Software Requirements Specification

for

eCommerce Application

Version 1.0

Prepared by

Yılmaz Tolga, 1902039

Course: COP4435
Date: 26.11.2024

Acknowledgement: Sections of this document are based upon the IEEE Guide to Software Requirements Specification (IEEE 830

1. Introduction

1.1 Purpose

The purpose of this document is to specify the software conditions for the eCommerce application developed using Kotlin in Android Studio. This document covers mobile eCommerce shopping service, banner ad display, product categories and listing of products in that category, prioritization of recommended products, product details, product commenting, compass of the whole operation including shopping cart.

1.2 Document Conventions

This SRS uses standard terminology for software requirements specifications. Use cases are represented using a standard format describing actors, preconditions, main success scenario, and alternate flows. Non-functional requirements are categorized and prioritized.

1.3 Intended Audience and Reading Suggestions

This SRS is intended for the following audience:

- Project Sponsors
- Development Team
- Testing Team
- Project Management
- Operations and Maintenance Teams

It is recommended that readers familiarize themselves with the overall architecture and functional requirements before delving into the specific design and implementation details.

1.4 Product Scope

This mobile application provides a comprehensive eCommerce platform. It encompasses the following key functionalities:

- Product Browsing and Search (by category)
- In app advertising with banners
- Product Details View (images, descriptions, models, price, rating, etc.)
- Shopping Cart Management (add, remove, update quantities)
- Shopping Cart View (products, quantities, prices, tax)

1.5 References

The references which designed according to APA rules:

- "Jetpack Compose Dersleri" video series by Demir, Tunahan. (n.d.). Youtube. Retrieved from https://www.youtube.com/playlist?list=PLwDmXhFTMvI_sN-N5Wp4b5kL84M4qP3TY
- worldsat's project208. (n.d.). GitHub. Retrieved from https://github.com/worldsat/project208
- "Online Shop Android Studio Project Jetpack Compose & Firebase – Ecommerce Programming" video content by UiLover Android (n.d.). Youtube. Retrieved from https://www.youtube.com/watch?v=XOmRT6x20RE

2. Overall Description

2.1 Product Perspective

This eCommerce application is a standalone mobile application, developed using Kotlin in Android Studio. It integrates Firebase for backend services such as real-time database. The application is designed to be intuitive, efficient, and scalable to meet the needs of both small and large-scale online retailers. It is built upon modern UI components like Jetpack Compose and adheres to Material Design principles.

2.2 Product Functions

- Product Catalog: Displaying product information, images, available models, price and rating.
- Search Functionality: Enabling users to find specific products by category search.
- Shopping Cart: Managing the user's selected items such as adding more product or removing.
- Product Commenting: Commenting on the selected product in the detailed screen of that product.

2.3 User Classes and Characteristics

The application serves two primary user groups:

- End Users (Customers): These are everyday users who browse and check out products. They require a user-friendly interface, quick navigation, and various models of products.

- Admin Users: Admins manage product listings, in-app advertisings, monitor ratings of products and handle adding new products. They need access to dashboard functionalities and analytics.

2.4 Operating Environment

On Android smartphones running compatible versions of the Android operating system, this eCommerce application functions. The Kotlin programming language is utilized in its development, and Android Studio the official Integrated Development Environment (IDE) for Android app development makes it easy to build and debug.

Furthermore, developers have the option to work on the project with Visual Studio Code, a code editor that is both lightweight and powerful. With the right modifications, it allows Kotlin development. This adaptability guarantees compatibility with Android development standards while enabling developers to utilize their favorite development environment.

The program needs enough RAM and processing power to run smoothly on the target Android devices when browsing in the app. It must also get along well with other applications and system functions so that there are no problems when using the Android platform. To provide a consistent user experience across various Android OS versions and device configurations, compatibility testing is crucial.

- Devices running Android 5.0 (API level 21) and above.
- Internet connectivity is required for Firebase Realtime Database operations.
- Devices with a minimum of 2GB RAM and 16GB storage.

2.5 Design and Implementation Constraints

- Firebase usage introduces limitations for backend operations (e.g., data structuring and query limits).
- Jetpack Compose and Kotlin constraints may affect UI design.
- Scalability requirements are not prioritized, not intended for large-scale user bases.

- Hardware Limitations: The application must operate efficiently on a range of Android devices with varying hardware specifications, including processing power and memory capacity.
- Software Interfaces: The software must adhere to Android platform standards and guidelines to ensure seamless integration with system components and APIs.
- Development Tools: The development and maintenance of the application will be primarily carried out using Android Studio for Kotlin-based development and Visual Studio Code with relevant extensions for code editing and debugging tasks.

Development choices are guided by these limitations to guarantee optimal performance, and compatibility with a variety of Android device setups.

2.6 User Documentation

User documentation components to be delivered with the software include:

- User Guide: Covers installation, core features (searching, viewing products, cart management), and common troubleshooting.
- Admin Guide: Step-by-step instructions for adding products, managing ads, and monitoring ratings.

2.7 Assumptions and Dependencies

- Users are assumed to have a stable internet connection to use the application.
- Firebase is assumed to be correctly configured and accessible.
- The application is optimized for smartphones; tablet or other platforms are not supported.

- Android Platform Updates: Assumption of continued compatibility with future Android OS versions, necessitating periodic updates and testing.
- Development Team Expertise: The project assumes that the development team possesses expertise in Kotlin programming, Android development using Android Studio, and familiarity with eCommerce frameworks for efficient development and maintenance.
- External Interfaces: Dependencies on external APIs or services for data analysis/reporting functionalities, requiring stable connections.
- Compliance Requirements: The project assumes adherence to relevant legal and regulatory requirements, especially concerning data privacy and security, impacting design and implementation decisions.

3. External Interface Requirements

3.1 User Interfaces

The user interfaces for the eCommerce application are crucial for providing an intuitive and engaging user experience. These interfaces include:

- Introduction screen: Displays a logo and slogan.
- Home screen: Displays banners, categories, recommended products and bottom navigation bar.
- Product details screen: Shows product information, images, available models, rating price and allows adding to cart.

- Shopping Cart screen: Shows products that added to cart, amount, prices, taxes/deliveries charges and check out button.
- Category screen: Shows products in the same category.

3.2 Hardware Interfaces

- Android devices with a minimum resolution of 720p.
- Supports devices with touchscreens, enabling easy navigation.
- Devices must have functional internet modules for real-time operations.

3.3 Software Interfaces

- Integrates with Firebase for database (storing product)
- Jetpack Compose facilitates a modern and responsive UI.
- Android OS (Version 5.0 and above): Interacts with system APIs for eCommerce application, storageaccess, and user interface components.
- Kotlin Programming Language: Utilized for coding logic and implementing app functionalities.
- Android Studio (Ladybug | 2024.2.1 Patch 2): Integrated development environment for coding, debugging,and testing the application.
- Third-party Libraries: Used for media format support, analysis/reporting tools, and enhancing application capabilities.

Data exchanges involve accessing the internet, handling user interactions, communicating with external libraries, and managing system resources efficiently.

3.4 Communications Interfaces

The eCommerce application does not heavily rely on external communications interface such as email, web browser, or network server protocols. However, it may utilize network communication for:

- Data Analysis/Reporting: Sending anonymized user data for analysis purposes to remote servers securely.
- Software Updates: Checking for and downloading application updates, if applicable.
- Secure HTTPS connections for Firebase interactions. Data formats (JSON, XML)

To ensure user privacy and application security, communication standards like HTTPS for safe data transfer, data encryption for sensitive information, and effective data transfer protocols are crucial.

4. System Features

4.1 Product Browsing and Search

4.1.1 Description and Priority

Allows users to browse the product catalog, view product details, and add items to their shopping cart. Priority: High

4.1.2 Stimulus/Response Sequences

- Users select a category to view related products.
- The system displays products with filters for price and ratings.

Use Case 1: Browse Products by Category

- Actor: Customer
- Main Success Scenario: User selects a category; application displays products in that category. User can scroll through products and select a product to view details.
- Alternate Flows: Category is empty (display a "no products found" message). Network error occurs (display an error message).

Use Case 2: View Product Details

- Actor: Customer
- Precondition: User has selected a product from the catalog.
- Main Success Scenario: Application displays detailed product information, including images, description, price, and availability. User can add the product to their cart.
- Alternate Flows: Product data is unavailable (display an error message).

4.1.3 Functional Requirements

- The application shall display product categories on the home screen.
- The application shall display product listings with images, titles, and prices.
- The application shall allow users to filter products by category.
- The application shall display detailed product information when a product is selected.
- The application shall allow users to add products to their shopping cart.

4.2 Shopping Cart Management

4.2.1 Description and Priority

Essential for managing user-selected items. The application provides a shopping cart where users can add, remove, or update the quantities of products. Priority: High

4.2.2 Stimulus/Response Sequences

- Add items to the cart and adjust quantities.
- Proceed to checkout with total price calculation..

4.2.3 Functional Requirements

- Users can add products to the cart from the product details page.
- Users can remove products or change quantities in the cart.
- The total price in the cart must update dynamically.

4.3 Admin Features and Dashboard

4.3.1 Description and Priority

High-priority feature for managing inventory and advertisements. Admins can add products, manage advertisements, and review user ratings of products.

4.3.2 Stimulus/Response Sequences

Admin adds new products, edits descriptions, or updates banners.

4.3.3 Functional Requirements

- Add new products and edit existing ones.
- Configure and manage advertisement banners.
- View product ratings submitted by users

5. Other Nonfunctional Requirements

5.1 Performance Requirements

eCommerce Application Response Time:

Requirement: After a user acts, such as hitting a button, the program must perform within two seconds.
Justification: Fast reaction times improve the app's user experience and engagement.

Utilization of Resources:

Requirement: During regular operations, the application's CPU and memory use cannot exceed 30%.
Justification: Efficient use of resources guarantees fluid perform without resulting in device crashesor slowdowns.

Latency in the network:

Requirement: Over regular network connections, the app must perform content with a buffering latency of less than three seconds.
Justification: Cutting down on buffering time enhances the viewing experience and lessens user annoyance.

5.2 Safety Requirements

Regular backups of Firebase database to prevent data loss.

5.3 Security Requirements

Encryption of Data:

Encrypt sensitive user data that is stored locally or sent over networks, such as user preferences.
Justification: Protect user data's confidentiality and integrity from tampering or illegal access.

5.4 Software Quality Attributes

Reliability:

Requirement: During a one-month period, the program must have at least 99% uptime with few crashes or faults.
Justification: In order to prevent disruptions during navigation, users anticipate eCommerce app to operate with dependability.

Usability:

Requirement: The application must follow Android design standards and include simple controls for adding a product to cart.
Justification: User happiness is increased, and app usage is encouraged when an app is easy to use.

5.5 Business Rules

Levels of User Access:

Rule: The ability to change and add products to the app is restricted to admin.
Justification: Prevents unwanted modifications to program.