bookings" section.
- Owner Notification via SMS: Upon booking, the app launches an SMS intent with a pre-filled message to the property owner's phone number, enabling instant notification.
- Direct Communication: Users can call the owner directly from the booking screen using Intent.ACTION_DIAL.

- User Interface: The UI was upgraded with a dark theme, rounded buttons, and card-based house display, providing a professional, modern look.

9.2 Testing and Validation

Manual testing was conducted on various Android devices to ensure feature functionality and performance consistency. Key focus areas during testing included:

- Login/logout flow
- Add/view property listing
- Booking process and SMS intent
- Wishlist addition/removal
- Firebase database synchronization
- Search bar behavior for partial and full queries

The app handled all test cases without crashing, and user data persisted correctly between sessions due to Firebase's real-time data syncing.

9.3 Discussion

The project effectively addresses several limitations found in traditional and digital real estate systems. The use of Firebase as a BaaS significantly reduced backend complexity while maintaining data integrity and security. Although automated SMS sending was considered, due to Android limitations, a user-assisted SMS intent was implemented to ensure compatibility and compliance. Future iterations of the app could benefit from features like image uploads, push notifications, chat integration, and payment gateway support.

CONCLUSION & FUTURE ENHANCEMENTS

Conclusion

The DreamHouse mobile application successfully fulfils its aim of providing a digital, efficient, and user-friendly platform for managing real estate rentals and sales. By leveraging Android Studio, Kotlin, and Firebase, the application delivers real-time functionality, secure authentication, and seamless data synchronization between users and the backend. The integration of essential features such as user authentication, property listing, real-time search, wish list management, direct calling, and SMS notifications creates a comprehensive solution tailored to the needs of both property owners and seekers.

Throughout the development and testing phases, the app demonstrated stability, accuracy, and responsiveness across all its modules. Users were able to interact with the platform effortlessly, from browsing properties to communicating directly with owners—eliminating the traditional friction associated with middlemen or delayed responses.

The project not only showcases the implementation of modern mobile development practices but also highlights the practical application of technology in solving real-world problems in the real estate domain. The modular design and Firebase-based infrastructure ensure that DreamHouse is scalable and adaptable for future enhancements.

Future Enhancements

While **DreamHouse** effectively meets its core objectives and delivers a smooth user experience, there are several enhancements that can be considered in future iterations to increase the app's usability, scalability, and feature richness:

1. **Image Upload and Gallery View**
   Allow owners to upload property images using Firebase Storage and display them in a swipe able gallery for better visualization by users.

2. **In-App Chat Functionality**
   Introduce a real-time chat system between users and owners to facilitate secure and direct communication within the app.

3. **Push Notifications**
   Implement Firebase Cloud Messaging (FCM) to notify users about booking confirmations, new property listings, and wish list updates.

4. **Payment Integration**
   Provide secure online payment options for users to make advance payments or security deposits using Razorpay, Stripe, or Google Pay APIs.

5. **Admin Dashboard**
   Add an admin interface to manage user accounts, property listings, and reported content to ensure platform safety and integrity.

6. **Map Integration**
   Integrate Google Maps API to show property locations and enable location-based search and navigation.

7. **Feedback and Rating System**
   Let users rate properties and owners to build trust and improve the quality of listings.

8. **Dark/Light Theme Toggle**
   Include a toggle to switch between light and dark UI themes based on user preference.

9. **Analytics Dashboard for Owners**
   Display insights like how many users viewed, wished, or booked a particular property to help owners make informed decisions.

10. **Multi-Language Support**
    Add localization features to support multiple regional languages for broader accessibility.

These enhancements would not only improve the functionality and reach of DreamHouse but also align it more closely with modern commercial real estate platforms.

CHAPTER-9

REFERENCES

1. Google Firebase Documentation – https://firebase.google.com/docs
2. Android Developer Documentation – https://developer.android.com
3. Kotlin Language Documentation – https://kotlinlang.org/docs/home.html
4. Stack Overflow Developer Community – https://stackoverflow.com
5. GeeksforGeeks Android Tutorials – https://www.geeksforgeeks.org/androidtutorial/
6. W3Schools XML Reference – https://www.w3schools.com/xml/
7. PlantUML Documentation – https://plantuml.com/
8. Google Maps API (for future enhancement reference) – https://developers.google.com/maps