**A picture containing text

Description automatically generated**

**School of Computing and Mathematical Sciences**

**CO7201 Individual Project**

**Preliminary Report**

**Finance Tracker**

**Vivek Shah**

**vps5@student.le.ac.uk**

**Student Id - 239062179**

**Project Supervisor: Dr. Babajide Afeni**

**Principal Marker: Dr. Craig Bower**

**Word Count: 2088**

**Submission Date – 28/02/2025**

**DECLARATION**

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people’s work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people’s work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: Vivek Shah

Date: 28/02/2025

Contents

[**1.** **Aims and Objectives** 3](#_Toc65510906)

[**2.** **Requirements** 4](#_Toc65510907)

[**3.** **Technical Specification** 5](#_Toc65510908)

[**4.** **Requirements Evaluation Plan** 6](#_Toc65510909)

[**5.** **Background Research and Reading list** 7](#_Toc65510910)

[**6.** **Time-plan and Risk Plan** 8](#_Toc65510911)

[**7.** **References** 11](#_Toc65510912)

# **Aims and Objectives**

**Aims and Objectives**

Managing personal finances effectively is crucial in today’s fast-paced world, where individuals often struggle to track their income, expenses, and savings. Many existing finance tracking applications are either too complex, require subscriptions, or lack essential features like multi-currency support and offline accessibility. This project aims to develop a user-friendly **Finance Tracker App** that provides a seamless experience for budgeting, tracking transactions, and visualizing financial data.

The project is particularly relevant due to:

* **Growing reliance on digital solutions** – Mobile apps have become the preferred way to manage finances.
* **Need for better financial awareness** – Many people struggle to track expenses, leading to overspending and poor savings habits.
* **Accessibility and security concerns** – Many apps lack offline access or have privacy concerns due to excessive data collection.
* **Customization and flexibility** – A finance tracker tailored to user preferences (multi-currency, filtering, data export) can improve financial management.

**Aims**

The primary goal of this project is to develop a secure, intuitive, and feature-rich mobile application that helps users manage their personal finances efficiently. The app will provide expense tracking, budgeting tools, financial data visualization, and multi-device synchronization to enhance user experience and financial awareness.

**Main Challenges**

1. **Ensuring Data Security & Privacy** – Handling sensitive financial data securely while allowing cloud backup.
2. **Optimizing Performance** – Managing database operations and visualizations efficiently.
3. **Implementing Multi-Currency Support** – Accurately handling exchange rates and conversions.
4. **User Experience Design** – Creating an intuitive UI that balances simplicity with feature richness.
5. **Asynchronous Data Handling** – Managing Firebase synchronization and offline data processing effectively.

By addressing these challenges this project aims to deliver a robust and practical financial management solution tailored to user needs.

# **Requirements**

**1. Essential Requirements**

These are the core features that the Finance Tracker App must have to fulfill its primary purpose.

* **User Authentication:** Users must be able to register and log in using email/password authentication.
* **Google Sign-In:** Support for Google authentication to provide an easier sign-in process.
* **Expense and Income Management:** Users should be able to add, edit, and delete expenses and income with details such as amount, category, and date.
* **Budget Management:** Implement budget tracking, allowing users to set and monitor monthly budgets.
* **Local Data Storage:** Store financial data locally using Room Database to enable offline access.
* **Data Visualization:** Provide charts and graphs to help users analyze their income and expenses.
* **Secure Data Storage:** Ensure proper encryption and authentication to protect sensitive financial data.

**2. Desirable Requirements**

These features will enhance usability and provide additional functionality to improve the user experience.

* **Cloud Synchronization:** Implement Firebase integration to allow users to sync financial data across multiple devices.
* **Notifications & Reminders:** Send alerts to notify users about budget limits and upcoming expenses.
* **Transaction Filtering & Searching:** Allow users to filter and search transactions by category, date, or amount for quick access.
* **Multi-Currency Support:** Integrate a currency conversion feature to support transactions in different currencies.
* **User-Friendly UI:** Design an intuitive and visually appealing interface to ensure smooth navigation and usability.

**3. Optional Features**

These features are not mandatory but would provide additional benefits and improve the overall experience.

* **Data Export**: Enable users to export financial records in CSV and PDF formats for external use.
* **Receipt Scanning (OCR)**: Implement Optical Character Recognition (OCR) to allow users to scan receipts and auto-fill expense details.
* **Fingerprint Authentication:** Allow biometric login for enhanced security and quick access.
* **Multiple Account Support**: Let users manage different financial accounts (e.g., personal and business).
* **Dark Mode**: Introduce a dark mode theme for better accessibility and user preference.

# **Technical Specification**

|  |  |  |
| --- | --- | --- |
| **Category** | **Technology/Tool** | **Reason for Selection** |
| Programming Language | Kotlin | Official language for Android, modern and concise |
| UI Framework | Jetpack Compose | Declarative UI, efficient, and recommended by Google |
| Database | Room Database | Provides an abstraction over SQLite, supports offline access |
| Authentication | Firebase Authentication | Secure and easy-to-implement authentication (Email/Google) |
| Cloud Storage | Firebase Firestore | Real-time synchronization and cloud backup |
| State Management | ViewModel & LiveData | Manages UI-related data efficiently |
| Networking | Retrofit | Simplifies API calls (e.g., currency conversion API) |
| Dependency Injection | Hilt (Dagger) / Koin | Improves modularity and testability |
| Background Tasks | Coroutines & Flow | Efficient async programming, recommended for Kotlin |
| Charts & Graphs | MPAndroidChart | Popular library for interactive data visualization |
| Multi-Currency Support | Exchange Rate API (e.g., OpenExchangeRates, Fixer.io) | Fetches real-time exchange rates for currency conversion |
| Data Export | Apache POI / OpenCSV | Enables CSV and PDF export for financial reports |
| Security | EncryptedSharedPreferences / SQLCipher | Ensures data security with encryption |
| Push Notifications | Firebase Cloud Messaging (FCM) | Sends budget reminders and transaction alerts |
|  |  |  |

The selected technology stack is based on my current knowledge and learning in Android development, aligning with industry best practices and recommended tools. However, it is subject to change if advancements in technology introduce more efficient solutions or if any of the chosen technologies become deprecated. Any modifications will be carefully evaluated to ensure they align with the project's objectives while maintaining optimal performance, security, and scalability.

# **Requirements Evaluation Plan**

|  |  |
| --- | --- |
| **Criteria** | **Evaluation Metrics** |
| Functionality | Ensure all essential features (authentication, transaction tracking, budget management, data visualization) work as intended. |
| Usability | Assess ease of use, navigation, and user experience using user feedback and surveys. |
| Performance | Measure app speed, responsiveness, and database efficiency (e.g., transaction retrieval time < 500ms). |
| Security | Ensure secure authentication, encrypted data storage, and protection against unauthorized access. |
| Reliability | Test system stability under different conditions (e.g., offline mode, cloud sync delays). |
| Compatibility | Verify app runs smoothly on different Android versions and device sizes. |
| Data Accuracy | Ensure correct budget calculations, transaction records, and currency conversions. |
| Scalability | Evaluate system behavior with an increasing number of transactions and users. |

**Stakeholders Involved in Evaluation**

* **End Users**: Gather feedback from test users to assess usability and functionality.
* **Supervisors & Reviewers**: Ensure project meets academic and technical expectations.
* **Developers (Self-Review)**: Identify bugs and areas for improvement through self-testing.

# **Background Research and Reading list**

To develop a Finance Tracker App, understanding android mobile development best practices, security protocols, and UI/UX design principles is crucial. This section outlines following

**1. Background Research Areas**

**A. Mobile App Development (Android)**

* **Jetpack Compose**: Modern UI framework for Android development.
* **Room Database**: Local data storage solution using SQLite abstraction.
* **Kotlin Coroutines & Flow**: Asynchronous programming techniques for smooth app performance.
* **Firebase Authentication & Firestore**: Secure login and cloud storage solutions.

**B. Financial Management Concepts**

* **Budgeting principles**: Understanding personal finance and expense tracking methodologies.
* **Multi-currency support**: Implementing exchange rate conversion using APIs.
* **Data visualization in finance**: How charts and graphs improve financial decision-making.

**C. Security & Data Privacy**

* **Encryption techniques**: Protecting sensitive financial data (AES encryption, SQLCipher).
* **OAuth and Firebase security rules**: Secure authentication methods for mobile applications.

**D. UI/UX Design for Finance Apps**

* **Material Design principles**: Best practices for an intuitive and visually appealing UI.
* **Accessibility considerations**: Implementing features like dark mode, high contrast UI, and biometric login.
* **User engagement strategies**: How notifications and reminders enhance financial awareness.

**Online Resources & Documentation**

1. **Official Android Documentation** – [developer.android.com](https://developer.android.com/)  
   Detailed documentation on Jetpack Compose, Room Database, and other essential components.
2. **Firebase Documentation** – [firebase.google.com/docs](https://firebase.google.com/docs)  
   Guidelines for authentication, Firestore integration, and cloud storage security.
3. **Kotlin Coroutines Guide** – [kotlinlang.org/docs/coroutines-overview.html](https://kotlinlang.org/docs/coroutines-overview.html)  
   Understanding coroutines for efficient background processing.
4. **Material Design Guidelines** – [material.io/design](https://material.io/design)  
   Best practices for building visually appealing and user-friendly Android applications.
5. **Room Database Guide** – [developer.android.com/training/data-storage/room](https://developer.android.com/training/data-storage/room)  
   Detailed guide on implementing local storage for offline financial data management.
6. **YouTube Channel** - Lackner, P. (n.d.). Philipp Lackner [YouTube channel]. YouTube. <https://www.youtube.com/@PhilippLackner>

# **Time-plan and Risk Plan**

**Time Plan**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Weeks** | | | | | | | | | | | | | |
|  |  | | | | | | | | | | | | | |
| **Tasks** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** |
| Project Proposal & Research |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UI/UX Design  (Wireframes, Prototyping) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Authentication  (Email, Google Sign-In) [Essential] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Core Features (Add, Edit, Delete Transactions) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Backend Setup  (Room Database, Firebase) [Essential] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Budget Management Feature [Essential] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Data Visualization  (Charts, Graphs) [Essential] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Data Security & Encryption  [Essential] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Notifications & Reminders [Essential] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Multi-Currency & API Integration [Desirable] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Data Export (CSV, PDF) [Desirable] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| OCR Receipt Scanning  [Optional] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Fingerprint Authentication  [Optional] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Dark Mode  [Optional] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Multiple Accounts Support  [Optional] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| User Testing & Feedback Collection |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Documentation & Report Writing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Final Refinements |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Viva Preparation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Presentation Preparation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Final Submission & Viva |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Milestone** | **Description** | **Deadline** |
| Project Proposal & Research | Define project scope, requirements, and conduct background research. | Week 2 |
| UI/UX Wireframes & Backend Setup | Create UI wireframes, set up Firebase authentication and Room Database. | Week 3 |
| User Login/Authentication | Registration and Firebase Authentication for user login, | Week 4 |
| Core Feature Implementation | Implement adding, editing, and deleting transactions. | Week 6 |
| Backend Setup | Implement the Room Database for offline storage and cloud sync functionalities. | Week 6 |
| Budget Management Feature | Develop functionality to allow users to set, update, and track monthly budgets.Ensure budget alerts and limits are implemented. | Week 7 |
| Data Visualization | Create interactive charts and graphs to display income and expense trends.  Utilize libraries like MPAndroidChart for graphical representation. | Week 8 |
| Data Security & Encryption | Implement authentication security, encrypt sensitive financial data,  and ensure secure storage practices to protect user information. | Week 8 |
| Notifications & Reminders | Set up push notifications to remind users about their budget limits, upcoming bills, or expense tracking habits. | Week 9 |
| User Testing & Feedback Collection | Conduct usability testing by gathering feedback from potential users.  Make necessary refinements based on their experience and input. | Week 11 |
| Documentation & Report Writing | Start writing the project report, documenting system architecture, design choices,  implementation details, and test results. | Week 12 |
| Final Refinements | Address any remaining bugs or issues based on testing and feedback.  Optimize performance and enhance UI/UX. | Week 12 |
| Viva Preparation | Prepare responses for the viva, anticipate possible questions,  and practice explaining the project’s technical aspects. | Week 12 |
| Presentation Preparation | Create presentation slides summarizing the project’s objectives, implementation, challenges, and outcomes. Practice delivery. | Week 12 |
| Final Submission & Viva | Submit the final report, codebase, and presentation. Participate in the viva and defend the project. | Week 14 |

**Risk Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risks** | **Likelihood** | **Impact** | **Mitigation Strategy** |
| Technical Complexity  (Multiple APIs, authentication, database integration) | Medium | High | Break tasks into smaller parts, use official documentation  and community forums for troubleshooting. |
| Time Management Issues | High | High | Stick to a structured schedule, prioritize essential features first,  and avoid adding unnecessary features |
| Security Vulnerabilities  (Weak authentication, data leaks) | Medium | High | Implement Firebase security rules, encrypt sensitive data, and follow OWASP guidelines. |
| Performance Issues  (Slow data processing, app crashes) | Medium | Medium | Optimize Room Database queries, test on multiple devices,  and use background processing (Coroutines). |
| Usability & Adoption Challenges | Low | High | Conduct user testing and iterate based on feedback to  ensure a user-friendly experience. |
| API Downtime  (Currency Conversion, Firebase) | Medium | Medium | Implement offline caching and fallback mechanisms for API failures. |
| Bugs & Unexpected Errors | High | Medium | Perform continuous debugging, logging,  and testing throughout the development cycle. |
| Data Loss Risks | Low | High | Use Firebase cloud sync and implement local database backup mechanisms. |
| Device Compatibility Issues | Medium | Medium | Test app on multiple Android versions and screen sizes. |
| Last-Minute Changes or Delays | Medium | High | Keep a buffer time in the schedule, maintain version control (Git). |

# **References**

**Research Papers & Journal Articles**

1. Oswal, S., & Koul, S. (2013). Big data analytic and visualization on mobile devices. In *Proc. Nat. Conf. New Horizons IT-NCNHIT* (p. 223).

**Online Resources & Documentation**

[1]Google Developers, *Official Android Documentation*, Available: [https://developer.android.com](https://developer.android.com/).

[2] Google Firebase, *Firebase Documentation*, Available: <https://firebase.google.com/docs>.

[3] JetBrains, *Kotlin Coroutines Guide*, Available: <https://kotlinlang.org/docs/coroutines-overview.html>.

[4] Google Material Design, *Material Design Guidelines*, Available: <https://material.io/design>. [Accessed: Day-Month-Year].

[5] Google Developers, *Room Database Guide*, Available: <https://developer.android.com/training/data-storage/room>.

[6] P. Lackner, *Philipp Lackner [YouTube Channel]*, YouTube. Available: <https://www.youtube.com/@PhilippLackner>.