RoomEase

Version 1.

May 5, 2025

Suelto, JanMark
Abejo, Azriel Jay
Bandivas, Earl Efraim

Contents

Document Overview

Area	Covered	Out-of-Scope
Functional	User registration, room search, map integration, booking system, payment processing.	Offline mode support, advanced analytics
Technical	Frontend (Java), backend (Java), APIs (), database structure.	Custom API server development.
Design	UI wireframes, navigation flow, color scheme, typography.	Detailed branding guidelines.
Testing	Unit tests, UI tests, manual test cases for core features.	
Deployment		

Audience

Audience	Purpose
Developers	To understand code structure, APIs, and setup instructions
Designers	To reference UI/UX flows and design assets.
Testers	To execute test cases and report bugs.
Stakeholders	To review features, timelines, and high-level architecture.

Project Overview

Executive Summary

App Name: RoomEase

Purpose: A digital solution to modernize boarding house operations by providing real-time room availability, seamless bookings, and location-based discovery for tenants and property owners.

Objectives

Business Goals	Technical Goals
Increase boarding house occupancy rates	Build a cross-platform app
Reduce manual booking inquiries	Implement real-time sync for room availability.

High-Level Features

- Interactive Map

  View boarding houses as pins with color-coded vacancies

  Tap pins to see details: photos, pricing, amenities.
- Real-Time Room Availability

  Owners mark rooms as occupied/vacant via admin panel.
- Booking & Payments

  Reserve rooms in-app with secure payments.
- User Roles

  Owners/Admin: Add properties, update room status, view bookings.

  Tenants: Browse, book, pay, manage reservations.

Problem Statement

User Needs

The app addresses critical gaps in the boarding house rental market:

For Tenants (Students/Professionals):

- Problem: Difficulty finding real-time vacancy updates, leading to wasted trips.
- Need: Instant visibility into available rooms, pricing, and location.

For Owners/Managers:

- Problem: Manual booking tracking (paper/Excel) causes overbookings.
- Need: Automated occupancy management and payment collection.

Market Analysis

Target Audience

Segment	Location	Key Behavior
University Students	Malaybalay City	Prefers affordable, near-campus housing.
Young Professionals	Urban Areas	Seeks convenience and work related purposes.

Competitive Landscape:

Competitor	Weakness	Our Edge
Facebook Groups	No real-time updates, scams.	Verified listings + live vacancy status.

Functional Specifications

Feature List

1. Room/Resource Booking

   Description:

   - Allows users to browse and book available rooms (e.g., classrooms, offices) or resources (e.g., projectors, laptops) in real-time.

     User Interaction Flow

   - Browse: User selects location (e.g., "Block A") → Filters by date/time → Views available rooms/resources.
   - Book: Clicks "Reserve" → Confirms details → Receives confirmation (email/app notification).

   Use Cases:

   - Primary: Student books a study room for 2 hours.
   - Alternate: Admin overrides a booking for emergency use.

2. Real-Time Availability Calendar

   Description:

   - Displays room/resource status (available, booked, maintenance) in a color-coded calendar view.

   User Interaction Flow:

   - User opens calendar → Selects date

3. Admin Dashboard

   Description:

   - Let's admins approve/deny bookings and add new resources

   User Interaction Flow:

   - Admin logs in → Views pending requests → Approves/rejects with notes.
   - Adds new room ("Room 3") with status.

   Use Case:

   - Primary: Admin allocates a room for a last-minute meeting.

4. Push Notifications

   Description:

   - Sends alerts for booking confirmations, reminders, and cancellations.

   User Interaction Flow:

   - User books → Instantly gets "Confirmed!" notification.
   - Receives reminder 1 hour before booking starts.

   Use Cases:

   - Primary: User reschedules after getting a "Conflict" alert.
   - Alternate: System notifies admin of no-shows.

5. Location Awareness (GPS/Maps)

   Description:

   - Shows room locations of the boarding house.

   User Interaction Flow:

   - User taps "View on Map" → Sees building pins → Selects pin for directions.

   Use Cases:

   - Primary: New student finds the booked seminar room.
   - Alternate: Redirects to accessible routes for wheelchair users.

User Stories & Requirements

1. User Stories (End-User Perspective)

   As a Student, I want to:

   - Browse available rooms, so I can find a study space quickly.

     Acceptance Criteria:
     Rooms show real-time availability.
     Filters for date, duration, and capacity work.

   - Book a room in one tap.

     Acceptance Criteria:
     Confirmation appears.
     Calendar syncs with my Google Calendar.

As an Admin, I want to:

- Update the availability of the room

  Acceptance Criteria:
  It should be real time.

- Add a room to occupy.

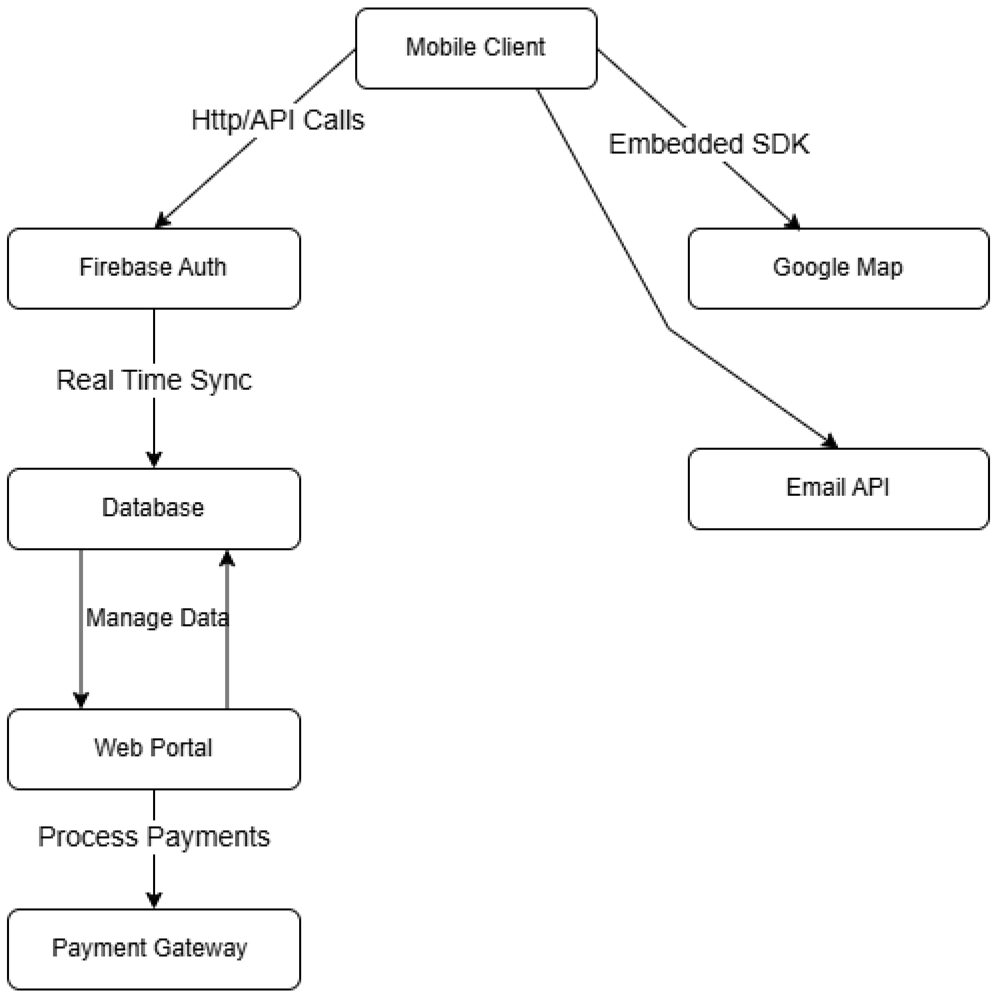
  Acceptance Criteria:
  It should update as quickly as possible.

Non-Functional Requirements

Category	Requirement	Metric
Performance	Load room listings in <5.5 seconds.	90% of requests under 4s.
Security	Encrypt all user data	Security Compliance Standard
Reliability	99.9% uptime outside maintenance windows.	Monitored via Firebase
Scalability	Support 500+ users	Cloud Firestore auto-scaling.
Usability	Achieve 85%+ task success rate in UX tests.	Measured via Maze.design tests.

Technical Specifications

Overview Diagram



Key Components:

- **Mobile Client**: Android app (Kotlin/Java) for users/admins.
- **Backend**: Java (Firebase Auth).
- **APIs:** Google Maps API (location pins).

  Email Messaging (notifications).
- **Database:** Firebase for real-time sync.

Design Patterns

Pattern	Role	Implementation
MVVM	Separates UI logic from business logic.	ViewModel fetches room data from Firebase, exposes LiveData to UI.
Repository	Centralizes data operations (e.g., cache/API calls).	RoomRepository handles Firestore or SQLite sync.
Singleton	Manages global resources (e.g., Firebase instance).	FirebaseAuth

Platform-Specific Considerations

iOS & Android Guidelines

Aspect	Android (Material 3)	iOS (Human Interface Guidelines)
Navigation	Bottom nav bar or drawer menu.	Tab bar at bottom (max 5 items).
Back Button	System back button (or gesture).	No physical button—swipe from left edge.
Typography	Roboto (default), variable font support.	San Francisco (dynamic type scaling).
Icons	Material Symbols (filled/outlined).	SF Symbols (Apple's unified icon set).
UI Components	Floating Action Button (FAB), Cards.	Navigation bars, action sheets.

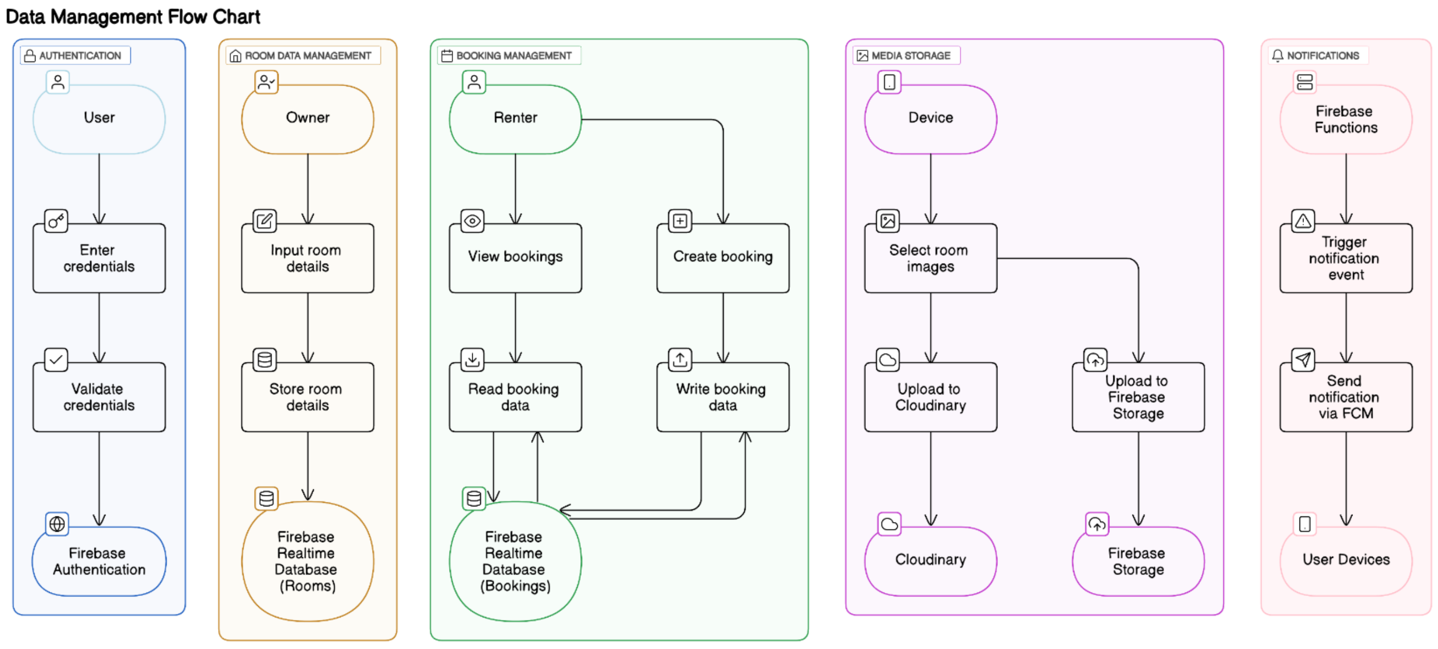
Key UX Differences:

- Android: More customization, heavier use of shadows/floating elements.
- iOS: Minimalist, prefers translucent blurs and tighter padding.

Hardware & OS Compatibility

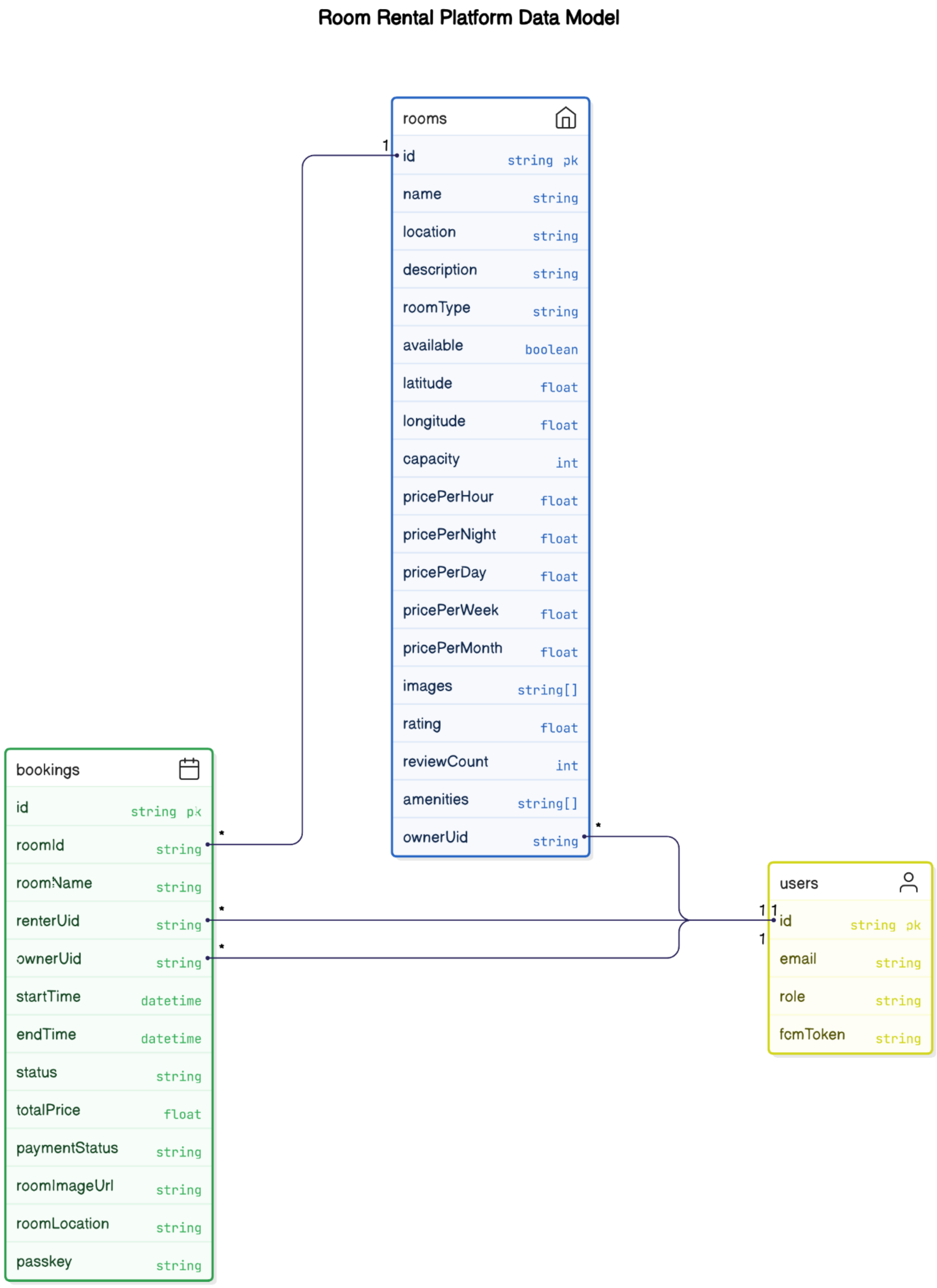
Requirement	Android	iOS
Min OS Version	Android 8.0 (API 26)	iOS 15+
Screen Support	Responsive layouts (phones, tablets, foldables).	iPhone/iPad (no foldable support).
Permissions	Request at runtime (e.g., location, storage).	Privacy descriptions in Info.plist.
Device Limitations	Fragmentation (test on 3+ OEMs).	Consistent hardware (test on 2-3 models).

Data Management



Database Schemas

Firebase Realtime Database Schema



API Endpoints (Firebase REST)

Endpoint	Method	Description
/rooms.json	GET	Retrieve all rooms
/rooms/{roomId}.json	GET	Retrieve specific room
/rooms.json	POST	Create new room
/rooms/{roomId}.json	PATCH	Update room details
/bookings.json	GET	Retrieve all bookings
/bookings.json	POST	Create new booking
/bookings/{bookingId}.json	PATCH	Update booking status

Authentication & Authorization

- **Email/Password** via Firebase Authentication.
- **Role-based Access**:
  - Admin: full access
  - Owner: manage their listings
  - Renter: browse and book rooms
- **JWT (Firebase Token)**: Session token used for API calls.
- **Security Rules**:
  - bookings only accessible to owners or renters of that booking.
  - rooms editable only by the creator.

Data Encryption

- **In-Transit**: All API calls use HTTPS.
- **At-Rest**: Data stored in Firebase is encrypted using AES-256.
- **Sensitive Data**:
  - Room passkeys are hidden until bookings are approved.

Compliance Requirements

- **GDPR**:
  - Explicit consent for location and data usage.
  - Users can request account and data deletion.
- **Data Policies**:
  - Retention aligned with local guidelines.
  - All users agree to Terms of Use and Privacy Policy.

User Data Handling

Data Type	Collected For	Stored In
Email, Role	Authentication	Firebase Auth
Room Info	Display, Search, Booking	Firebase Realtime DB
Bookings	History and Notifications	Firebase Realtime DB
FCM Token	Push notifications	Firebase DB
Images	Room listings	Cloudinary / Firebase

Third-Party Integration

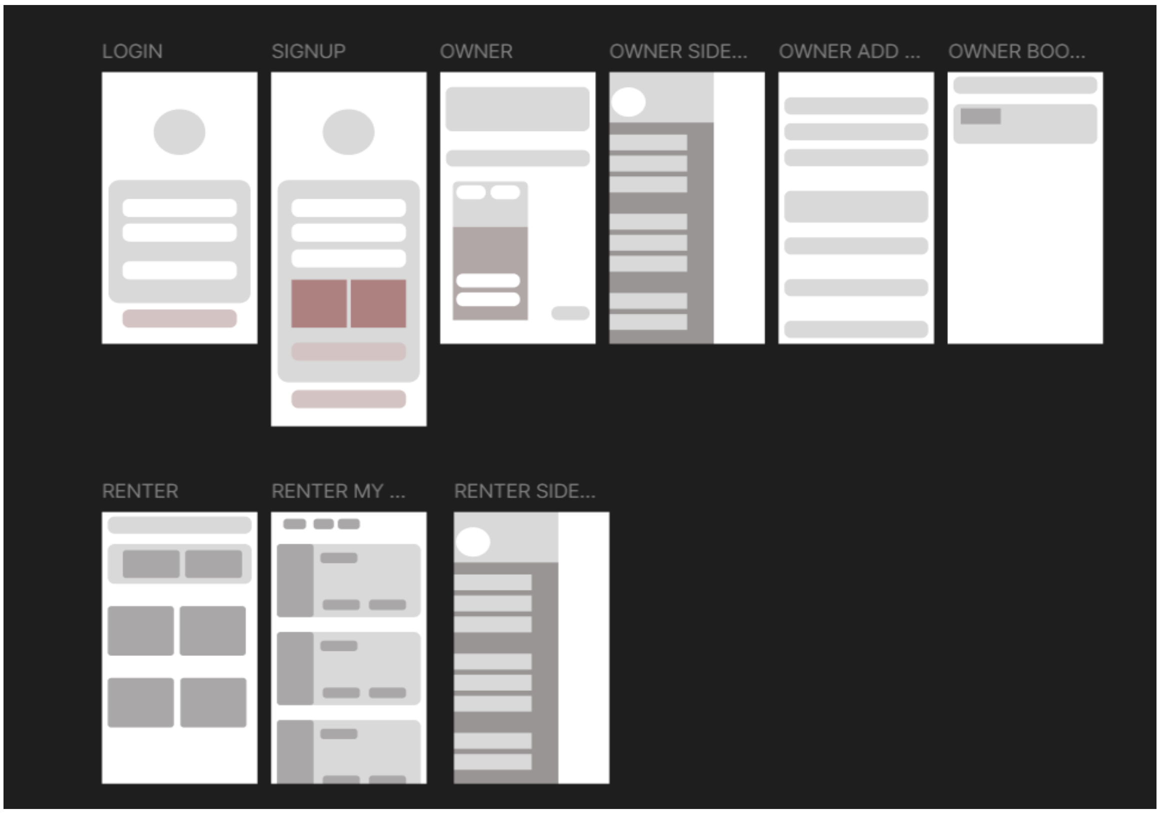
Tool / SDK	Purpose
Firebase SDK	Auth, Realtime DB, Cloud Functions
Firebase Storage	Image or file storage
Google Maps SDK	Display room locations on map
Cloudinary SDK	Upload and transform images
AndroidX Components	UI components, navigation, WorkManager

Integration Details

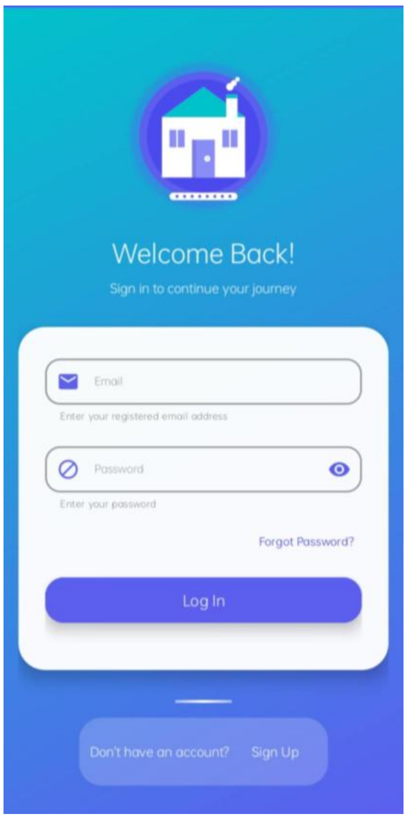
- **Firebase Initialization**: Set in RoomEaseApp.java
- **Cloudinary**: Configured via API key and upload presets
- **Google Maps**: Embedded on Room Details and Search screen
- **Cloud Functions**: Auto-triggered email and notification alerts when:
  - A booking is created or cancelled
  - A booking is approved

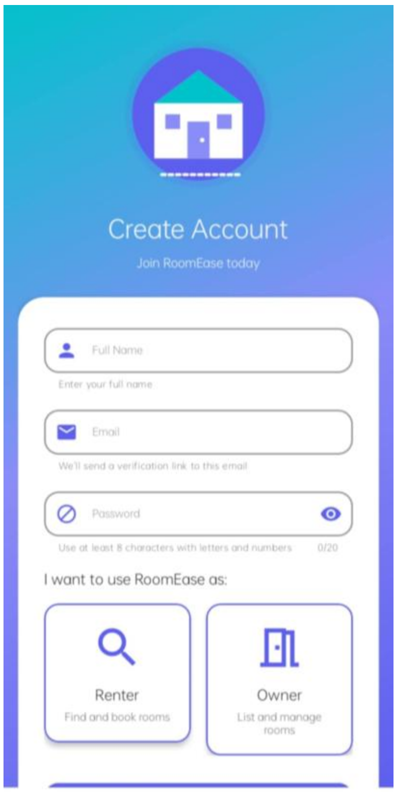
UI/UX Design Specifications

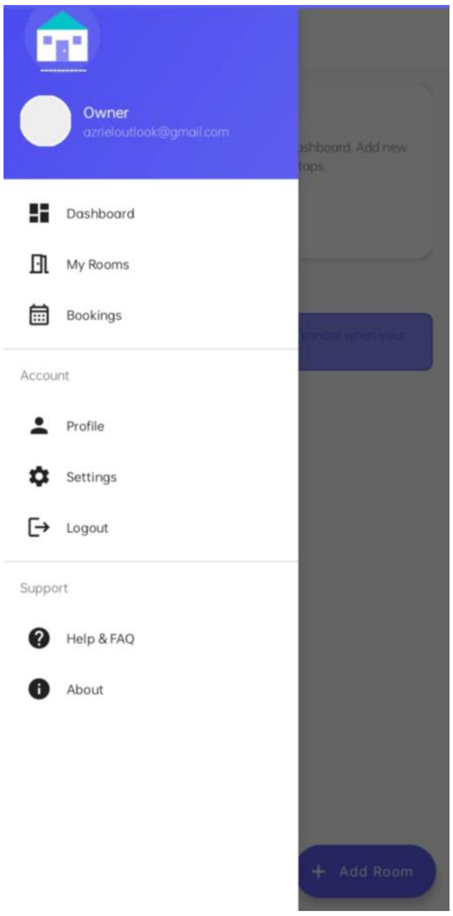
Wireframes & Mockups

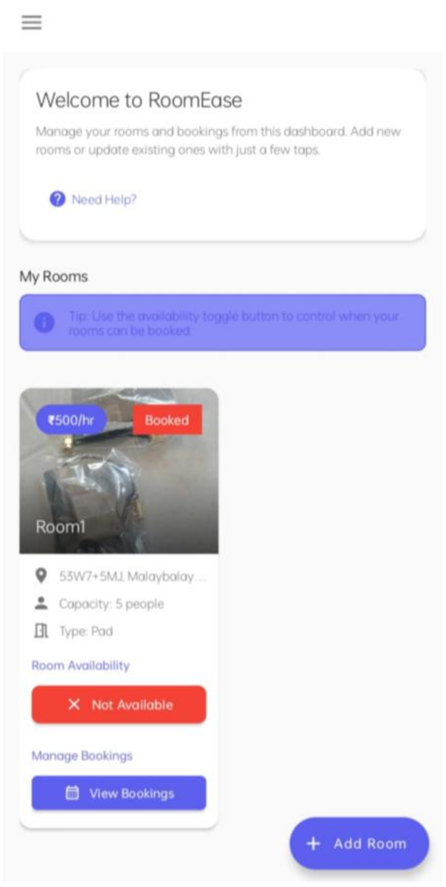


High-Fidelity Designs



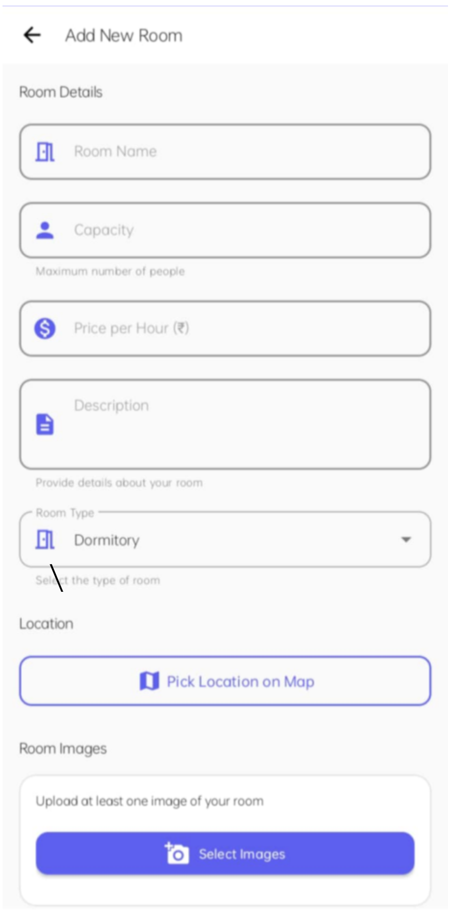





Welcome to RoomEase

Manage your rooms and bookings from this dashboard. Add new rooms or update existing ones with just a few taps.

Tip: Use the availability toggle button to control when your rooms can be booked.



Add New Room

Room Details

Room Name

Capacity

Maximum number of people

Price per Hour (₹)

Description

Provide details about your room

Room Type

Dormitory

Select the type of room

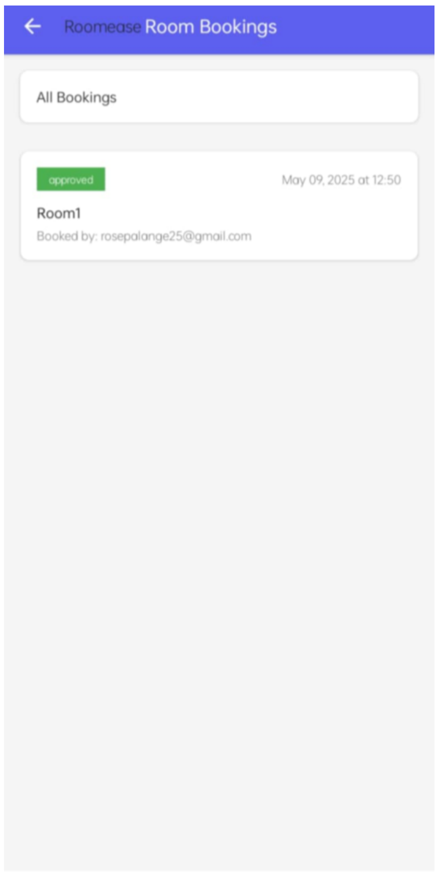
Location

Pick Location on Map

Room Images

Upload at least one image of your room

Select Images



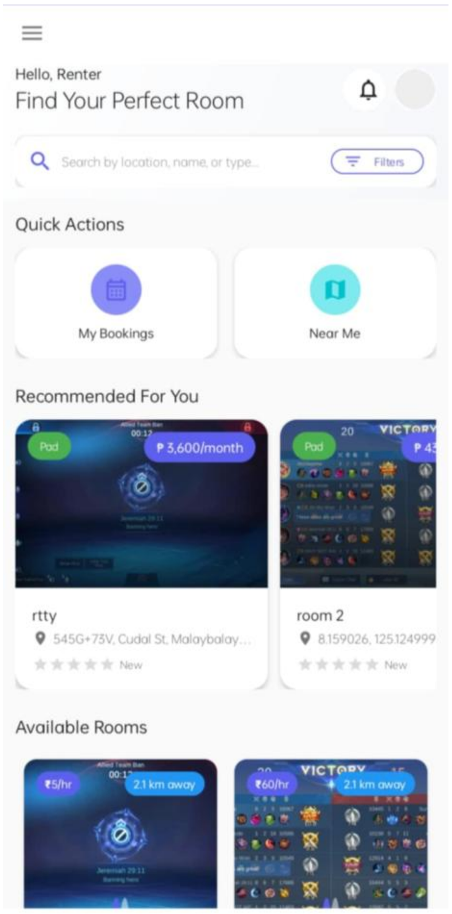
Roomease Room Bookings

All Bookings

approved May 09, 2025 at 12:50

Room1

Booked by: rosepalange25@gmail.com



Hello, Renter

Find Your Perfect Room

Search by location, name, or type... Filters

Quick Actions

My Bookings

Near Me

Recommended For You

Pod ₱ 3,600/month

rtty

545G+73V, Cudal St, Malaybalay...

New

Pod ₱ 4

room 2

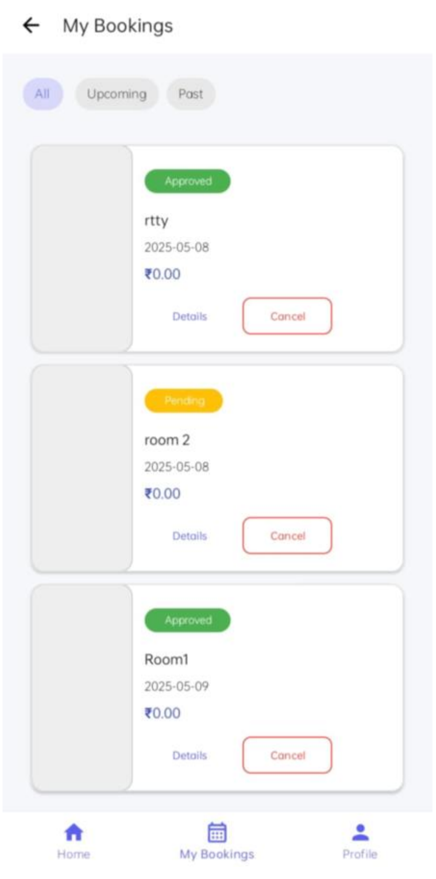
8.159026, 125.124999

New

Available Rooms

₹5/hr 2.1 km away

₹60/hr 2.1 km away



My Bookings

All Upcoming Past

Approved

rtty

2025-05-08

₹0.00

Details Cancel

Pending

room 2

2025-05-08

₹0.00

Details Cancel

Approved

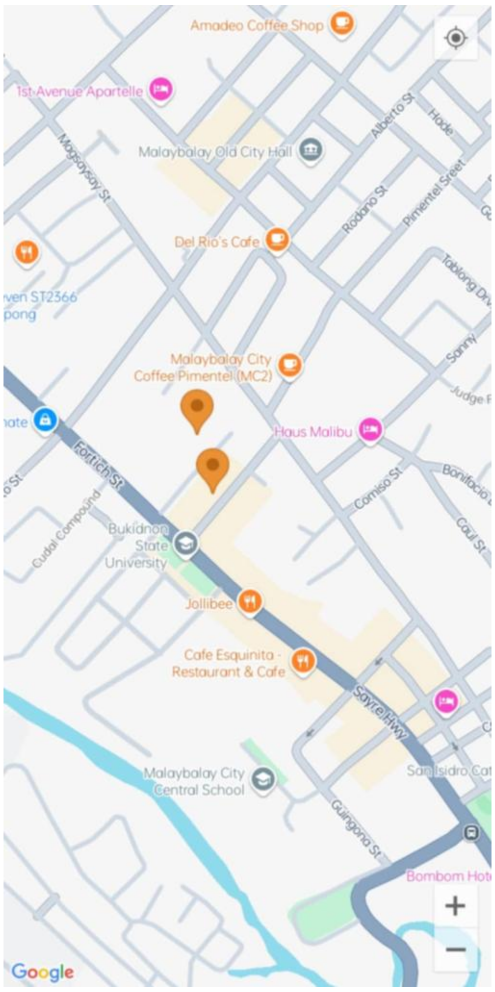
Room1

2025-05-09

₹0.00

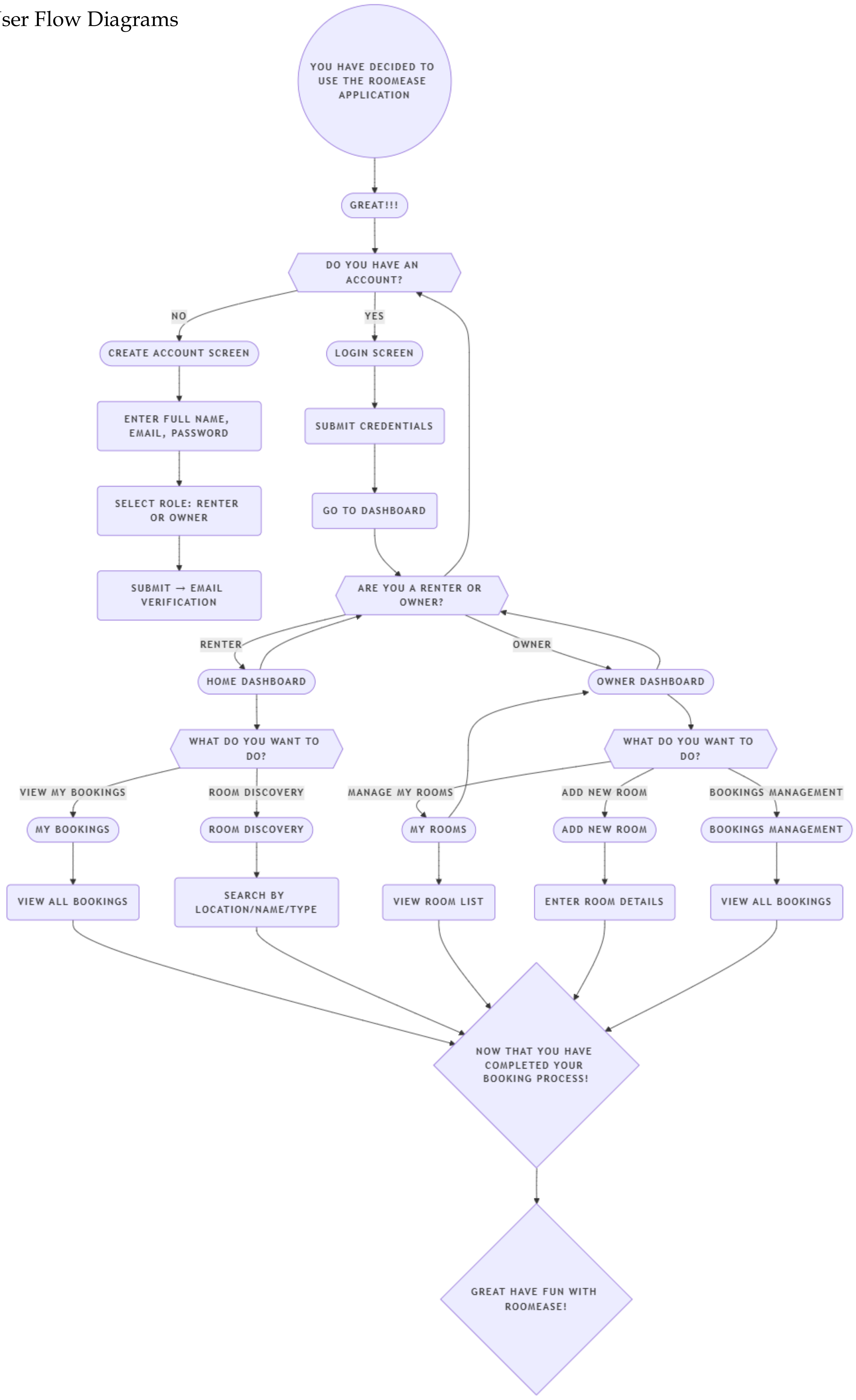
Details Cancel

Home My Bookings Profile


Amadeo Coffee Shop
1st Avenue Apartelle
Malaybalay Old City Hall
Magsaysay St
Alberto St
Rodano St
Pimentel Street
Del Rio's Cafe
Tablong Dr
Malaybalay City Coffee Pimentel (MC2)
Sonny
Judge
Haus Malibu
Fortich St
Comiso St
Bonifacio
Cudal Compound
Bukidnon State University
Caul St
Jollibee
Cafe Esquinita - Restaurant & Cafe
Sayre Hwy
San Isidro Cat
Malaybalay City Central School
Guingona St
Bombom Hot
Google

Navigation & Flow

User Flow Diagrams



Accessibility Guidelines

1. Text & Readability

Implemented in the App:

Consistent font sizes with good legibility: Large headers ("Find Your Perfect Room", "Welcome to RoomEase") and smaller, readable labels for buttons like "My Bookings" and "Add Room".

SP units likely used (based on Android styling conventions) — ensures Dynamic Type scaling works.

Adequate white space and padding prevent text from feeling cramped.

Improvements:

Ensure text scales well up to 200% by avoiding fixed-height containers (especially on card views).

2. Color & Contrast

Implemented in the App:

Primary buttons (e.g., "View Bookings", "Add Room") have sufficient contrast against the background (blue on white).

Status tags like "Booked" (in red) and price labels (in blue) are clearly distinguishable.

Background-to-text contrast in cards and headers is strong.

Improvements:

Use a contrast checker to confirm all text (especially gray subtext like room addresses and ratings) meets the WCAG AA standard of 4.5:1.

Ensure that color-coded tags (e.g., “Pad” in green) are supplemented with icon/text labels for users with color blindness.

3. Screen Reader (TalkBack) Support

Visible Intent:

Interactive elements such as the navigation bar icons, room cards, and booking buttons are distinguishable and likely have contentDescription set.

Implementation Reminders:

All buttons (e.g., Filters, Add Room, View Bookings) should have meaningful content descriptions.

Use LiveRegion to announce booking updates, availability changes, or errors dynamically.

Decorative icons (e.g., the house icon on the sign-up page) should be ignored by screen readers.

4. Touch Targets & Interaction

Implemented in the App:

Action buttons (e.g., "Add Room", "View Bookings", toggle availability) are large and centered, easily tappable.

Cards for rooms, quick actions, and booking details have enough spacing and padding.

Enhancements:

Double-check that all touch targets are ≥ 48x48 dp.

For icons like star ratings, ensure at least 8dp padding to prevent accidental touches.

5. Navigation & Gestures

Present in the App:

Bottom navigation bar with clear icons and spacing.

FAB (Floating Action Button) for "Add Room" indicates its interactive purpose through color and position.

Visual hierarchy supports intuitive gesture navigation (e.g., scrollable room lists).

Provide manual buttons for any swipe-based actions (like "Cancel" or "Delete").

6. Testing & Validation

Your team's design already considers:

Clean layouts, scalable UI, and proper spacing — a good base for accessibility compliance.

Final Steps:

Run Android Accessibility Scanner on real devices.

Test with TalkBack enabled to check label coverage and focus order.

Simulate grayscale or colorblind mode to verify information is still understandable.

VIII. Deployment & Maintenance

Deployment Plan

Build Generation

Tool: Android Studio

Outputs:

Signed APK / AAB for production distribution.

Debug builds for internal testing.

Security & Optimization:

Enable ProGuard/R8 for code shrinking and obfuscation.

Testing

Internal QA:

Tool: Firebase Test Lab (automated & manual test cases)

Test Scenarios:

Renter: Book a room, view booking confirmation.

Owner: Add/manage room listings, toggle availability.

Authentication: Validate signup, login, and role selection.

Device Coverage:

Prioritize Android 10+ (Minimum SDK).

Target mid-range devices and tablets common among BukSU users.

Play Store Submission

Metadata Requirements:

Description: Highlight key RoomEase benefits:

"Book rooms instantly with real-time availability."

"List and manage your rooms anytime, anywhere."

Screenshots: Showcase both renter and owner interfaces.

Privacy Policy:

Explain usage of Firebase Analytics, email data, and location tracking.

Publishing Process:

Upload AAB (preferred for reduced app size).

Use staged rollout:

Start at 10%, monitor performance.

Increase to 50%, then 100% after stability is confirmed.

Post-Launch Monitoring

Monitoring Tools:

Firebase Crashlytics: Real-time crash tracking and analytics.

Google Play Console: Analyze ANRs (App Not Responding), user feedback, and performance.

Hotfix Mechanism:

Firebase Remote Config: Dynamically disable broken features without releasing a new build.

Rollout Strategy

Phase	Audience	Focus	Tools Used
Alpha	Dev team (5–10)	Core functionality, crash testing	Firebase Test Lab, GitHub
Beta	BukSU students (50–100)	UX feedback, real-world use	Firebase Surveys, Google Forms
Public	General users	Scalability, server load	Play Console, Firebase Crashlytics

XI. Appendices

Glossary

Term	Definition	RoomEase Usage Example
MVVM	Model-View-ViewModel pattern that separates UI and logic	Used for booking screen and room dashboard
FCM	Firebase Cloud Messaging for push notifications	Sends booking confirmation to renters and owners
JWT	JSON Web Token for secure user sessions	Encoded after login to authorize database requests
LiveData	Observable lifecycle-aware data holder	Automatically refreshes room availability on the screen

References & Resources

Google Firebase Documentation. (n.d.). Firebase Realtime Database, Authentication, and Cloud Functions. Retrieved from https://firebase.google.com/docs

Google Maps Platform Documentation. (n.d.). Maps SDK for Android. Retrieved from https://developers.google.com/maps/documentation/android-sdk

Android Developers. (n.d.). Material Design Guidelines. Retrieved from https://developer.android.com/design

Apple Developer Documentation. (n.d.). Human Interface Guidelines. Retrieved from https://developer.apple.com/design/human-interface-guidelines/

World Wide Web Consortium (W3C). (n.d.). Web Content Accessibility Guidelines (WCAG) 2.1. Retrieved from https://www.w3.org/TR/WCAG21/

Cloudinary Documentation. (n.d.). Media Upload, Storage, and Delivery. Retrieved from https://cloudinary.com/documentation