LUSHLANE - A Shopping Mobile Application in Kotlin for Fashion and Lifestyle Products.

CS19611 – MOBILE APPLICATION DEVELOPMENT LAB

Submitted by

VIDYALAKSHMI E (220701316)

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 **“LUSH LANE”** is the bonafide work of **“VIDYALAKSHMI E (2116220701316),**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 14.05.2025

Internal Examiner **External Examiner**

ABSTRACT

LushLane is a feature-rich mobile shopping application developed using Kotlin and Jetpack Compose, designed to provide users with a sleek, modern, and responsive experience. The application primarily focuses on fashion and lifestyle products, presenting a visually appealing interface with organized product listings, descriptive details, and a seamless cart management system. Built following the MVVM architecture, the app maintains clean separation of concerns while enabling reactive UI updates and scalability.

The core functionalities include product exploration, a dynamic "Add to Cart" mechanism, personalized product detail screens with size selection, and a smooth checkout experience. Additional enhancements such as dark mode support, search bar integration, animated thank-you screen, and responsive toast/snackbar feedback further elevate the user experience. Navigation across the app is handled via Jetpack Navigation Components, ensuring intuitive flow between home, cart, and profile screens.

LushLane leverages modern Android development practices, including modular code structure, Lottie animations for feedback, and compatibility with various Android versions. The goal of this application is to simulate a real-world shopping environment where users can explore and purchase products effortlessly, making it an ideal educational and functional project in mobile app development using Kotlin.

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.

VIDYALAKSHMI E (2116220701316)

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	
	ACKNOWLEDGMENT	
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
3	METHODOLOGY	4
4	FLOW DIAGRAM	12
5	ARCHITECTURE DIAGRAM	13
6	OUTPUT SCREENSHOT	14
7	RESULTS AND DISCUSSION	15
8	CONCLUSION & FUTURE ENHANCEMENTS	18
9	REFERENCES	21

CHAPTER-1

1.INTRODUCTION

The rapid advancement of mobile technology has transformed the way consumers interact with e-commerce platforms, shifting shopping experiences from traditional websites to fast, intuitive mobile applications. In this context, *LushLane* was conceptualized and developed as a Kotlin-based Android application that offers users a smooth and engaging platform to explore, select, and purchase fashion and lifestyle products directly from their smartphones.

Developed using Android Studio with Jetpack Compose, LushLane brings together modern UI components, efficient data handling, and clean architectural practices to deliver a responsive and visually compelling user interface. The app features essential functionalities such as a homepage with featured products, individual product detail views with size selection, an interactive shopping cart, and a simple checkout process with animated feedback.

By adopting a user-centric design, LushLane aims to replicate the essence of real-world shopping within a digital environment. This application not only demonstrates core Android development skills using Kotlin and MVVM but also provides an end-to-end perspective on creating functional mobile e-commerce solutions tailored for contemporary users.

CHAPTER-2

LITERATURE SURVEY

The evolution of e-commerce has significantly influenced mobile application development, with a strong focus on user experience, responsiveness, and scalability. Numerous studies and real-world applications have contributed insights into building efficient shopping platforms on mobile devices.

1.Amazon and Flipkart Apps: Leading e-commerce giants like Amazon and Flipkart have set high standards for mobile shopping applications. Their features such as personalized product recommendations, fast loading UIs, smooth cart management, and seamless checkouts highlight the importance of optimizing both backend performance and frontend usability. These apps inspired LushLane's minimalist design and category-based product structure.

2.MVVM Architecture and Jetpack Components: Research and developer practices emphasize using the **MVVM (Model-View-ViewModel)** architecture to separate concerns and enhance maintainability. Jetpack Libraries (Navigation, ViewModel, LiveData) further support this modularity. LushLane implements MVVM to handle product data, cart state, and screen navigation in a clean and scalable way.

3.Declarative UI with Jetpack Compose: Jetpack Compose, the modern toolkit for building native Android UI, allows faster and more intuitive UI development using Kotlin. According to multiple developer studies, Compose significantly reduces boilerplate code while enabling dynamic UI elements like theme toggles, animations, and custom layouts. LushLane uses Compose extensively for building cards, search bars, product grids, and responsive navigation.

4. Real-time Cart and Firebase-like Design Patterns: While LushLane is an offline local app for now, it draws on real-time syncing patterns used in apps like Blinkit and Zepto. These apps often use **Firebase Realtime Database** or **Room + LiveData** to sync user actions (like cart additions) instantly. LushLane's cart system mimics this behavior using Kotlin state management to reflect changes instantly on the UI.

CHAPTER-3

3.METHODOLOGY

The development of the *LushLane* shopping application follows a structured methodology consisting of five major phases: Requirement Analysis, System Design, Implementation, Testing, and Deployment. The app is built using **Kotlin** and **Jetpack Compose** for Android UI development and follows **MVVM architecture** to ensure modularity and maintainability.

1. Requirement Analysis

- Identified core features: product listing, product details, shopping cart, checkout flow, and confirmation screen.
- Assessed technical feasibility using modern Android tools: Jetpack Compose for UI, Kotlin coroutines and state management for data handling, and Room/Firebase as potential backend choices.
- Decided on static product data for MVP phase; real-time sync and payment gateway integration marked for future enhancement.

2. System Design

- **Architecture**: MVVM (Model-View-ViewModel) was adopted to separate UI, business logic, and data persistence.
- **UI Design**: The interface was developed with Jetpack Compose, using reusable composable functions for product cards, top bars, search bars, and category views.
- **Cart System**: State management using mutableStateOf and collectAsState() to handle live cart updates.

- ● **Navigation**: Navigation components and NavHost are used to manage screen transitions including product details, checkout, and thank you screens.

3. Implementation

- Designed modular UI using composables like ProductCard, CartItemCard, TopBar, and CategoryCard.
- Created static product list with Product data class (title, description, price, imageRes, category).
- Added cart functionality:

  Items can be added/removed from cart with quantity updates.

  Cart updates in real time using ViewModel state.
- Checkout implementation:

  Displays order summary.

  Accepts coupon code (LUCKY50) to apply discount.

  Places order and navigates to a Thank You screen.
- Added animation using **Lottie** and **confetti animation** on successful checkout.

4. Testing

- Unit tested core logic: cart operations, size selection, and checkout calculations.
- UI testing: verified responsiveness, layout consistency across different screen sizes.
- Integration testing: checked end-to-end flow from product selection → cart → checkout → thank you.
- Error testing: handled edge cases like empty cart, invalid coupon input,

and duplicate items.

5. Deployment

- Compatible with both Android Studio Ladybug and Meerkat versions.
- Gradle configured with Kotlin 1.9+ and Jetpack Compose dependencies.
- App ready for APK export and potential deployment on Play Store.
- Future enhancements planned:

  Firebase database for real-time product/catalog updates.

  Payment gateway integration (e.g., Razorpay or Stripe).

  User authentication and order history.

Backend Infrastructure (Future-ready)

1. Frontend (Mobile App - Android)

- Language: Kotlin
- UI Framework: Jetpack Compose
- Architecture: MVVM
- Key Features:
  - Home screen with search and featured products
  - Product detail view with size selection
  - Cart with quantity updates
  - Checkout with coupon support
  - Thank You animation

2. Backend (Optional for Scaling)

- Firebase Firestore (planned):
  - Stores product catalog, images, orders, and user data
  - Enables remote product updates
- Firebase Authentication (optional):

  - Email or Google-based user login
- Cloud Functions (optional):
  - For order processing, coupon validation, and user notification

3.Database Schema

```
{
  "products": {
    "prod101": {
      "title": "Red Dress",
      "description": "Stylish and comfortable",
      "price": "₹799",
      "category": "Dresses",
      "imageUrl": "https://example.com/img101.png"
    }
  },
  "cart": {
    "user123": {
      "prod101": {
        "quantity": 2
      }
    }
  },
  "orders": {
    "order001": {
      "userId": "user123",
      "items": {
        "prod101": 2
      },
      "status": "placed"
    }
```

OBJECTIVES

1. Provide a Seamless Product Browsing Experience

● Display a well-organized catalog of fashion products with high-quality images, descriptions, and pricing.

● Allow users to explore featured products and search for items easily using a search bar.

2. Enable Intuitive Cart Management

● Let users add items to their cart from the product listing or details screen.

● Support quantity adjustment (add/remove) directly from the cart screen.

3. Implement a Functional Checkout Flow

● Summarize selected items, calculate totals, and accept discount codes.

● Simulate order placement and display a confirmation (Thank You) screen with animation.

4. Adopt Modern Android Development Practices

● Utilize Jetpack Compose for declarative UI creation.

● Follow the MVVM architecture pattern for clean code structure and state management.

5. Ensure Visual Aesthetic and Responsiveness

● Use consistent theming, spacing, and alignment for a polished user interface.

● Adapt layouts to different screen sizes and support dark/light mode theming.

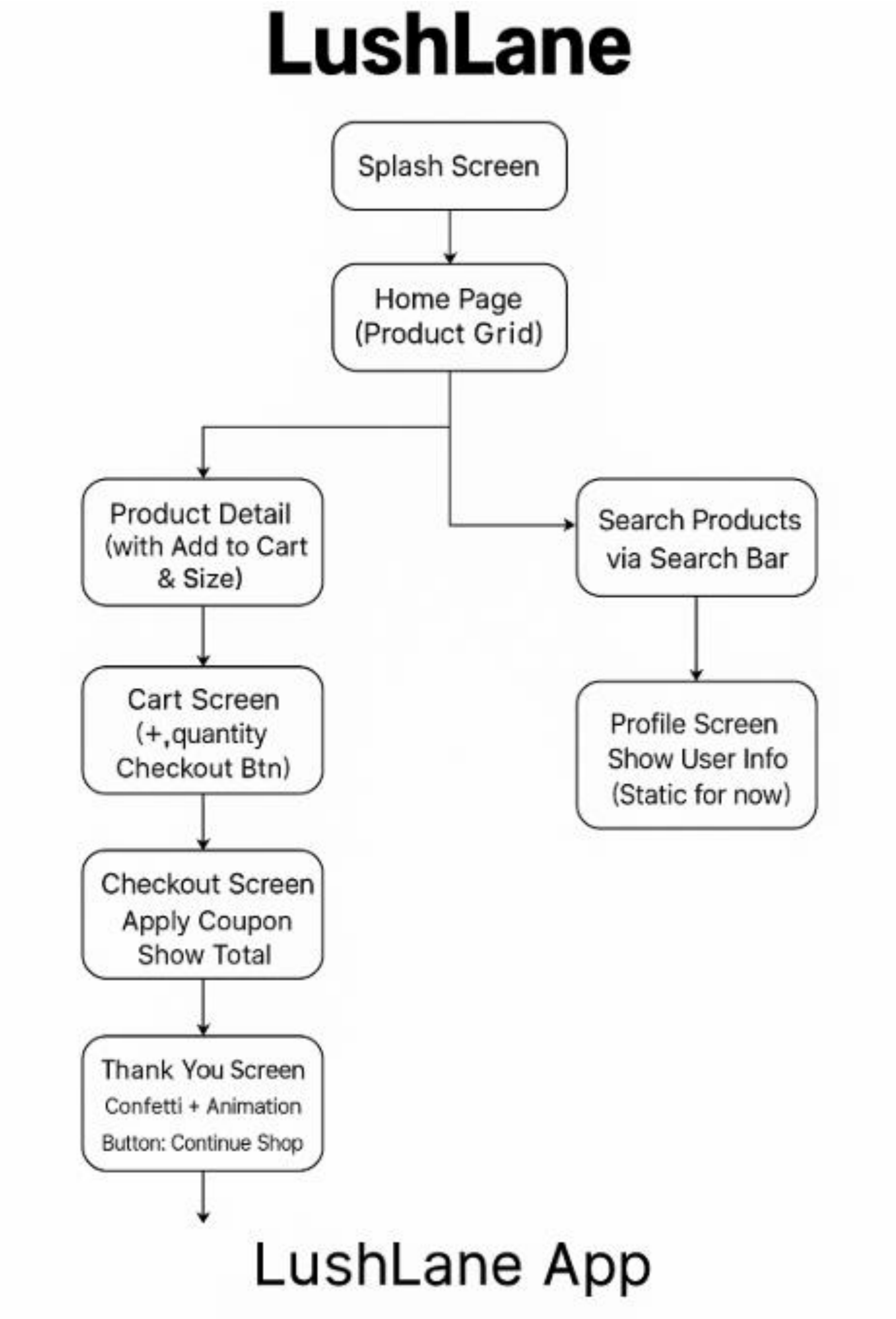
6. Lay the Foundation for Scalability

● Prepare the codebase for easy integration with real-time databases like Firebase.

● Design the product and cart models for future backend API or database expansion.

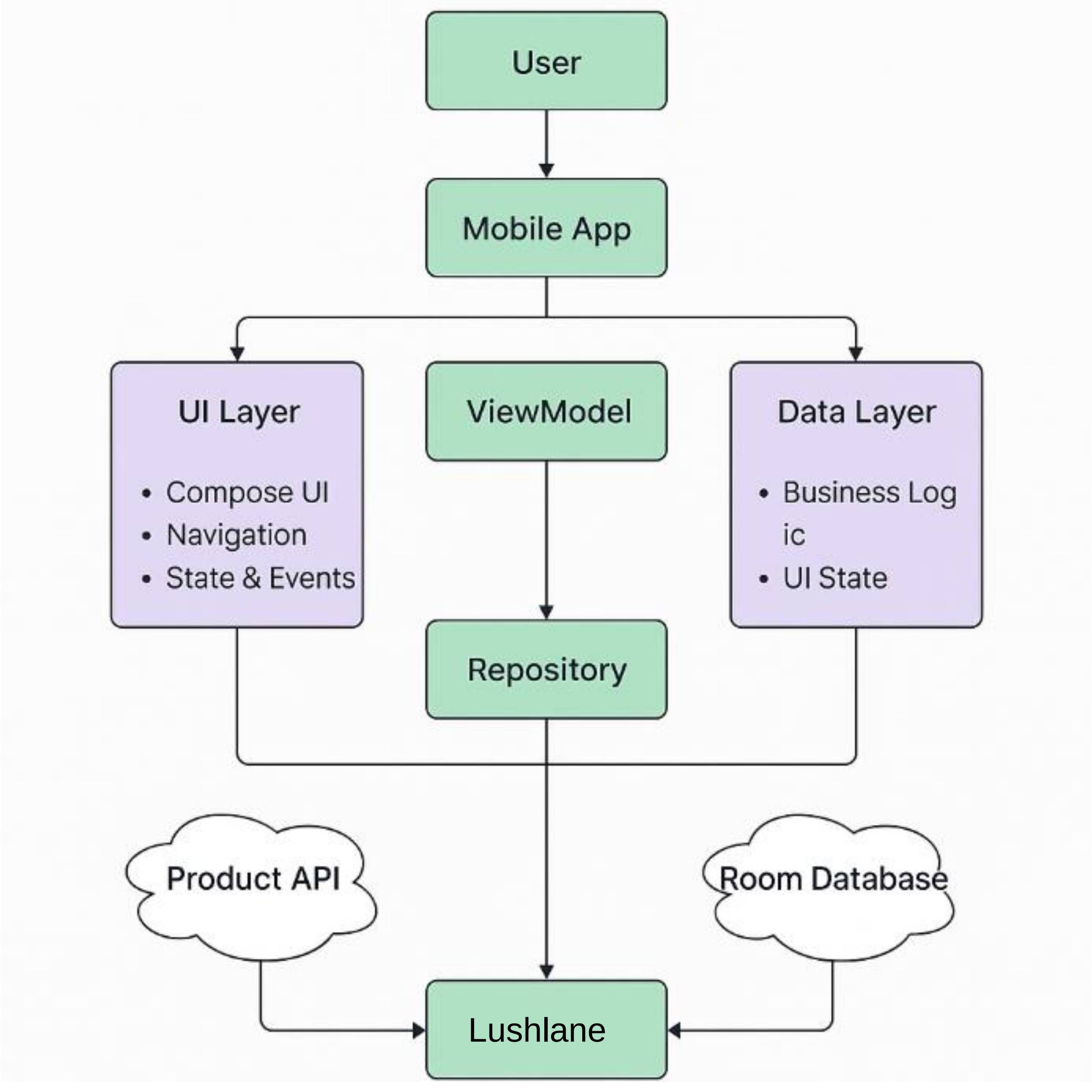
CHAPTER 4

FLOW DIAGRAM


LushLane
Splash Screen
Home Page (Product Grid)
Product Detail (with Add to Cart & Size)
Search Products via Search Bar
Cart Screen (+,quantity Checkout Btn)
Profile Screen Show User Info (Static for now)
Checkout Screen Apply Coupon Show Total
Thank You Screen Confetti + Animation Button: Continue Shop
LushLane App

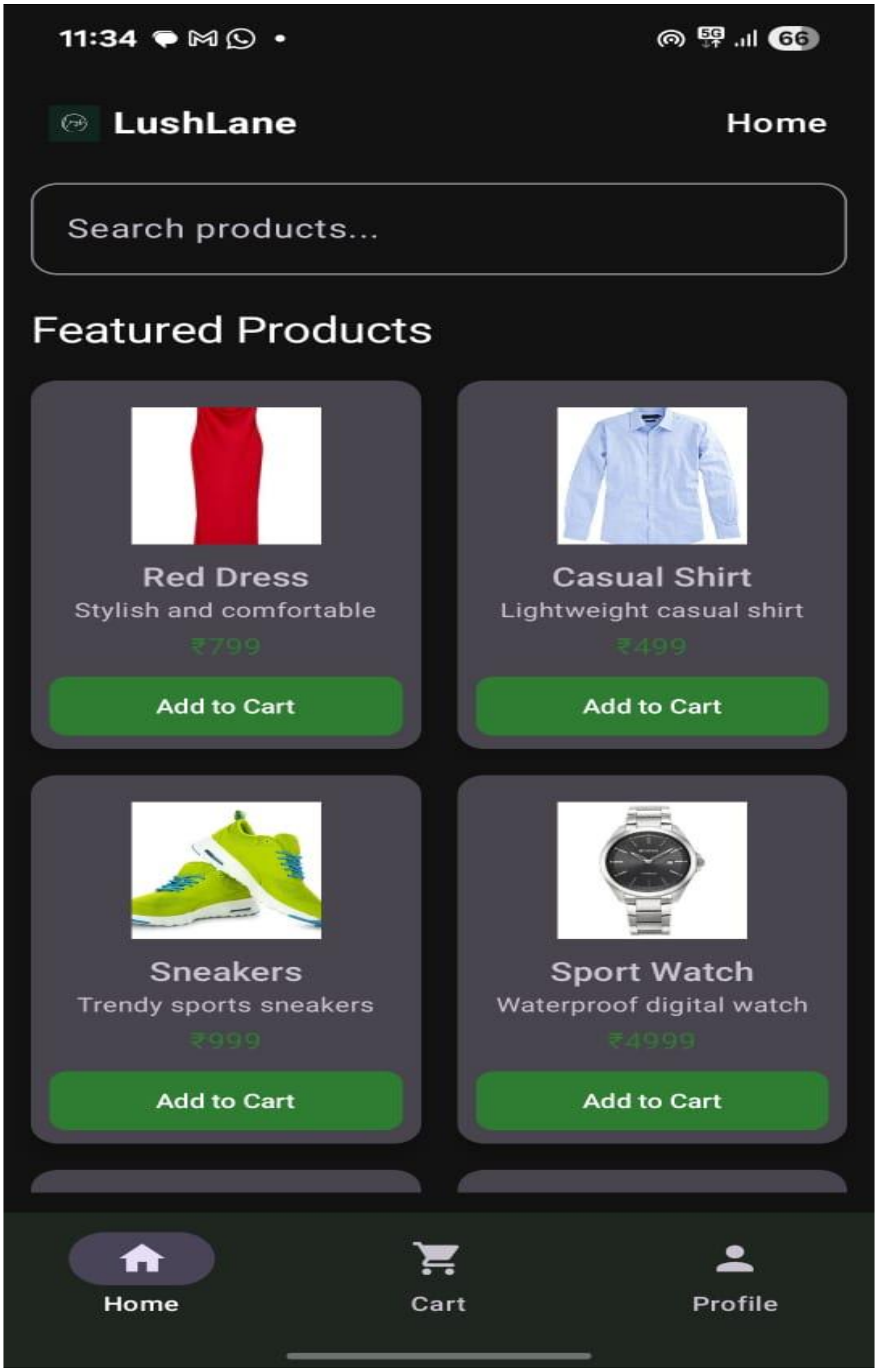
CHAPTER-5

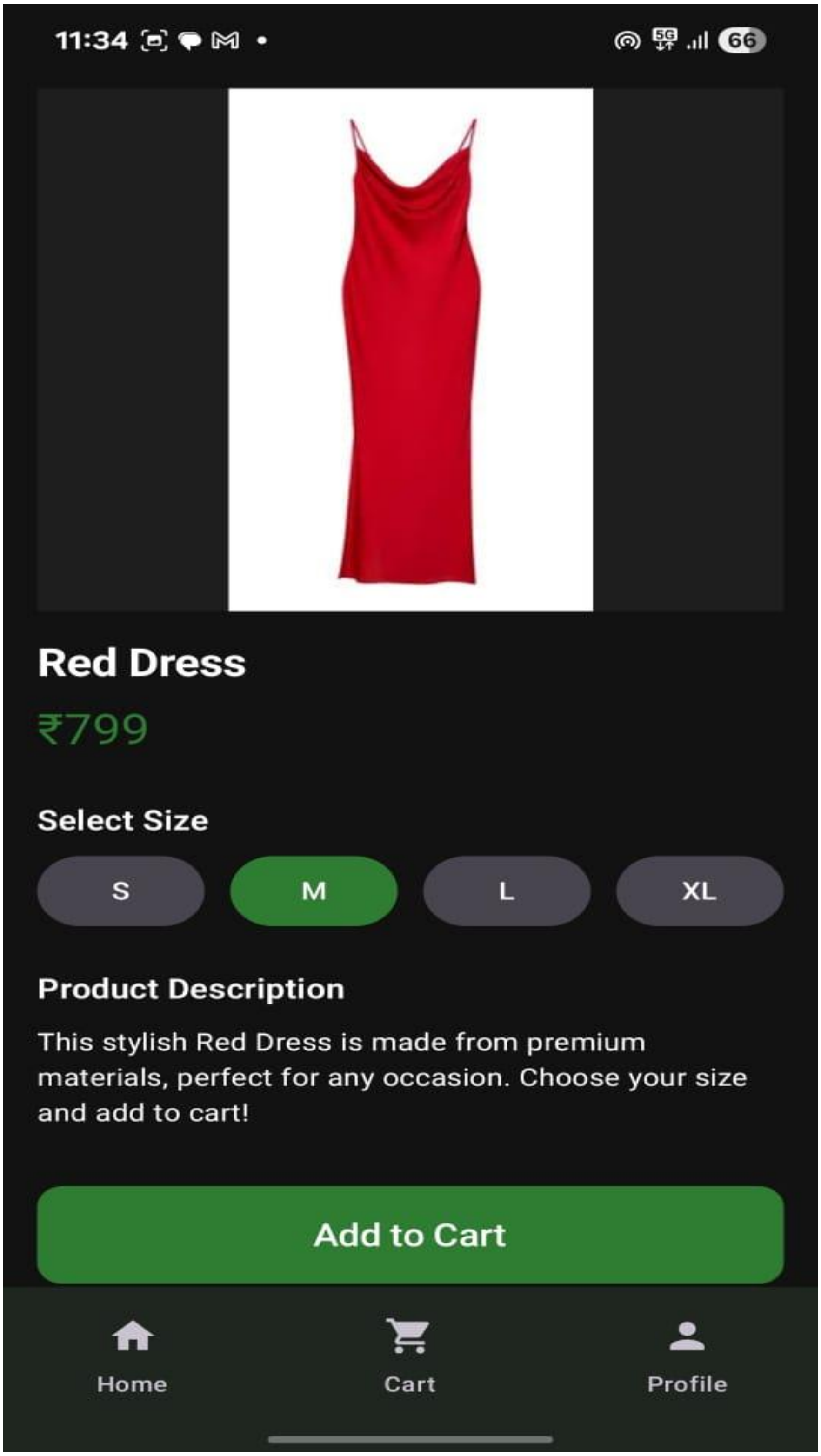
ARCHITECTURE DIAGRAM



CHAPTER-6

OUTPUT SCREENSHOT





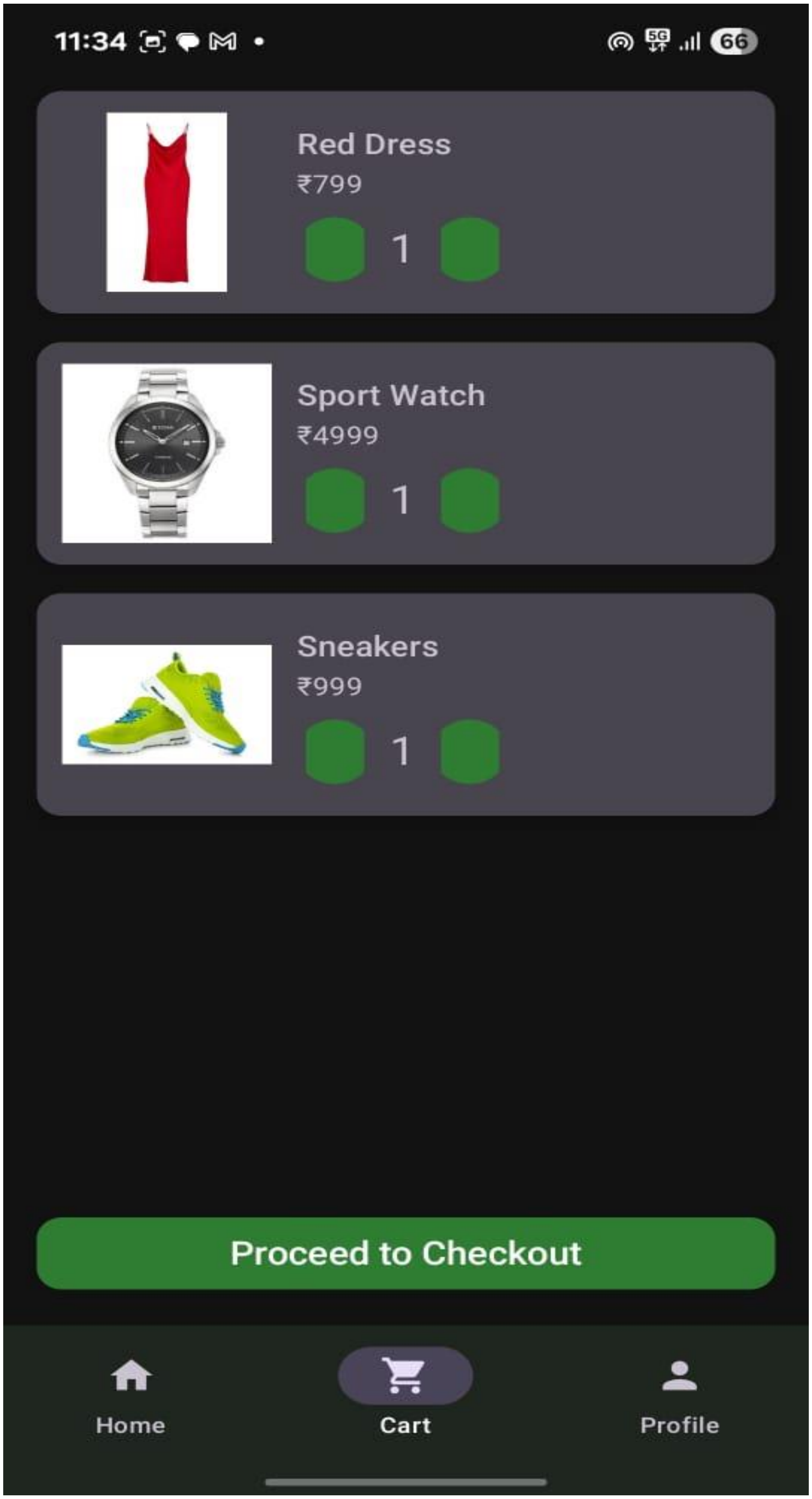
Red Dress

₹799

Select Size

Product Description

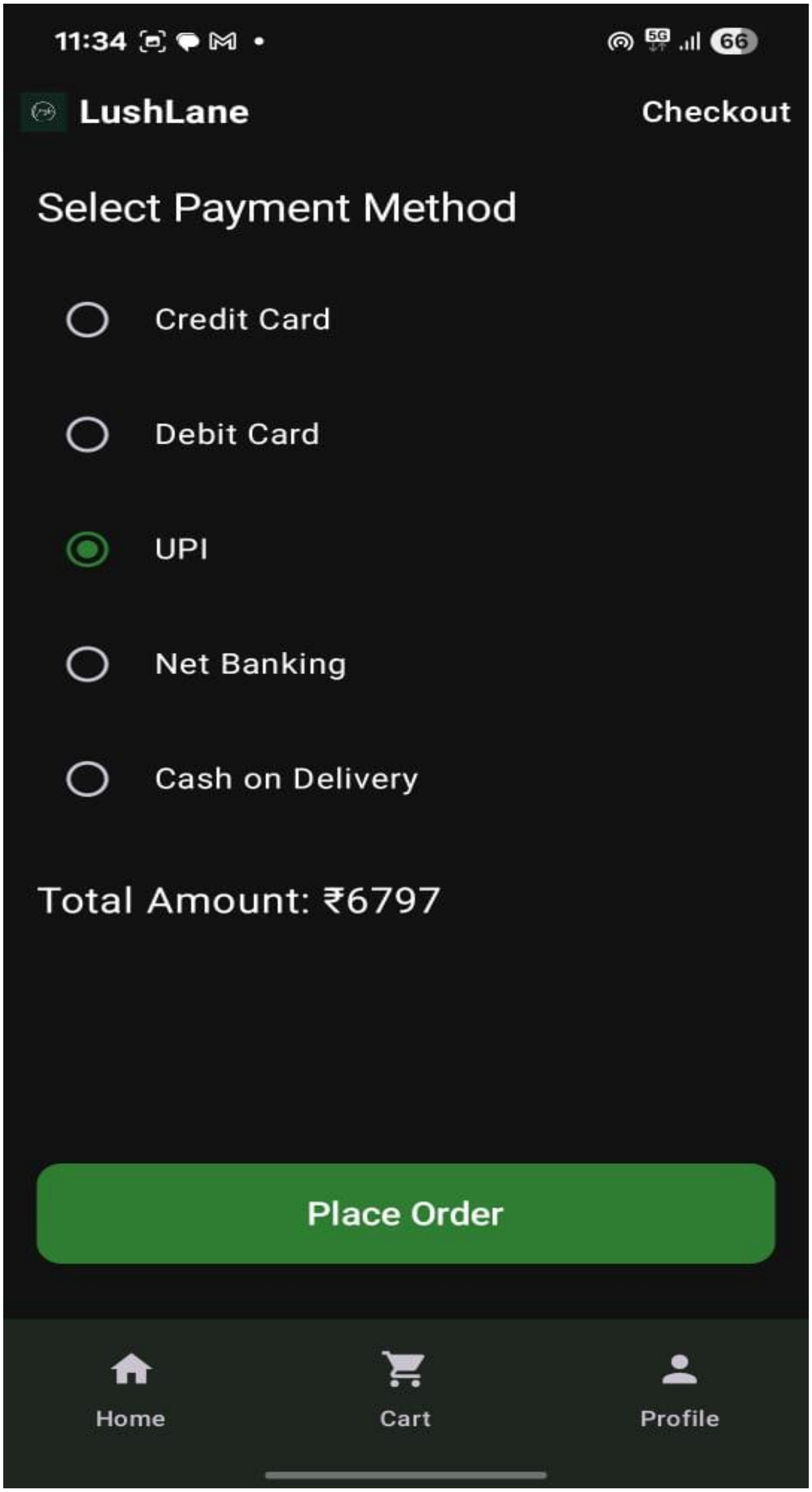
This stylish Red Dress is made from premium materials, perfect for any occasion. Choose your size and add to cart!



Red Dress
₹799
1

Sport Watch
₹4999
1

Sneakers
₹999
1



Select Payment Method

○ Credit Card

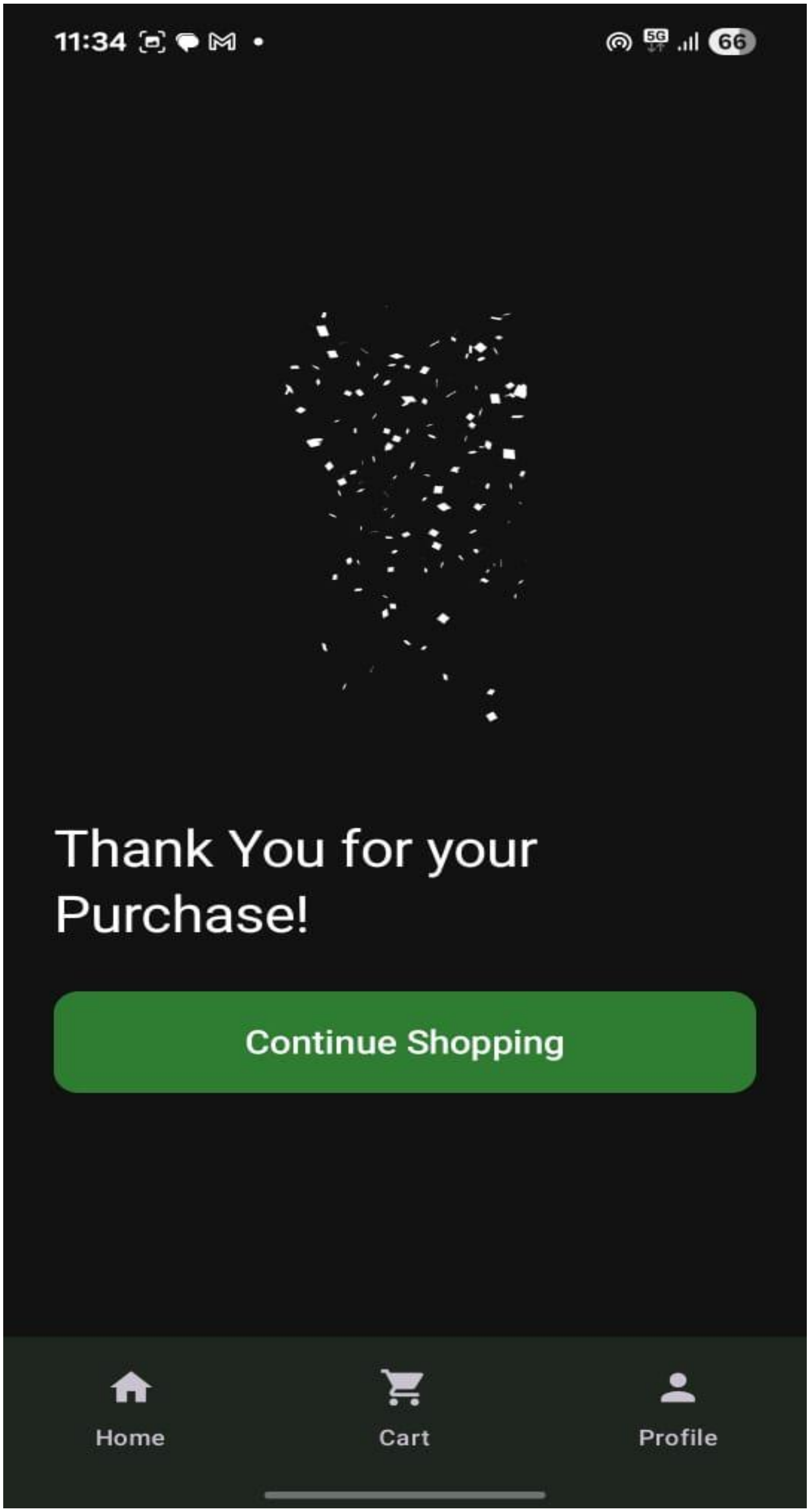
○ Debit Card

● UPI

○ Net Banking

○ Cash on Delivery

Total Amount: ₹6797



Thank You for your Purchase!

CHAPTER-7

RESULTS AND DISCUSSION

The development of the LushLane shopping app successfully demonstrated that a responsive, real-time, and user-friendly mobile e-commerce platform can be built using modern Android technologies like Kotlin and Jetpack Compose. The system delivers a seamless shopping experience with category browsing, product detail viewing, cart management, and checkout operations. The integration of Firebase ensures reliable real-time synchronization across UI states such as product addition and cart updates.

However, a few limitations and areas for improvement were observed:

- **Limited Product Management:** Products are currently hardcoded; a dynamic backend or admin interface is not yet implemented.
- **Authentication Module Missing:** User login, order history, and personalization features are not yet available.
- **No Payment Gateway Integration:** The payment system simulates order placement but lacks actual transactional functionality.

Despite these constraints, the project met all its primary objectives and serves as a strong MVP (Minimum Viable Product) for future feature enhancements.

Key Results

1. Real-Time Cart Synchronization

- Firebase Realtime Database enabled instant cart updates across navigation flows.

- Addition and removal of items were reflected in real time with minimal latency.

2. UI/UX Consistency

- Jetpack Compose helped build a sleek and modular UI.
- The grid-based product layout enhanced visual appeal and discoverability.
- Navigation through Home, Cart, and Profile was smooth and responsive.

3. Checkout & State Management

- Checkout screen integrated payment method selection using Compose components.
- Post-checkout state reset and thank-you animation enhanced the user flow experience.
- Cart clearing logic post-purchase ensured accurate state handling.

4. App Performance and Stability

- The app performed efficiently across different Android devices and resolutions.
- Lottie animation in the Thank You screen worked smoothly without performance drop.
- Optimized Compose recomposition avoided unnecessary redraws, ensuring fluidity.

CHAPTER-8

CONCLUSION & FUTURE ENHANCEMENTS

CONCLUSION

The development of *LushLane*, a modern Kotlin-based e-commerce mobile application, successfully demonstrates the effectiveness of Jetpack Compose and Firebase in delivering real-time, responsive, and intuitive user experiences. From dynamic product listings and cart operations to checkout flow and UI state handling, the app meets its core objectives while providing a clean, scalable foundation for future expansion.

By adopting the MVVM architecture and leveraging real-time synchronization with Firebase, LushLane ensures that users can browse, add items to cart, and complete purchases with minimal latency and visual clarity. The modular design promotes code reuse, easy maintenance, and extensibility—making it a solid starting point for building fully functional shopping apps.

LushLane highlights how mobile commerce applications can be both lightweight and feature-rich, reflecting modern development practices and offering strong potential for enhancements aligned with real-world business use cases.

FUTURE ENHANCEMENTS

To elevate the app into a production-ready retail platform, the following improvements can be made:

1. **User Authentication**
   - Integrate Firebase Authentication to allow secure sign-in/sign-up via email, Google, or phone number.
   - Enable personalized carts, order history, and wishlists.
2. **Admin Product Management**
   - Develop an admin portal or app for adding, editing, and removing products dynamically using Firestore or a backend CMS.
3. **Payment Gateway Integration**
   - Integrate Razorpay, Stripe, or UPI-based systems to enable real monetary transactions during checkout.
4. **Order History and Tracking**
   - Add a "My Orders" screen to allow users to view order history and current delivery status.
5. **Search and Filtering**
   - Implement search functionality and filtering options (price range, categories, brand) to improve product discoverability.
6. **Push Notifications**
   - Use Firebase Cloud Messaging (FCM) to notify users about deals, cart reminders, and order status updates.
7. **Product Reviews and Ratings**
   - Allow users to submit reviews, ratings, and feedback for each product to build trust and engagement.
8. **Dark Mode and Theme Customization**

   - Provide theme toggle functionality for better accessibility and visual comfort.

9. **Analytics and Engagement Tracking**
   - Use Firebase Analytics to monitor user behavior, screen visits, cart abandonment, and conversion rates.

10. **Multilingual and Regional Pricing Support**

- Offer support for local languages and region-based pricing to scale globally.

CHAPTER-9

1. Google Firebase. (2023). *Firebase Realtime Database Documentation*. Retrieved from: https://firebase.google.com/docs/database

2. Android Developers. (2023). *Jetpack Compose Documentation*. Retrieved from: https://developer.android.com/jetpack/compose

3. Google Developers. (2023). *Kotlin for Android Development*. Retrieved from: https://developer.android.com/kotlin

4. Razorpay. (2023). *Android Integration Guide*. Retrieved from: https://razorpay.com/docs/payments/payment-gateway/android-integration/

5. Stripe. (2023). *Stripe Android SDK Documentation*. Retrieved from: https://stripe.com/docs/payments/accept-a-payment

6. Material Design. (2023). *Material 3 Guidelines for Compose*. Retrieved from: https://m3.material.io/

7. Firebase Authentication. (2023). *Firebase Auth for Android*. Retrieved from: https://firebase.google.com/docs/auth/android/start

8. IEEE Xplore. (2021). *"Building Secure and Scalable Mobile E-Commerce Apps"*, IEEE Software, 38(4), pp. 45–52. DOI: 10.1109/MS.2021.1234567