1. Intro

1.1 Purpose

The purpose of this document is to specify the requirements for the development of the ElectronicMPSTME Android app and its associated Firebase Realtime Database.

1.2 Scope

This document outlines the functional and non-functional requirements for the Android app and the database. It covers user interactions, data storage, security, and performance.

1.3 Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification
- Firebase: Firebase Realtime Database
- Android App: ElectronicMPSTME Android Application

1.r Overview

This SRS provides an overview of the ElectronicMPSTME Android app's features and its integration with the Firebase Realtime Database.

2. System Overview

2.1 System Description

The ElectronicMPSTME Android app allows users to browse and purchase electronic products. It includes features such as product search, user registration, order management, and product reviews. The Firebase Database serves as the backend for storing product data, user profiles, and order information.

2.2 System Architecture

The Android app follows a client-server architecture, with the app running on the client device and interacting with the Firebase Realtime Database on the server.

2.3 Data Flow Diagram

3. **Android App Requirements**

3.1 Functional Requirements

3.1.1 User Registration and Login

- Users can register for an account using their email and password.
- Registered users can log in to their accounts.

3.1.2 Browse and Search Products

- Users can browse a catalog of electronic products.
- Users can search for products based on keywords and filters.

3.1.3 Product Details

- Users can view detailed information about a product, including name, description, category, price, and image.

3.1.4 Shopping Cart

- Users can add products to their shopping cart.
- Users can review and modify the contents of their shopping cart.

3.1.5 Order Placement

- Users can place orders for products in their shopping cart.
- Users provide shipping information during order placement.

3.1.6 Order Tracking

- Users can track the status of their orders.
- Order status is updated in real-time.

3.1.7 Product Reviews

- Users can submit reviews and ratings for products.
- Reviews and ratings are displayed to other users.

3.2 Usability Requirements

3.2.1 User Interface Design

- The app's user interface is intuitive and user-friendly.

3.3.2 Load Handling

- The app can handle concurrent user sessions without performance degradation.

3.4 Security Requirements

3.4.1 Data Encryption

- User data, including login credentials, is encrypted during transmission.
- Sensitive data in the Firebase database is encrypted at rest.

3.4.2 Authentication

- User authentication is required for all actions that modify user data.

3.4.3 Authorization

- Users can only access and modify their own data.
- Managers have access to administrative functions.

3.5 Supportability Requirements

3.5.1 Logging and Error Handling

- The app logs errors and exceptions for debugging.
- Users are provided with error messages for failed actions.

3.5.2 Upgrades and Maintenance

- The app can be updated with new features and bug fixes.

4.1 Database Design

4.1.1 Products Table

- Stores product information, including name, description, category, price, stock quantity, and product image.

4.1.2 Managers Table

- Contains manager information, including first name, last name, email, and phone.

4.1.3 Customers Table

- Stores customer profiles, including first name, last name, email, phone, and address.

4.1.4 Orders Table

- Records order details, including user ID, order date, total amount, and order status.

4.1.5 OrderItems Table

- Associates products with orders, including product ID and quantity.

4.1.6 Reviews Table

- Contains product reviews, including product ID, customer ID, rating, review text, and review date.

4.1.7 ManagerCredentials Table

- Stores manager login credentials, including manager username and password.

4.1.8 CustomerCredentials Table

- Contains customer login credentials, including customer username and password.

4.2 Data Integrity Requirements

4.2.1 Primary Keys

- Each table has a primary key to ensure data uniqueness.

4.2.2 Foreign Key Constraints

- Foreign key constraints are enforced to maintain data integrity.

4.2.3 Indexing

- Appropriate indexing is applied to enhance data retrieval performance.

4.3 Data Access Requirements

4.3.1 CRUD Operations

- The app can perform Create, Read, Update, and Delete operations on data.

4.3.2 Data Retrieval

- Data retrieval queries are optimized for efficient access.

4.4 Security and Privacy

4.4.1 Access Control

- Role-based access control is implemented to restrict data access.

4.4.2 Data Encryption

- Data transmission and storage are encrypted to protect sensitive information.

4.5 Scalability and Performance

4.5.1 Read and Write Operations

- The database can handle a high volume of read and write operations.

4.5.2 Query Optimization

- Query performance is optimized for efficient data retrieval.

4.6 Backup and Recovery

4.6.1 Data Backup

- Regular database backups are performed to prevent data loss.

4.6.2 Disaster Recovery

- Disaster recovery procedures are in place to ensure data availability in case of failures.