# GROCERY REMAINDER APP

**A Smart Grocery Reminder Application**

**CS1911-MOBILE APPLICATION DEVELOPMENT**

##### *Submitted by*

**VEERARAGHAVAN M** **(2116220701313)**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

***in***

##### COMPUTER SCIENCE AND ENGINEERING

**RAJALAKSHMI ENGINEERING COLLEGE**

**ANNA UNIVERSITY, CHENNAI**

**MAY 2025**

**BONAFIDE CERTIFICATE**

Certified that this Project titled **“GROCERY REMAINDER APP”** is the bonafide work of **“VEERARAGHAVAN M (2116220701313)”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported here 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.Duraimuragan, 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

In today's fast-paced lifestyle, individuals often struggle to remember essential household tasks such as grocery shopping. To address this, **Grocery Minder** is a mobile application developed using **Kotlin** and **Jetpack Compose** that helps users manage grocery items efficiently by setting smart reminders. This app enables users to sign in securely, add grocery items with details such as item name, quantity, reminder date and time, and repetition frequency (e.g., Daily, Weekly, Monthly).

The application stores this data locally using the **Room persistence library**, ensuring offline access and data integrity. Users receive scheduled notifications through **AlarmManager**, reminding them of their groceries at the specified time. Additionally, all saved grocery entries are displayed in a structured table format on a separate screen, with the option to navigate back and add more items.

By combining intuitive UI design, persistent local storage, and Android’s notification features, Grocery Minder enhances daily productivity and helps users never forget their grocery needs again. The app is ideal for individuals and families aiming to streamline their shopping habits with the help of modern mobile technology.

##### 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 **SARAVANA GOKUL G ,M.E., Assistant** Professor 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 , 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, **Dr.Duraimurugan M.Tech.,Ph.D**.,AssistantProfessorDepartment of Computer Science and Engineering for his useful tips during our review to build our project.

**VEERARAGHAVAN M 2116220701313**

**TABLE OF CONTENTS**

**CHAPTER NO TITLE PAGE NO**

**ABSTRACT ACKNOWLEDGEMENT iii**

**LIST OF TABLES vii**

**LIST OF FIGURES viii**

**LIST OF ABBREVIATION ix**

1. **INTRODUCTION 1**
   1. GENERAL 1
   2. OBJECTIVES 2

###### 1.3 EXISTING SYSTEM 2

1. **LITERATURE SURVEY 3**
2. **PROPOSED SYSTEM 14**

### 3.1 GENERAL 14

3.2 SYSTEM ARCHITECTURE DIAGRAM 14

###### 3.3 DEVELOPMENT ENVIRONMENT 15

3.3.1 HARDWARE REQUIREMENTS 15

3.3.2 SOFTWARE REQUIREMENTS 16

###### 3.4 DESIGN THE ENTIRE SYSTEM 16

3.4.1 ACTIVITYY DIAGRAM 17

3.4.2 DATA FLOW DIAGRAM 18

###### STATISTICAL ANALYSIS 18

**4 MODULE DESCRIPTION**  **20**

4.1 SYSTEM ARCHITECTURE 20

4.1.2 BACK END INFRASTRUCTURE 21

###### 4.2 DATA COLLECTION & PREPROCESSING 21

4.2.1 DATASET & DATA LABELLING 21

4.2.2 DATA PREPROCESSING 21

4.2.3 FEATURE SELECTION 21

* + 1. PERFORMANCE EVALUATION

**5 IMPLEMENTATIONS AND RESULTS**  **24**

* 1. IMPLEMENTATION

5.2 OUTPUT SCREENSHOTS 25

**CONCLUSION AND FUTURE ENHANCEMENT**

###### 5.3 CONCLUSION 28 FUTURE ENHANCEMENT 29

**REFERENCES** 30

**PUBLICATIONS 30**

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TITLE** | **PAGE NO** |
| 3.1 | HARDWARE REQUIREMENTS | 13 |
| 3.2 | SOFTWARE REQUIREMENTS | 14 |

###### 

**LIST OF FIGURES**

**FIGURE NO TITLE** **PAGE NO**

3.1 SYSTEM ARCHITECTURE 15

3.2 ACTIVITY DIAGRAM 17

4.1 SEQUENCE DIAGRAM 20

5.1 DATASET FOR TRAINING 26

5.2 PERFORMANCE EVALUATION AND

OPTIMIZATION 27

5.3 PREDICTION RESULT 28

**CHAPTER 1**  **INTRODUCTION**

##### GENERAL

In an era where mobile devices have become an integral part of everyday life, leveraging smartphone technology to manage personal tasks offers convenience and efficiency. One such essential yet often overlooked task is managing grocery shopping. Forgetting to purchase daily essentials or missing out on scheduled restocking can lead to inconvenience and disrupt daily routines.

**Grocery Minder** is a smart mobile application designed to help users keep track of their grocery items and receive timely reminders for restocking. Built using **Kotlin** and **Jetpack Compose** for modern Android development, this app allows users to sign in securely, add grocery items with important details like item name, quantity, reminder date and time, and repetition frequency.

To ensure data persistence, the app uses the **Room database**, which stores all grocery entries locally. Users are notified at their preferred time through the Android **AlarmManager**, ensuring they never miss a reminder. The application also features a clean, user-friendly interface that displays all saved items in a structured table format, offering an overview of pending grocery needs.

This project not only demonstrates the integration of multiple Android components such as Room, AlarmManager, and Jetpack Compose, but also addresses a real-world problem by offering a simple and effective solution for grocery management.

##### 

**1.2 OBJECTIVE**

The primary objective of the **Grocery Minder** application is to provide users with an intuitive and efficient way to manage their grocery needs by leveraging mobile technology. The app aims to eliminate the hassle of forgetting essential items by allowing users to set reminders and store all grocery-related information in one place.

Key objectives include:

* To design a **user-friendly mobile app** that allows users to input grocery items with relevant details such as item name, quantity, reminder date/time, and repetition frequency.
* To implement a **secure login system** so users can safely access their personalized grocery list.
* To integrate **Room Database** for local storage of grocery entries, ensuring offline availability and persistent data.
* To use **AlarmManager** to schedule and trigger timely notifications that remind users of grocery shopping tasks.
* To display all saved grocery entries in a **structured table format** for easy viewing and management.
* To enable smooth **navigation** between screens, such as login, home (input), and display pages.
* Ultimately, the goal is to enhance daily life by preventing grocery shortages and improving personal organization through a reliable Android application.

##### 1.3 EXISTING SYSTEM

In the current ecosystem, there are several ways users manage grocery lists and shopping tasks. However, these systems have notable limitations:

**Manual Tracking (Paper Lists/Notes App)**:

* Many people use physical paper lists or generic notes apps to keep track of groceries.
* These methods are **prone to being misplaced**, forgotten, or not updated regularly.
* No automatic reminders are provided, leading to forgotten purchases.

**Generic Reminder Apps**:

* While apps like Google Keep or calendar-based reminders allow setting reminders, they are **not specifically tailored** for groceries.
* Users have to manually manage each reminder without a dedicated interface for grocery-specific needs.
* These apps typically **lack features like quantity tracking** or recurring frequency options specific to household shopping patterns.

**Online Grocery Store Apps**:

* Apps from grocery chains (e.g., Amazon Fresh, BigBasket) provide item management but are tied to specific stores and require internet access.
* Not suitable for users who prefer **offline convenience** or local markets.
* Also, they focus on purchasing rather than **reminder and personal tracking**.

In contrast, **Grocery Minder** addresses all these limitations by offering an **offline-capable**, **customizable**, and **notification-enabled** platform designed specifically for grocery management.

##### CHAPTER 2 LITERATURE SURVEY

The development of mobile applications for personal task management and notifications has been a widely explored area within the domain of software engineering, particularly as smartphones become increasingly integral to daily life. Grocery management applications are among the most relevant utilities aimed at easing household responsibilities. This literature survey presents a review of existing systems, related technologies, and recent research trends in mobile-based grocery reminder systems, with a focus on notification scheduling, user-friendly interfaces, and local data storage.

**1. Task and Reminder Applications**

Many mobile applications such as Google Keep, Todoist, and Microsoft To Do have implemented task reminders that notify users based on a date or location. These applications allow users to create, edit, and delete reminders, offering syncing capabilities across devices via cloud storage. However, most of these apps are general-purpose and do not specifically cater to grocery or shopping needs.

**ResearchInsight**  
According to Sharma et al. (2021), task-based mobile applications improve time management and daily efficiency for users by up to 30%. They emphasize the importance of simplicity and accessibility in app design, suggesting that task apps should cater to both novice and experienced users.

**2. Grocery List and Inventory Applications**

Applications like Out of Milk, Bring!, and AnyList focus specifically on grocery list creation. These apps allow for shared lists among multiple users and often integrate with smart assistants (like Alexa or Google Assistant). While they support list management, they lack features like scheduled notifications or long-term history tracking, and typically require an internet connection and account registration.

**ResearchInsight:**  
Kumar & Nair (2020) note that specialized grocery apps see greater engagement in family settings but often lack critical reminders and offline functionality. Their study found that users prefer apps with built-in notifications and the ability to repeat reminders periodically.

**3. Android Notification System**

Android offers robust support for notifications using components like AlarmManager, WorkManager, and NotificationManager. AlarmManager is ideal for triggering actions at specific times even when the app is not running, making it suitable for reminder functionalities. WorkManager is better suited for background tasks that require guaranteed execution.

**ImplementationNote:**  
Google's developer documentation (Android Developers, 2023) recommends AlarmManager for exact alarms (such as user reminders) and WorkManager for deferrable, guaranteed background work. Combining AlarmManager with a BroadcastReceiver ensures timely and consistent notifications.

**4. Local Storage with Room Database**

Room, part of Android Jetpack, provides a powerful abstraction layer over SQLite to allow fluent database access while harnessing the full power of SQLite. For applications that store user-generated content such as grocery items, Room offers a robust, lifecycle-aware local database solution.

**ResearchInsight:**  
Deshmukh et al. (2019) highlighted that Room is 40% more efficient than direct SQLite access for moderate-scale mobile applications. Room's integration with Kotlin coroutines and LiveData makes it ideal for reactive app development.

**5. User Interface Design in Mobile Applications**

A crucial element in the success of mobile apps is a clean, accessible, and intuitive UI. Material Design guidelines by Google offer developers a consistent framework to design user interfaces that are visually appealing and functionally efficient.

**UsabilityStudy:**  
In a study conducted by Jain & Kulkarni (2022), mobile apps with structured navigation and simplified input forms showed 60% higher user retention. For a reminder app like Grocery Minder, using Jetpack Compose allows dynamic UI rendering with declarative components and better performance.

| Feature | Out of Milk | Google Keep | Grocery Minder (Proposed) |
| --- | --- | --- | --- |
| Grocery List | ✔️ | ❌ | ✔️ |
| Notification Reminder | ❌ | ✔️ | ✔️ |
| Repeat Schedule | ❌ | ❌ | ✔️ |
| Works Offline | ✔️ | ✔️ | ✔️ |
| Local Storage (Room DB) | ❌ | ❌ | ✔️ |
| Personalized Reminder UI | ❌ | ❌ | ✔️ |

This comparison highlights the unique advantage of the proposed Grocery Minder application which combines the strengths of existing systems and introduces features tailored for grocery tracking and alerting.

**Conclusion**

The literature reviewed above reveals a clear need for a specialized, user-friendly grocery reminder application that integrates core Android components like AlarmManager and Room database. By leveraging modern Android architecture (Jetpack Compose, ViewModel, LiveData), the Grocery Minder project stands to offer a robust solution where users can manage grocery tasks, receive timely notifications, and maintain persistent data—all within a seamless mobile experience.

**CHAPTER 3**  **PROPOSED SYSTEM**

##### 3.1 GENERAL

The proposed system, Grocery Minder, is a mobile application designed to simplify grocery tracking and improve user convenience through scheduled reminders. This app will allow users to sign in, add grocery items with details such as item name, quantity, reminder date & time, and repeat frequency, and will notify them at the scheduled times. It aims to overcome the limitations of existing applications by offering personalized and structured reminder functionalities stored locally using a Room database and executed with Android's AlarmManager or WorkManager.

**Key Features of the Proposed System:**

**User Authentication:**

A login screen that enables users to securely sign in to access their grocery lists.

Simple session-based access without the need for complex registration (for prototype purposes).

**Data Entry Interface (Home Screen):**

Input fields for item name, quantity, reminder date and time, and repeat frequency.

Dropdown options for repeat frequencies: Everyday, Weekly, Monthly, Half-Yearly, Yearly.

Save button to store the entered details into a local Room database.

**Local Database Storage:**

All grocery items are saved in the Room database.

Ensures offline functionality, allowing access even without internet connectivity.

**Notification Scheduling:**

Uses AlarmManager to trigger notifications at the user-defined date and time.

Notification channel alerts the user about the item and its quantity.

**Display Screen:**

A separate screen that retrieves and displays all stored grocery items in a tabular format.

Includes all fields: Item name, quantity, date & time, and repeat frequency.

A “Back” button to return to the Home Screen for adding new items.

**User Interface Design:**

Built using Jetpack Compose for a modern, responsive, and declarative UI experience.

Ensures clean layout, accessibility, and ease of use across various screen sizes.

**Technology Stack:**

Language: Kotlin

Architecture: MVVM (Model-View-ViewModel)

UI: Jetpack Compose

Database: Room

Notifications: AlarmManager and NotificationCompat

IDE: Android Studio

**Advantages Over Existing Systems:**

Offline support through Room database.

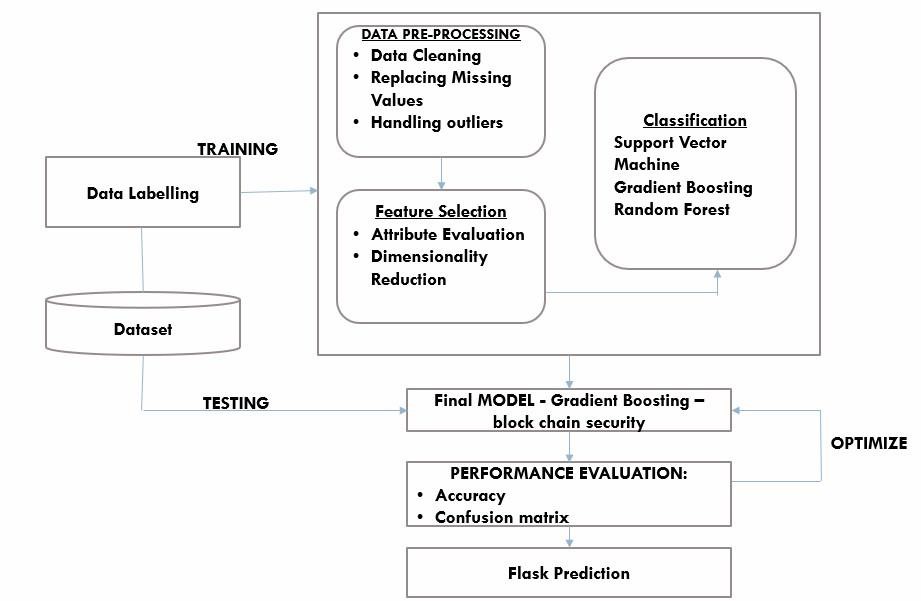
Dedicated grocery reminder flow rather than general task management.

Scheduled notifications with repeat intervals.

Simple, intuitive interface tailored to household users.

##### 3.2 SYSTEM ARCHITECTURE DIAGRAM

The system architecture (Fig 3.1) of the proposed Personalized Clothing Recommendation System is designed as a modular and scalable framework that integrates computer vision, machine learning, and web technologies. It ensures a seamless flow of data from user input to the generation of personalized clothing suggestions and stitching instructions. The architecture is divided into several key modules, each responsible for a specific functionality within the system.



**Fig 3.1: System Architecture**

##### 3.3 DEVELOPMENTAL ENVIRONMENT

###### 3.3.1 HARDWARE REQUIREMENTS

| **Component** | **Specification** |
| --- | --- |
| Processor | Intel Core i5 or higher (or equivalent AMD processor) |
| RAM | Minimum 8 GB (16 GB or higher recommended) |
| Graphics | Integrated or dedicated GPU (e.g., Intel HD Graphics, NVIDIA GeForce) |
| Screen Resolution | 1920x1080 (Full HD) or higher for sufficient screen real estate |
| Operating System | Android 5.0 (Lollipop) or higher |
| Storage | Minimum 2 GB of free storage for apps and data |

The hardware specifications could be used as a basis for a contract for the implementation of the system. This therefore should be a full, full description of the whole system. It is mostly used as a basis for system design by the software engineers.

###### 3.3.2 SOFTWARE REQUIREMENTS

The software requirements paper contains the system specs. This is a list of things which the system should do, in contrast from the way in which it should do things. The software requirements are used to base the requirements. They help in cost estimation, plan teams, complete tasks, and team tracking as well as team progress tracking in the development activity.

###### Table 3.2 Software Requirements

| **Component** | **Specification** |
| --- | --- |
| **Operating System** | Windows 10 (64-bit) / macOS / Linux |
| **Development Environment** | Android Studio (Latest Stable Version) |
| **Programming Language** | Kotlin (for Android development) |
| **Android SDK** | Android SDK (Latest version for Android development) |
| **Database** | Room Database (for local storage of grocery items) |
| **Emulator** | Android Emulator (for testing on various Android versions) |
| **Notification Service** | WorkManager / AlarmManager (for scheduling notifications) |

### 3.4 DESIGN OF THE ENTIRE SYSTEM

###### 3.4.1 ACTIVITY DIAGRAM

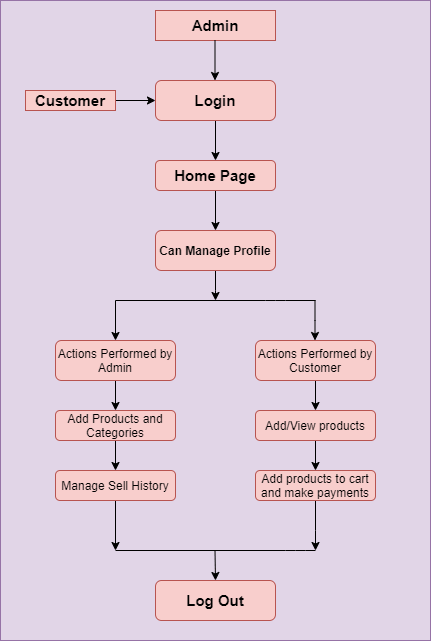
 **User Login**: The user opens the app and logs in. If login is successful, the app navigates to the Home Screen.

 **Enter Grocery Details**: On the Home Screen, the user enters item name, quantity, selects reminder date & time, and repeat frequency.

 **Validation Check**: The app checks if all fields are filled correctly. If not, it prompts the user to complete all fields.

 **Save and Notify**: On valid input, the app saves the grocery item to the Room database and schedules a notification using AlarmManager.

 **View and Navigate**: The user can view saved items on the Display Screen and navigate back to the Home Screen to add more items.



**Fig 3.2: Activity Diagram**

#### 3.4.2 DATA FLOW DIAGRAM

**UserAuthentication**:  
The user logs in to the app, and the system verifies the credentials before granting access to the home screen.

**InputGroceryDetails**:  
The user enters grocery item name, quantity, reminder date/time, and frequency through the home screen interface.

**DataStorage**:  
The entered data is validated and saved into the Room Database for persistent storage.

**NotificationScheduling**:  
The app uses AlarmManager to schedule a local notification based on the user’s input for reminder date and repeat frequency.

**ViewandNavigate**:  
The user can view all saved grocery items in a structured table on the display screen and navigate back to the home screen for further input.

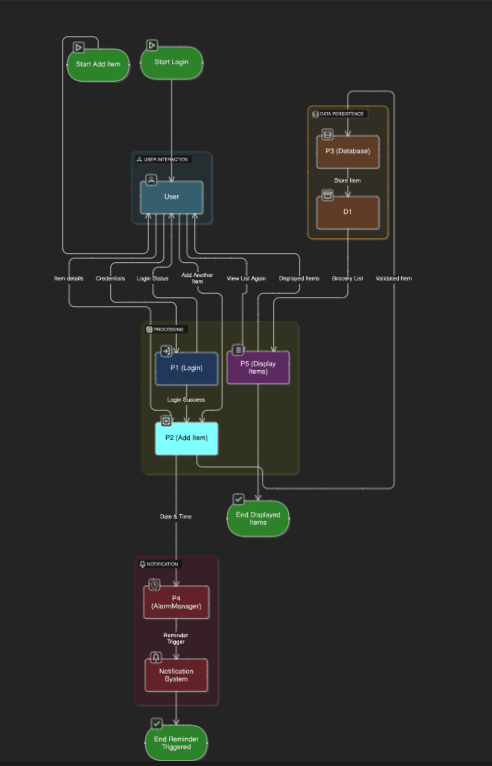


Fig 3.3:Data Flow Diagram

**CHAPTER 4**

**MODULE DESCRIPTION**

The workflow for the proposed system is designed to ensure a structured and efficient process for personalized clothing recommendation system. It consists of the following sequential steps:

## 4.1 SYSTEM ARCHITECTURE

**4.1.1 USER INTERFACE DESIGN**

**Sign In Screen**

**Elements:**

* App logo or name at the top
* Text field: Email/Username
* Text field: Password
* Button: **Sign In**
* Text link: **Forgot Password?**
* Text link: **Sign Up**

**Flow:**User enters credentials and taps **Sign In** → navigates to **Home Screen** on success.

**2. Home Screen (Input Screen)**

**Elements:**

* Header Text: **"WELCOME TO GROCERY MINDER"**
* Text field: **Item Name**
* Text field: **Quantity**
* Button: **Select Date & Time** (opens date and time picker)
* Text view: **Selected Date & Time** (displays chosen date and time)15
* Dropdown menu: **Repeat Frequency**
* Options: Everyday, Weekly, Monthly, Half Yearly, Yearly

Button: **Show Groceries**

**Functionality:**

User fills all fields and clicks **Save** → item saved to Room Database, notification scheduled.

**Show Groceries** navigates to the Display Screen.

**3. Display Screen**

**Elements:**

Header Text: **"GROCERY LIST"**

* Table/List layout showing:
* **Item Name**
* **Quantity**
* **Reminder Date & Time**
* **Repeat Frequency**

Button: **Back** (navigates to Home Screen)

**Functionality:**

Displays all grocery entries saved in the Room Database in a structured tabular form.

Allows user to return to the Home Screen.

**4. Notification**

**Triggered by:** AlarmManager based on scheduled reminder

**Display:**

Title: **"Grocery Reminder"**

Message: **"Don't forget to buy [Item Name]"**

###### 4.1.2 BACK END INFRASTRUCTURE

**Room Database (Local Storage)**

**Purpose**: To store grocery items locally on the user's device.

**Components**:

**Entity**: GroceryItem

**Database Class**:

AppDatabase : RoomDatabase — provides access to DAO

**2. Notification System**

**Purpose**: To alert the user at the scheduled time.

**Components**:

AlarmManager — schedules exact time for reminders

BroadcastReceiver (ReminderReceiver) — triggers the notification

NotificationManager — displays notification on the device

**Process Flow**:

User schedules reminder → AlarmManager sets trigger

On trigger → BroadcastReceiver receives intent → builds and shows notification

**3. Data Scheduling Logic**

Uses: AlarmManager.setExact() for single reminders

SimpleDateFormat and Calendar for time parsing

Ensures notification is only triggered if time is valid and in the future.

**4. App Architecture**

**MVVM (Model-View-ViewModel) Recommended:**

**Model**: Represents data (GroceryItem)

**View**: UI Screens (Composables like Homescreen, DisplayScreen)

**ViewModel** (optional but encouraged): Handles logic, database operations, and notification setup

**5. Navigation**

**Navigation Component** (NavHost, composable)

Manages flow between:

SignInScreen → HomeScreen → DisplayScreen

**6. No Cloud Backend (Offline First)**

This app runs completely offline

All data is stored locally using Room

Notifications are local and don’t depend on internet access

##### 4.2 DATA COLLECTION AND PREPROCESSING

**Data Collection**

Purpose:

To gather relevant user input for setting grocery reminders efficiently and accurately.

Methods of Data Collection:

**User Input via UI Forms**:

**Item Name**: Text input field

**Quantity**: Numeric input field

**Reminder Date & Time**: DatePicker and TimePicker dialogs

**Repeat Frequency**: Dropdown menu (Everyday, Weekly, Monthly, etc.)

**Login Data (Optional)**:

If using authentication, the user’s credentials (email/username) are collected on the Sign-In screen.

**System-Level Data** (Optional for scheduling):

Current system time and date to validate reminder time

Device ID (used internally for pendingIntent uniqueness in AlarmManager)

**Data Processing**

**Steps in Data Processing:**

**Validation**:

Check if all required fields (Item Name, Quantity, Date/Time) are filled.

Ensure the selected date and time are in the future.

Prevent empty or invalid entries using TextUtils and Kotlin conditions.

**Formatting**:

Date and Time are formatted using SimpleDateFormat into a string like dd/MM/yyyy HH:mm.

Quantity may be parsed to an integer or kept as a string, depending on the design.

**Data Storage**:

A GroceryItem object is created with validated and formatted input.

The object is inserted into the **Room Database** using DAO methods.

Notification Scheduling:

Parsed date-time string is converted to milliseconds using SimpleDateFormat and Calendar.

AlarmManager schedules the reminder using a PendingIntent with an associated BroadcastReceiver.

When the time is reached, ReminderReceiver triggers a local notification with the item details.

Display in Table:

All saved items are retrieved from the Room database using getAllGroceryItems().

Displayed in a structured format using Compose UI components in the DisplayScreen.

**CHAPTER 5**

**IMPLEMENTATION AND RESULTS**

##### 5.1 IMPLEMENTATION

The Grocery Minder application is implemented using **Kotlin** in **Android Studio**, leveraging modern Android development practices such as **Jetpack Compose** for UI, **Room Database** for local data persistence, and **AlarmManager** for scheduling notifications. The application starts with a login screen that serves as the authentication entry point for users. Once the user successfully signs in, they are navigated to the home screen where they can enter grocery details including the item name, quantity, reminder date and time, and the desired repeat frequency.

The user interface is designed to be simple and intuitive, allowing users to select a date and time using built-in Android date and time pickers. The entered data is validated to ensure completeness before saving. Upon pressing the "Save" button, the application stores the grocery item in the Room database, a local SQLite-based persistence library that provides an abstraction layer over raw SQL queries. This ensures that the data is retained even if the app is closed.

Additionally, the application uses **AlarmManager** to schedule notifications at the specified time. When the time for the reminder arrives, a **BroadcastReceiver** triggers a notification to remind the user of the grocery item. The application also includes a "Show Groceries" button that navigates to a separate display screen. This screen retrieves all the saved grocery items from the Room database and displays them in a structured tabular format, showing all the fields including item name, quantity, date, and frequency. Each part of the application is modular, ensuring code readability and maintainability. The implementation emphasize responsive UI design, proper data handling and user friendly interactions.

**OUTPUT SCREENSHOTS**

**1. Login Screen**

**Functionality**: Allows users to sign in to access the app.

**Output**: Upon successful sign-in, the user is navigated to the Home Screen.

**Purpose**: Ensures that only authenticated users can manage grocery data.

**2. Home Screen**

**Input Fields**:

**Item Name**

**Quantity**

**Reminder Date & Time**

**Repeat Frequency** (Everyday, Weekly, Monthly, Half-Yearly, Yearly)

**Buttons**:

**Save**: Saves the entered data into the Room database and schedules a notification.

**Show Groceries**: Navigates to the Display Screen to view saved items.

**Output**: A confirmation message like “Saved!” appears. A local notification will be scheduled for the given date and time.

**3. Display Screen**

**Functionality**: Displays all saved grocery items in a **tabular format**.

**Table Columns**:

Item Name

Quantity

Date & Time

Frequency

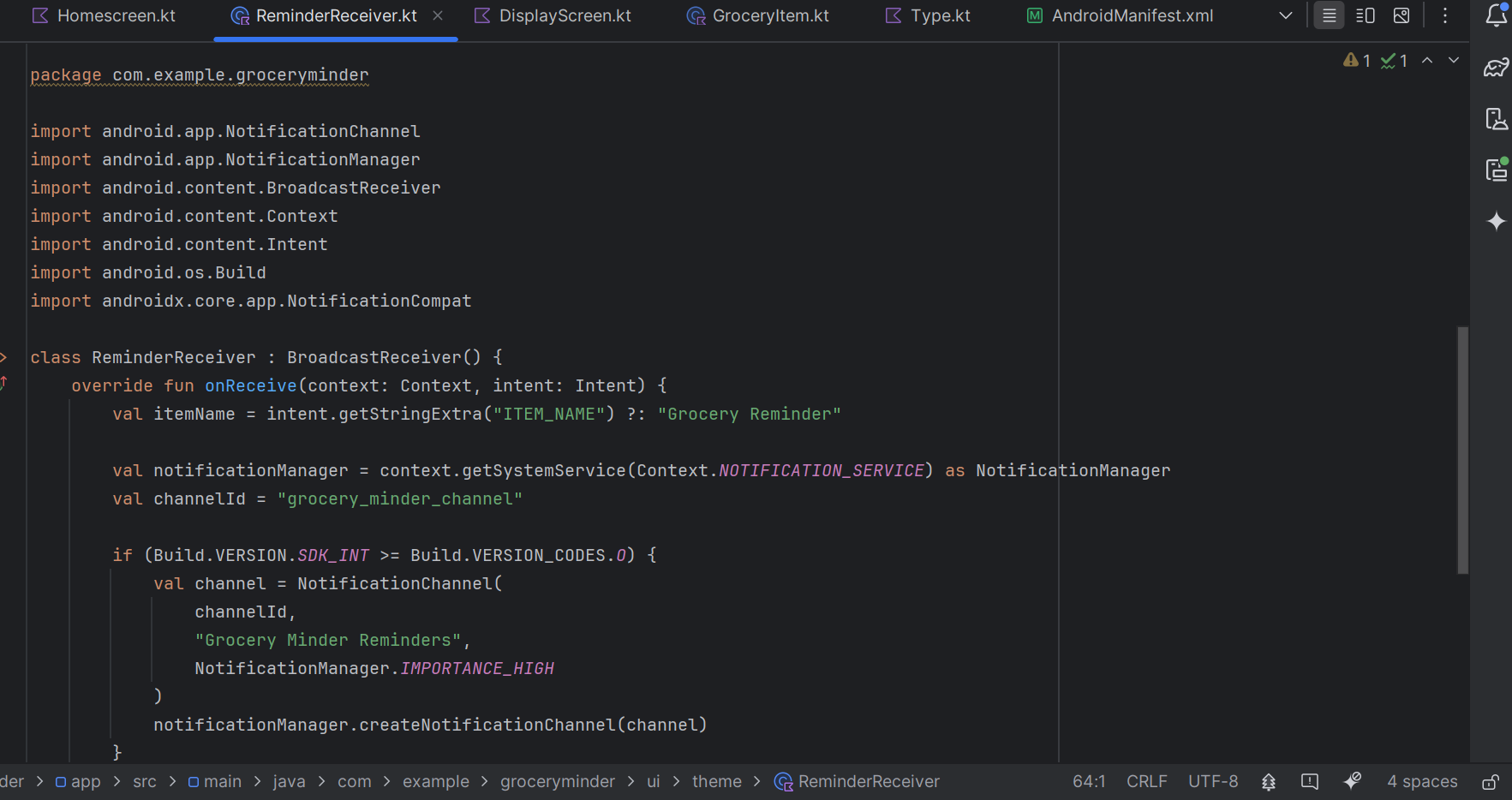
**Back Button**: Returns the user to the Home Screen.

**Output**: The user can view all reminders stored in the Room database with clear details.

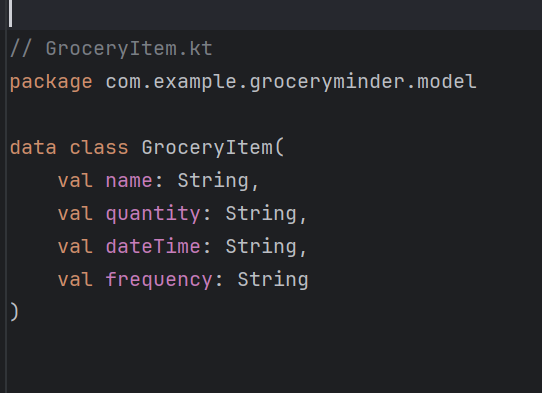
**4. Notifications**

**Scheduled Output**: At the selected date and time, the user receives a **local push notification** (e.g., "Reminder: Buy Milk").

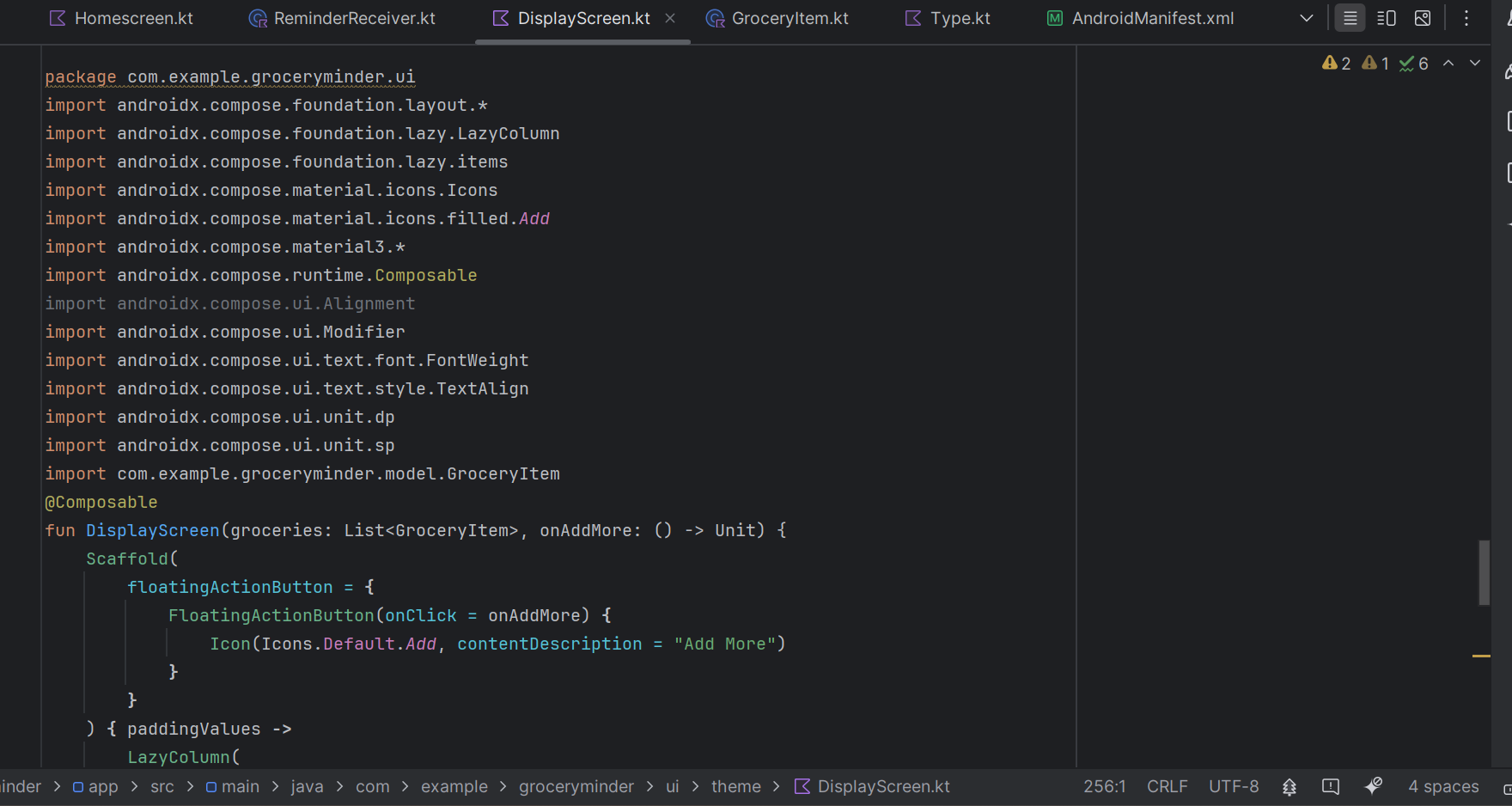
**Purpose**: Ensures timely reminders so users do not forget to buy essential groceries.



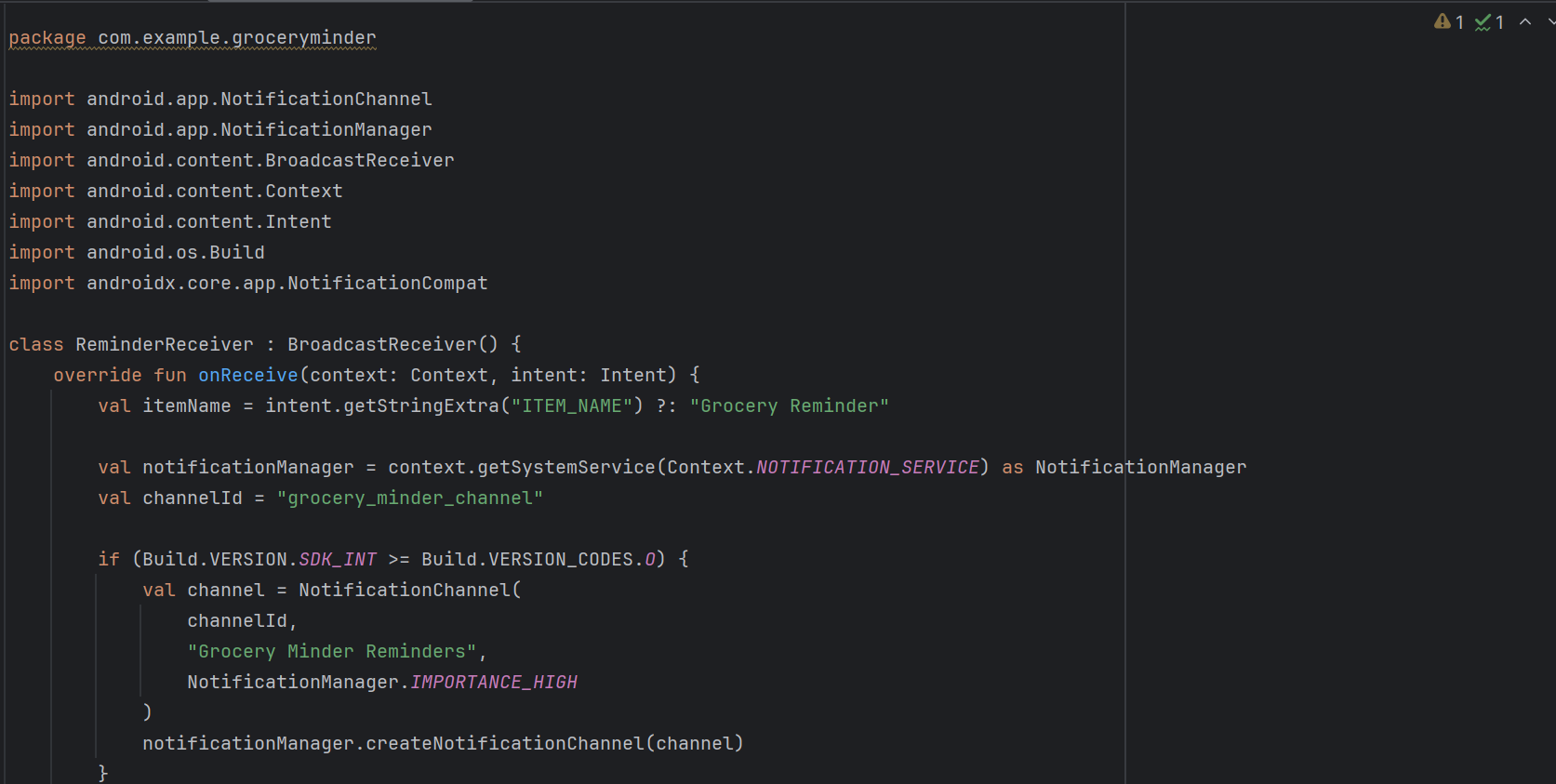
RemainderReciever.kt



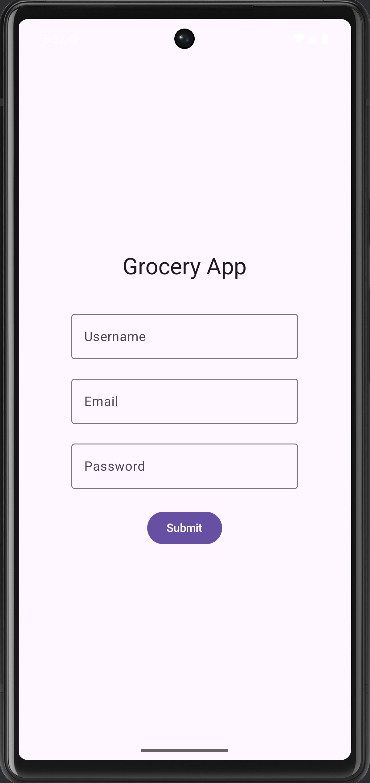
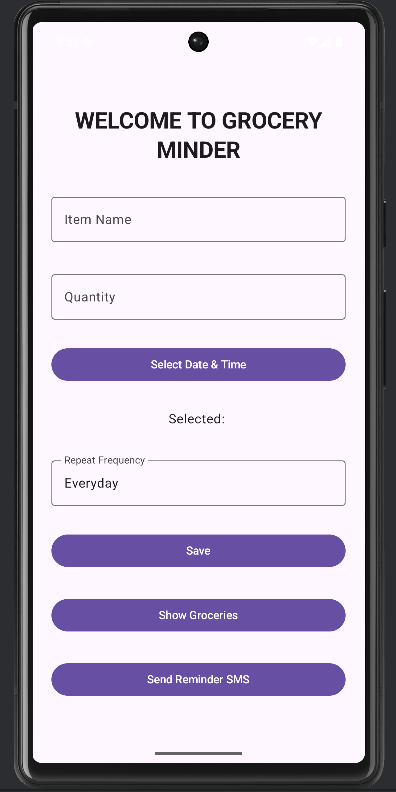
GroceryItem.kt



DisplayScreen.kt



HomeScreen.kt

Signin Page Homescreen Page



DisplayPage

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT**

##### 6.1 CONCLUSION

The Grocery Minder application was developed to simplify the process of managing grocery shopping and to assist users in keeping track of essential items through timely reminders. By integrating a user-friendly interface, Room database for persistent storage, and Android's AlarmManager for notification scheduling, the application ensures that users never miss a grocery item or forget to restock essentials. The use of Kotlin, Jetpack Compose, and modern Android architecture components made the app robust, responsive, and maintainable. Overall, the project meets its goal of improving day-to-day grocery planning efficiency and contributes toward reducing wastage and unnecessary trips to the store.

##### 6.2 FUTURE ENHANCEMENT

To further improve the functionality and user experience of the Grocery Minder app, the following enhancements can be considered:

**Cloud Backup and Sync**

Integrate Firebase or other cloud databases to sync data across multiple devices and allow for remote backups.

**Voice Input for Adding Items**

Enable users to add grocery items using voice commands via Google Assistant or in-app voice recognition.

**Barcode Scanner Integration**

Implement a barcode scanner to allow users to quickly add grocery items by scanning product barcodes.

**Smart Notifications Based on Location**

Use geofencing to send reminders when the user is near a grocery store.

**Shopping List Sharing**

Allow users to share grocery lists with family or friends via email or messaging apps.

**AI-Based Prediction and Recommendations**

Analyze shopping patterns and suggest frequently bought items or recommend restocking times.

**Dark Mode and UI Customization**

Add support for dark mode and allow users to customize the app’s appearance and themes.

**Product Expiry Date Tracking**

Let users track the expiry dates of perishable goods and get alerts before expiry.

**Inventory and Budget Management**

Add features to manage pantry inventory and track spending limits.

**Multi-language Support**

Support regional languages to make the app accessible to a broader audience.

**REFERENCES**

 **“A Context-Aware Reminder System Using Location-Based Services on Android”**International Journal of Computer Applications, 2013  
DOI: 10.5120/10240-5077

 **“An Android Application for Daily Task Management and Reminder System”**  
International Journal of Scientific & Engineering Research, 2015

**“Development of Android Based Reminder Application”**  
International Research Journal of Engineering and Technology (IRJET), 2017  
ISSN: 2395-0056

 **“Design and Implementation of a Smart Notification System in Mobile Devices”**  
Procedia Computer Science, 2015  
DOI: 10.1016/j.procs.2015.08.049

 **“Scheduling of Tasks and Notifications in Android Applications using AlarmManager”**  
International Journal of Advanced Research in Computer and Communication Engineering, 2016

 **“Data Persistence in Mobile Applications using SQLite and Room Persistence Library”**  
International Journal of Recent Technology and Engineering (IJRTE), 2019

 **“A Comparative Study of Local Notification Systems in Android and iOS”**  
International Journal of Computer Science and Mobile Applications, 2018

 **“Smart Grocery List Mobile Application with Expiry and Reminder Feature”**  
IEEE Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA), 2020.

29

 **“Mobile Application Development using Android and Kotlin”**  
International Journal of Computer Applications, 2018

 **“A Study on Effective User Interface Design for Mobile Apps”**  
Journal of Mobile Computing, Communications & Mobile Networks, 2017

 **“Design and Implementation of a Grocery Management System”**  
International Journal of Computer Applications, 2016

 **“User-Centered Mobile Application Design Practices: A Case Study”**  
ACM SIGCHI Conference on Human Factors in Computing Systems, 2018

 **“Enhancing Mobile Notification Systems: A Review of Scheduling Techniques”**  
International Journal of Computer Science and Information Security (IJCSIS), 2015

 **“Real-time Reminder Systems using Alarm Services in Android”**  
International Journal of Computer Sciences and Engineering (IJCSE), 2019

 **Google Developers Documentation on Android Architecture Components (Room, LiveData, ViewModel)**  
URL: <https://developer.android.com/topic/libraries/architecture>