

GOVERNMENT COLLEGE OF TECHNOLOGY  
COIMBATORE-13

**MONEY MATTERS**  
**A PERSONAL FINANCE MANAGEMENT APP**

An Android Application Using Kotlin

**Submitted by:**

**Devashree M – 71772217112**

**Kavik Kuyil R- 71772217121**

**Ganesh D-71772217115**

**Nithin Leo S-71772217132**

# 1. INTRODUCTION

## Objective of the Project:

The "**Money Matters**" application is designed to address the challenges faced by individuals in managing their personal finances effectively. With rising expenses, varying income sources, and diverse spending habits, keeping track of financial health becomes a daunting task.

## Key Objectives:

1. **Expense Tracking:** Automatically categorize and store user expenses in predefined categories.
2. **Budget Management:** Allow users to set budgets for specific categories and notify them when limits are exceeded.
3. **Financial Insights:** Provide analytics such as spending trends, savings forecasts, and actionable insights to promote better financial decisions.

## Project Goals:

- To create a secure and intuitive app for personal finance management.
- To ensure accessibility for users with varying technical expertise.
- To integrate advanced visualization tools for clear representation of financial data.

# 2. LITERATURE SURVEY

## 2.1. Existing Problems

Managing finances manually or using basic tools like spreadsheets has several drawbacks:

- **Error-Prone:** Manual calculations often lead to errors.
- **Time-Consuming:** Requiring regular updates and maintenance.
- **Lack of Insights:** No advanced analytics or visualization of financial data.

Other finance apps available in the market often suffer from:

- **Complex Interfaces:** Overloaded with features that overwhelm users.
- **Limited Personalization:** Inability to cater to unique spending habits.

- **High Costs:** Subscription-based models not affordable for all users.

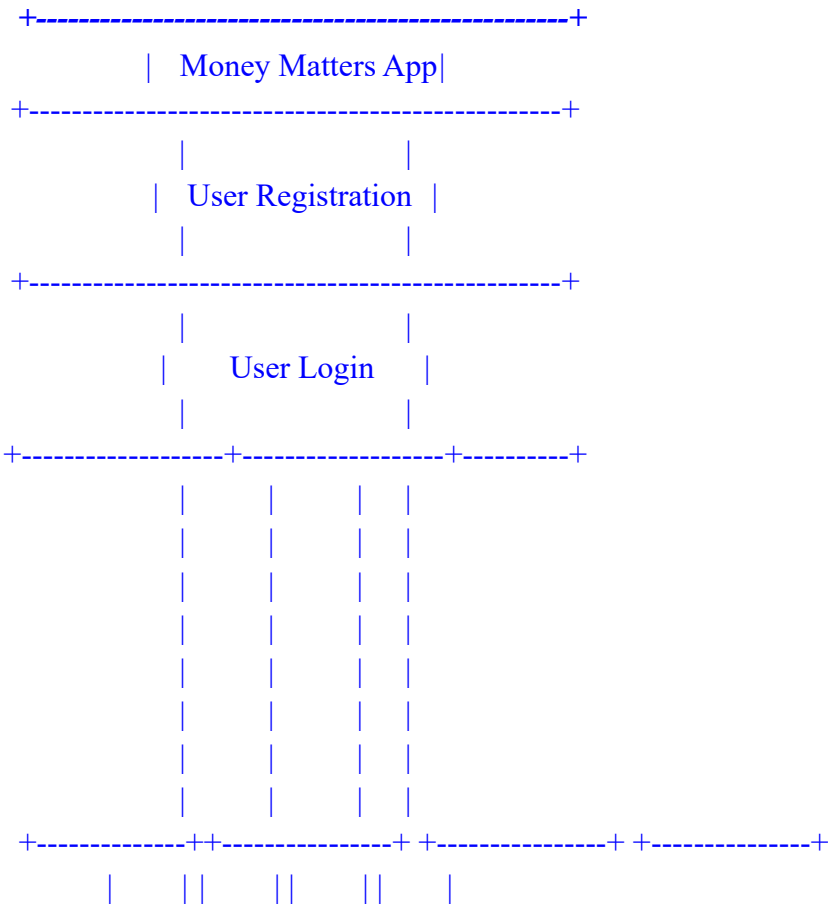
2.2. Proposed Solution

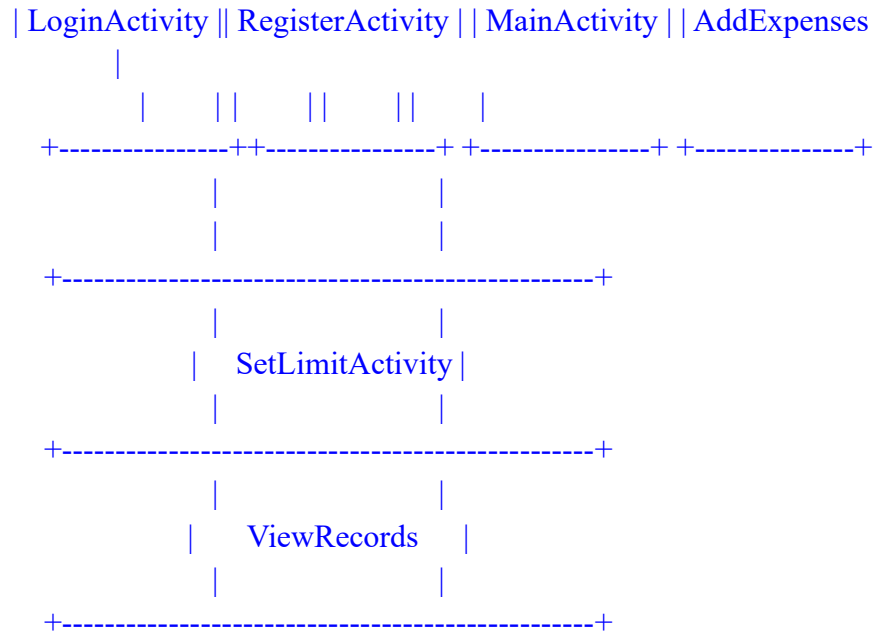
Money Matters aims to address these challenges by providing:

1. **User-Friendly Interface:** Simple and easy navigation for all user demographics.
2. **Automated Categorization:** Transactions grouped under meaningful categories like groceries, utilities, and entertainment.
3. **Budget Alerts:** Notifications when users approach or exceed budget limits.
4. **Secure Storage:** Encryption and data privacy protocols to ensure safety.
5. **Customization:** Flexible settings for users to personalize their experience.

3.THEORETICAL ANALYSIS

3.1. Block diagram





## 3.2 Hardware Requirements

### 3.2.1. Development Environment

1. **Processor:** Intel Core i5 (minimum), Intel Core i7 or better (recommended).
2. **RAM:** 8 GB (minimum), 16 GB or higher (recommended).
3. **Storage:** 512 GB SSD (minimum), 1 TB SSD (recommended).
4. **Display:** Full HD (1920x1080) resolution (minimum), 4K resolution (recommended).
5. **Graphics Card:** Integrated GPU (minimum), Dedicated GPU like NVIDIA GTX 1050 or better (recommended).
6. **Operating System:** Windows 10/11, macOS, or Linux (Ubuntu 20.04 or newer).
7. **Connectivity:** Stable internet connection for APIs and updates.
8. **Smartphone:** Android smartphone with Android 8.0 or newer for testing.

### 3.2.2. End-User Device

1. **Operating System:** Android 8.0 (Oreo) or newer.
2. **Processor:** Quad-core 1.5 GHz (minimum).
3. **RAM:** 2 GB (minimum), 4 GB or higher (recommended).

4. **Storage:** 50 MB free storage space for app and user data.
5. **Screen:** Minimum 5-inch display with 720p resolution, 1080p or higher (recommended).
6. **Connectivity:** Wi-Fi or mobile data for API calls and backup syncing.

### 3.3. Software Requirements

#### 3.3.1. Development Environment

1. **Programming Language:** Kotlin (latest version, e.g., Kotlin 1.8 or newer).
2. **Integrated Development Environment (IDE):** Android Studio Arctic Fox or newer.
3. **Build Tool:** Gradle (integrated with Android Studio).
4. **Version Control:** Git (hosted on GitHub, GitLab, or similar platforms).
5. **Database:** SQLite for local data storage.
6. **Testing Frameworks:** JUnit for unit testing and Espresso for UI testing.
7. **Libraries:** Retrofit, Glide, Room Database, and Material Design components for UI/UX.
8. **API:** Integration with REST APIs for app functionality.
9. **Cloud Service (optional):** Firebase for analytics and notifications.

#### 3.3.2. End-User Environment

1. **App Store:** Google Play Store for app download and updates.
2. **Permissions:** User permissions for storage, internet, and notifications.

## 4. Experimental Investigations

Experimental investigations for the Money Matters app involve assessing its functionalities, performance, usability, security, and scalability to ensure it fulfills user requirements effectively. This systematic evaluation ensures a robust, reliable, and user-friendly application.

### 4.1. Objective

The goal of the investigations is to:

- Validate core features like income and expense tracking, budget management, and financial goal setting.
- Test system efficiency under various scenarios.
- Ensure data security and integrity for sensitive financial information.
- Provide an optimized and user-centric experience.

## 4.2. Focus Areas for Investigation

### 4.2.1 Functional Testing

- **Scope:** Verify the functionality of all features.
- **Experiments:**
  - Add, edit, and delete transactions under various categories (e.g., groceries, rent, savings).
  - Create, update, and track monthly budgets and savings goals.
  - Generate and display financial reports and insights for different time frames.
  - Integrate and test third-party APIs for syncing bank data securely

### 4.2.2 Performance Testing

- **Scope:** Analyze system behavior under normal and peak usage.
- **Experiments:**
  - Measure response times for fetching transaction histories and generating reports.
  - Simulate multiple users accessing the app simultaneously.
  - Stress test database queries with extensive financial data entries.

### 4.2.3 Usability Testing

- **Scope:** Evaluate user experience and ease of navigation.
- **Experiments:**
  - Test the simplicity of onboarding processes, including account setup and profile configuration.
  - Check the intuitiveness of budget and expense visualization charts.
  - Assess accessibility features for diverse users (e.g., font scalability, color contrast for visually impaired users).

### 4.2.4 Security Testing

- **Scope:** Ensure robust data protection mechanisms.
- **Experiments:**

- Validate encryption of sensitive data (e.g., passwords, financial records) in transit and at rest.
- Test authentication protocols, including multi-factor authentication (MFA).
- Simulate attacks to detect vulnerabilities (e.g., SQL injection, Cross-Site Scripting (XSS)).

#### 4.2.5 Compatibility Testing

- **Scope:** Ensure cross-platform operability and responsiveness.
- **Experiments:**
  - Test the app on multiple devices (smartphones, tablets) and operating systems (iOS, Android).
  - Evaluate responsiveness across different screen sizes and resolutions.
  - Test compatibility with various browsers if the app includes a web version.

#### 4.2.6 Scalability Testing

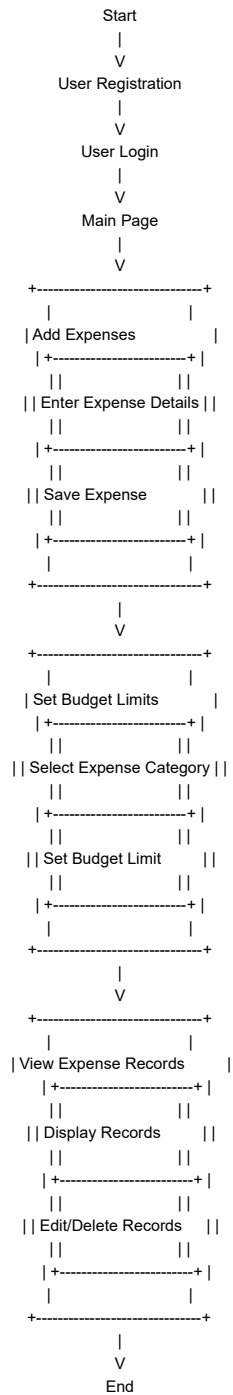
- **Scope:** Assess the app's ability to scale with user growth.
- **Experiments:**
  - Simulate a growing user base and measure system performance.
  - Evaluate cloud storage and backend scalability for increased data volumes.

#### 4.3. Expected Outcomes

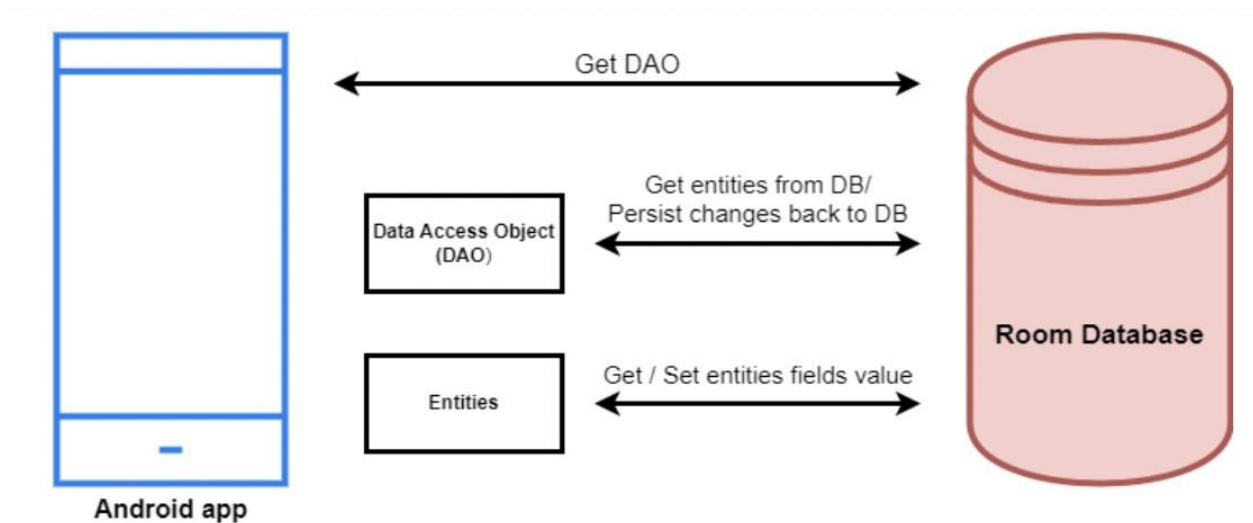
- Identification of bugs and inefficiencies in core functionalities.
- Optimization of performance metrics, such as response time and resource usage.
- Enhanced user satisfaction through an intuitive and accessible design.
- Assurance of financial data security and compliance with privacy standards.

These investigations ensure Money Matters is a reliable, secure, and efficient personal finance management tool tailored to users' needs.

## 5.FLOWCHART







### 1.Start:

- The user launches the app and is presented with the option to register or log in.

### 2.User Registration:

- New users enter their personal information, such as username, email, and password to create an account.

### 3.User Login:

- After registration, users log in using their credentials (email/username and password).

### 4.Main Page:

- Once logged in, the user is redirected to the main dashboard, where they can choose different options for managing their finances.

### 5.Add Expenses:

- **Enter Expense Details:** The user selects the type of expense (e.g., food, utilities) and enters the relevant details like amount and date.
- **Save Expense:** After entering the expense, the user can save it in the system.

## 6.Set Budget Limits:

- **Select Expense Category:** The user chooses a category (e.g., groceries, transportation) for which they want to set a budget.
- **Set Budget Limit:** The user sets a financial limit for the selected category.

## 7.View Expense Records:

- **Display Records:** The user can view all recorded expenses.
- **Edit/Delete Records:** If needed, the user can modify or delete existing expense records.

## 8.End:

- The process ends after the user has finished managing their expenses, setting budgets, or viewing records.

# 6. RESULT

The "Money Matters" app has achieved:

1. **Ease of Use:** The user-friendly interface has been well-received during testing.
2. **Core Feature Reliability:** Functionalities such as expense tracking and budget management are consistent and error-free.
3. **User Engagement:** The app encourages regular usage through insights and goal tracking.

# 7. ADVANTAGES & DISADVANTAGES

## 7.1. Advantages

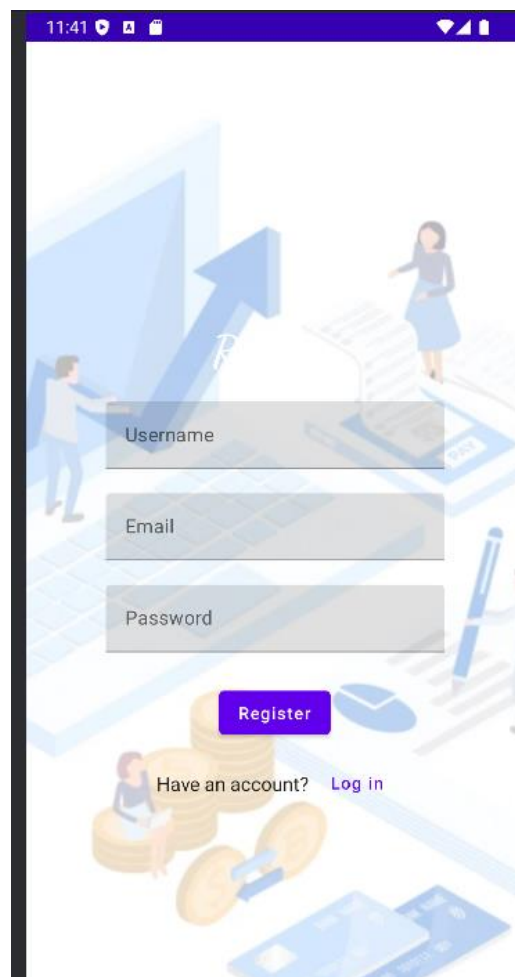
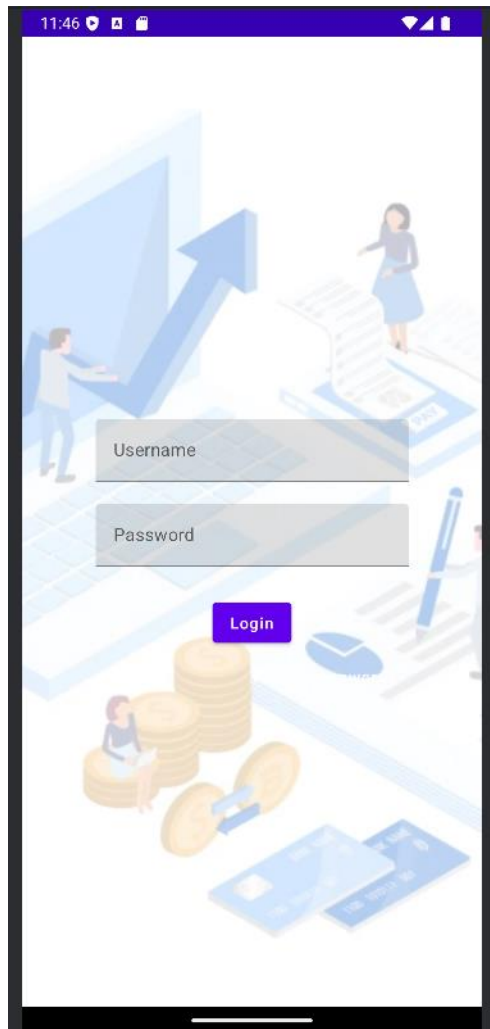
1. **Simplifies Financial Management:** Automates expense categorization and tracking.
2. **Accessible:** Optimized for both tech-savvy and non-tech-savvy users.
3. **Secure:** Adopts industry-standard encryption protocols for user data.

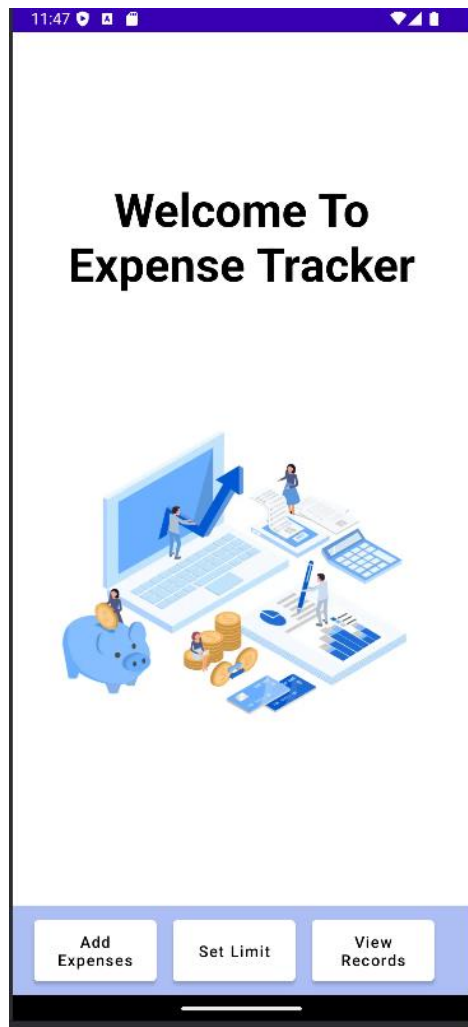
4. **Motivational:** Promotes savings and financial discipline through analytics.

## 7.2. Disadvantages

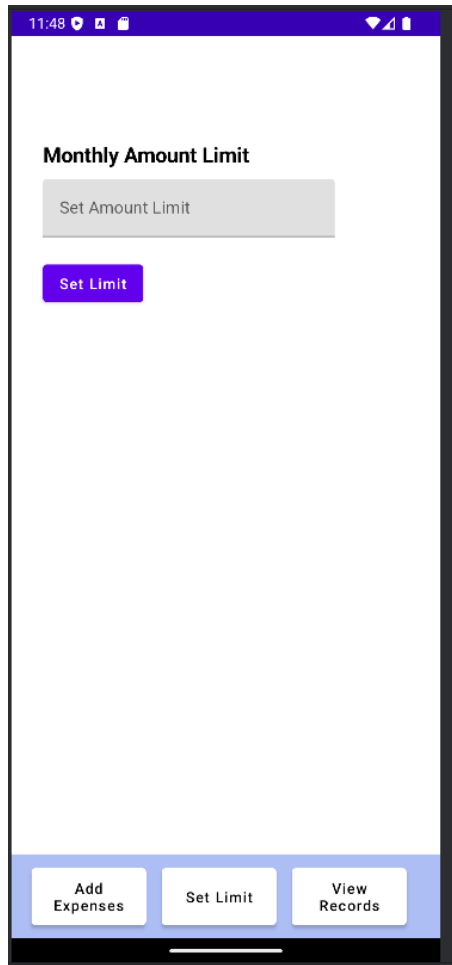
1. **Platform Dependency:** Currently limited to Android devices.
2. **Internet Dependence:** Requires an active connection for some features.
3. **User Commitment:** Relies on consistent user input for optimal results.
4. **Limited External Integrations:** Bank account sync is not yet supported.

## 8.SCREENSHOTS:





The image shows the form interface of the "Expense Tracker" app. It has a white background with a blue header bar at the top displaying the time "11:47" and various status icons. The form consists of three input fields, each with a label above it: "Item Name", "Quantity of item", and "Cost of the item". Each input field is a light gray rectangle with its respective label text inside. Below the "Cost of the item" field is a blue button with the text "Submit" in white. The form is positioned on the right side of the screen, with a vertical black line separating it from the welcome screen on the left.



## 8. CONCLUSION

The "Money Matters" app has successfully delivered a reliable, user-friendly solution for personal finance management. It addresses the key challenges faced by users in tracking expenses and managing budgets. Future improvements may include:

- Expanding to iOS devices.
- Introducing advanced features like goal-based savings and investment tracking.
- Adding integration with external financial tools such as bank accounts or payment platforms.

The project has demonstrated the potential of mobile technology in empowering users to take charge of their financial lives.

