



# Team C: Financial Management App

Android and iOS

Jingjing Yang

Razib Hasan

Xiaosi Huang

5G00FT06-3004 Software Project  
Autumn 2024

Bachelor's Degree Programme in Software Engineering

## CONTENTS

1	Sprint 1 Report.....	3
1.1	Achievements .....	3
1.2	Challenges.....	3
1.3	Solutions and Adjustments.....	3
1.4	Team Learning and Growth .....	4
1.5	Goals Next .....	4
2	Sprint 2 Report.....	5
2.1	Achievements .....	5
2.2	Adjustments .....	6
2.3	Goals Next.....	7
2.4	Agreement and commitment.....	8
3	Sprint 3 Report.....	9
3.1	Accomplishments .....	9
3.2	Goals Next .....	10
3.3	Discussion.....	10
4	Sprint 4 Report.....	13
4.1	Key Accomplishments .....	13
4.2	Technical Accomplishments:.....	15
4.3	Next Goals .....	15
5	Sprint 5 Report.....	17
5.1	Challenges and Lessons Learned .....	17
5.1.1	Initial Planning and Adjustments .....	17
5.1.2	Turning Challenges into Opportunities .....	17
5.1.3	Development Process and Key Takeaways .....	17
5.2	Final Reflections .....	18
	APPENDICES.....	19
	Appendix 1. UI version 1 demo.....	19
	Appendix 2. Summary of Tests and Expected Results .....	24

## **1 Sprint 1 Report**

### **1.1 Achievements**

In Sprint 1, we successfully completed the following tasks:

- 1) User Registration and Login Pages: We implemented user registration and login functionality using Firebase Authentication. Users can register and log in through Firebase.
- 2) Page Navigation: After successful login, users are directed to the receipt list page. On this page, the left floating button navigates to the chart page where users can visualize their expenses, while the right button allows navigation to the "Add Receipt" page, where users can manually add a receipt or scan a receipt to add it.
- 3) Backend Integration: We integrated Firebase as the backend and successfully stored user and receipt data in the Firestore database. We ensured that Firebase Authentication and Firestore collections work properly.
- 4) Technology Validation: Throughout the development process, we validated the usability of several new technologies, including Firebase Authentication (Auth), Text Extraction, Firestore Collections, and Chart visualizations.

### **1.2 Challenges**

During this phase, we encountered several challenges:

- 1) IDE Configuration and Emulator Issues: We faced some issues with IDE configuration and running the app on the emulator. Specifically, the emulator couldn't test the receipt scanning functionality, which was a significant roadblock.
- 2) iOS Platform Testing: To resolve the emulator issue, we learned how to test the app on an iPhone. However, we faced difficulties configuring the OCR (Optical Character Recognition) functionality on iOS. We spent two days debugging this issue but were unable to resolve it.

### **1.3 Solutions and Adjustments**

To avoid the entire team being blocked by the OCR issue, we decided to push this task to the next sprint. This adjustment allows the rest of the team to continue working on other tasks, ensuring progress is made across the board.

### **1.4 Team Learning and Growth**

During this sprint, the team gained valuable experience working with new technologies such as Firebase, Flutter, and iOS development. In particular, learning how to configure and test the app on different platforms has been a significant achievement. Despite some challenges, the team has grown through problem-solving and technical exploration.

### **1.5 Goals Next**

In the upcoming sprint, our goals are as follows:

- 1) **Optimize Existing Pages:** We will work on improving the UI/UX, enhancing the performance of data handling, and optimizing the responsiveness of the current pages.
- 2) **Resolve the OCR Issue:** We plan to tackle the iOS OCR configuration issue in the next sprint, ensuring the scanning functionality works as expected.
- 3) **Feature Expansion:** We will continue to develop and expand the app's features, including error handling improvements, further testing, and refining page navigation.

## 2 Sprint 2 Report

### 2.1 Achievements

The primary focus of Sprint 2 was to restructure the Firebase database to better align the relationships between categories and receipts, improve the user interface, enhance data visualization features, and implement key functionalities such as sorting, filtering, and receipt management.

#### 1) Firebase Database Restructuring:

In this sprint, we restructured our Firebase Firestore database to improve how categories and receipts are linked. This adjustment was necessary to:

- Ensure that each receipt is correctly associated with a specific category.
- Optimize queries for retrieving receipts based on categories, which improves app performance when filtering and displaying data by category.

This change simplified the receipt management process by ensuring data consistency and making it easier to perform filtering and sorting operations. It also highlights the importance of early determination of the backend data structure. If not planned early, such changes can become time-consuming later on.

#### 2) User Interface Improvements:

We made significant improvements to the user interface:

- Enhanced the overall design for better usability and a more intuitive layout.
- Streamlined navigation across different sections of the app, fixing bugs that previously caused issues when moving between views.
- Improved responsiveness across multiple device screen sizes, ensuring a seamless user experience.

#### 3) Expense Analysis: Bar and Pie Charts:

To provide users with insightful expense analysis, we integrated data visualization features:

- Pie Charts: Providing a breakdown of spending by category, giving users a quick overview of how their expenditures are distributed.
- Bar Charts: Displaying expenses over time, helping users track their spending patterns by day, week, or month.

These charts enable users to visualize their spending habits and better manage their finances.

#### 4) Sorting and Filtering Functionality:

To enhance the user experience with the receipt list, we implemented:

- **Sorting Options:** Users can now sort receipts by date, amount, or category, allowing for a more organized display of receipts.
- **Filtering Options:** We added filters so that users can easily view receipts based on specific criteria such as date ranges or categories. This makes it easier for users to locate specific receipts quickly.

#### 5) Receipt Management: Edit and Delete:

We added functionalities to allow users to:

- **Edit Receipts:** Users can now update details such as amount, category, or date of a receipt.
- **Delete Receipts:** Unwanted receipts can now be easily removed from the database.

These features improve the flexibility and accuracy of the app, allowing users to maintain a clean and organized record of their receipts.

#### 6) Improved Period Setting Widget:

We enhanced the period setting widget to provide more flexible date range selection options. Users can now easily check their expenses for predefined periods such as:

- One Week
- One Month
- One Season (quarter)
- One Year

This improvement simplifies the process of analyzing expenses over different time periods, making it easier for users to track their spending habits.

## 2.2 Adjustments

### 1) Early Determination of Project Structure:

It's important to note the value of early determination of the project's structure, including decisions related to backend services, screens, and components to ensure effective reuse. Establishing project conventions early, such as refactoring practices and naming

conventions, also helps maintain consistency throughout development and makes future changes more manageable.

## 2) Testing and Issue Resolution:

During this sprint, we also conducted tests across all features and aspects of the application. We identified and resolved various issues to improve functionality and overall performance. These adjustments ensured that each component works as expected and that the app provides a smooth experience for users.

3) During Sprint 2, we also decided to drop the automatic matching of merchants to corresponding receipt categories, as it proved to be more complex than anticipated. Different regions have varying merchant categorizations, and stores like supermarkets sell items across multiple categories. Since categorizing receipts manually is relatively simple for users, we opted to focus on other features. As a result, we completed Sprint 3's planned functionality of chart analysis earlier than expected, delivering bar and pie charts for expense tracking and visualization.

4) Sprint 2 brought significant improvements to both the backend (Firebase database restructuring) and the frontend (user interface and features). These updates provide a more intuitive and user-friendly experience, allowing users to better manage their receipts and analyze their spending patterns. Moving forward, our focus will be on expanding the functionality of the app, refining the user experience further, and ensuring the stability of these new features.

## 2.3 Goals Next

For Sprint 3, our goal is to improve the styling and make the app look more modern. Although we now have most of the features needed for an MVP, to release it, we need to ensure the app has a user-friendly style. This includes improving elements such as:

- 1) The display of text across the app.
- 2) Enhancing the pop-up selection interfaces.
- 3) Ensuring clarity of page contents and layout.
- 4) Providing clear and intuitive functionality for the user.

The goal is to create a polished design that users can navigate easily and enjoyably.

In Sprint 3, in addition to improving the styling for a more modern and user-friendly look, we will introduce a new feature: budget setting. This will allow users to set spending limits and monitor their expenses against the budget, helping them manage their finances more effectively.

## **2.4 Agreement and commitment**

Resolve the OCR feature!



### 3 Sprint 3 Report

#### 3.1 Accomplishments

During Sprint 3, we achieved significant progress in enhancing user functionality and optimizing backend processes in our application. Here's a summary of our key accomplishments:

1) Category-Specific Budget Settings

We added the capability for users to set budgets specific to each category. This feature allows users to track their spending more granularly, helping them manage their finances effectively within predefined limits for different spending areas.

2) Monthly Summary Page

A new summary page was implemented, displaying monthly spending versus the allocated budgets for each category. This feature provides users with a clear overview of their financial status, showing them how closely they are adhering to their set budgets each month.

3) OCR Issue Resolution

To improve the accuracy and reliability of text extraction from receipts, we resolved previous OCR challenges by switching to a different library from Firebase. This transition resulted in smoother and more accurate text recognition, which is critical for automatically filling receipt information.

4) Automatic Receipt Data Filling

We implemented an algorithm for autofilling new receipts based on extracted text from images. Currently, this feature supports English and Finnish, and processes currencies in USD and EUR. This automation simplifies the user experience by reducing the manual entry of receipt data, allowing for faster and more convenient expense tracking.

5) UI Design Update

In this sprint, we also found a suitable Figma template to streamline our UI design process. This template will serve as a foundation for creating a visually cohesive and user-friendly interface, aligning with our goal of enhancing the user experience.

Overall, Sprint 3 saw the successful completion of several important features and technical improvements, bringing us closer to our project goals.

### 3.2 Goals Next

In Sprint 4, our focus will be on (Vote Some!) the following updates and new features:

a. UI Update

Refreshing the interface design according to the agreed Figma template.

b. Multiple Account Support

Enabling users to manage shared finances across multiple accounts, such as for family members.

c. Data Export Functionality

Allowing users to export expense data for backup and sharing purposes.

d. Trend Analysis

Displaying growth trends for each spending category, helping users track changes in different expense areas.

e. Enhanced Search Functionality

Improving search capabilities to enable users to filter expense records by various criteria.

f. Customizable Background Colors

Expanding category customization options by allowing users to set background colors, in addition to the existing icon customization.

g. Total Budget Setting

Adding a total budget feature to complement the existing category budget settings, helping users monitor overall spending limits.

These updates will further enhance the app's functionality and provide users with more control over their financial management.

### 3.3 Discussion

- 1) Is it acceptable if we don't strictly follow the sprint plan and instead focus on implementing only one advanced feature in each sprint?
  - a. We feel that concentrating on a single high-impact feature each sprint could improve our project quality. Would this approach still align with our project requirements?
- 2) How important is the UI design for the evaluation criteria of this project?
  - a. Since we are focusing on implementing a Figma-based UI template, we'd like to know how much weight UI design carries. Are there

aspects of the design that we can prioritize over others to meet the evaluation criteria?

3) Which aspects are less critical to project success?

- a. We want to ensure our resources are allocated efficiently. Are there areas, like certain backend optimizations or non-essential UI elements, that might carry less weight in the final evaluation?

4) Key documents and deliverables for the final submission of our software project:

a. User Manual

A comprehensive guide detailing how users can navigate and use the application, covering key features and troubleshooting tips.

b. Code in a ZIP File

The complete source code organized and compressed into a ZIP file for easy submission and review.

c. Sprint Report

A detailed report covering each sprint, including our progress, any challenges faced, and how we addressed them. This should also highlight key milestones and completed features. **Should we maintain the existing format, or is a standardized format needed for all reports?**

d. Work hours

e. Peer review

5) For the final presentation in Sprint 5:

a. Project Overview

A summary of the project goals, intended users, and the main problems our solution addresses.

b. Feature Demonstration

A live or recorded demo showcasing the app's primary features and user experience, walking through typical use cases.

c. Technical Architecture

An overview of the application's architecture, key technologies used, and any notable design decisions or challenges.

d. Project Challenges and Learnings

A brief discussion on any major challenges encountered, how we overcame them, and key takeaways for future development.

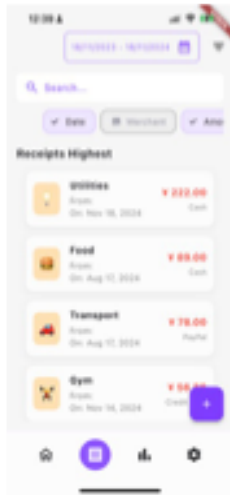
e. Future Enhancements

Ideas for additional features or improvements that could be added in future iterations.

## 4 Sprint 4 Report

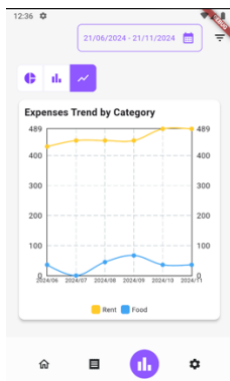
### 4.1 Key Accomplishments

- **Search Bar with Chips for Filtering**



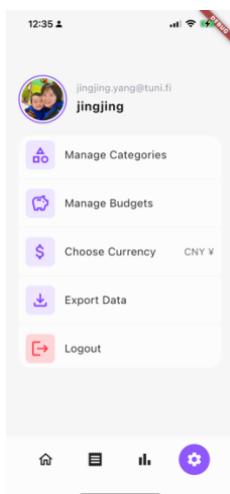
- Introduced a search bar with **chips** for selecting specific keys (e.g., merchant, amount, categoryName) to refine search queries.
- Made each chip selectable, allowing users to search by multiple keys dynamically.
- Utilized a horizontal SingleChildScrollView for smooth scrolling and single-row chip display.

- **Expense Trends by Category Chart**



- The category trend chart displays monthly spending trends for different categories using a line chart.
- X-axis shows months, Y-axis displays amounts, supporting multi-category comparison.
- Legend provides clear labels with color-coded categories for better visualization.

- **Settings Page Improvements**

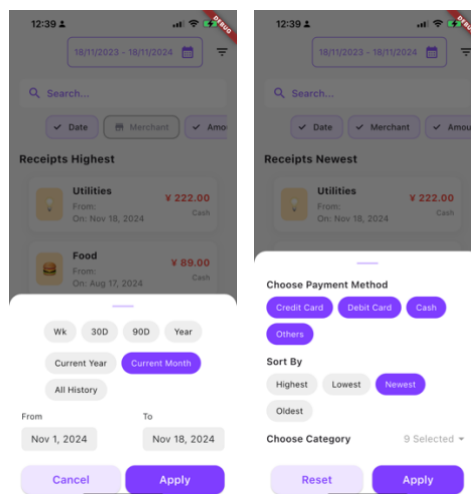


- Centralized management of key features including currency settings, category management, and budget configuration.
- Redesigned layout for easier access to core app settings.
- Enhanced user experience by logically grouping related settings.

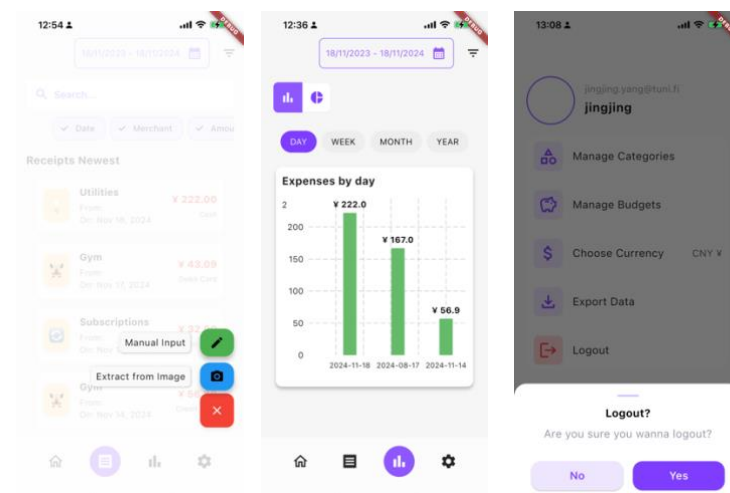
- **Implemented dynamic background color mapping for category icons**



- Enhanced receipt filtering and grouping capabilities



- UI/UX Enhancements



Aligned the app's UI structure with Figma designs, significantly improving user experience. We implemented custom widgets, smooth transitions, and visually cohesive styling. This included updating color schemes, layout components, font sizes, and spacing to ensure a consistent look and feel across all pages.

## 4.2 Technical Accomplishments:

We implemented the Provider pattern with ChangeNotifier to improve state management and data consistency across the app. This approach centralizes state management, making the application more modular, predictable, and scalable. Here's a breakdown of our implementation:

- **Provider Structure:**
  - CategoryProvider: Manages category data (fetching, adding, deleting)
  - BudgetProvider: Handles budget data for each category
  - UserProvider: Manages user profile information
  - ReceiptProvider: Handles receipt-related operations and visualizations
- **ChangeNotifier Implementation:**
  - Each provider extends ChangeNotifier for real-time UI updates
  - notifyListeners() triggers rebuilds of subscribed widgets
  - Example: BudgetProvider updates UI after budget changes
- **Inter-Provider Dependencies:**
  - Used ChangeNotifierProxyProvider for shared data access
  - Allows providers to receive updates based on other providers' data
- **Performance Optimization:**
  - Implemented lazy loading to avoid unnecessary data fetching
  - Used addPostFrameCallback for proper context initialization

These implementations significantly enhanced navigation, data passing between pages, and overall app responsiveness.

## 4.3 Next Goals

- Complete full **UI upgrade** across all pages.
- Conduct comprehensive **functionality testing**.
- Prepare **project documentation**.
- Record a **demonstration video** of the app.
- **Package app files** for teacher review instead of app store publication.

This sprint marks significant progress in both functionality and user interface, bringing the project close to completion. Our focus now shifts to final polishing, testing, and documentation to ensure a high-quality deliverable.





## 5 Sprint 5 Report

Through this project, our team gained valuable insights into project development, including lessons learned from challenges and inefficiencies. Below is a summary of what we learned and how our experiences shaped the outcome.

### 5.1 Challenges and Lessons Learned

#### 5.1.1 Initial Planning and Adjustments

While we initially aimed to follow a structured plan, we encountered challenges due to **insufficient preparation in the early stages**:

- **Page Redesigns:**

The **lack of a finalized page layout and design early on** led to rework, requiring us to revise and reset several pages later in the process. This consumed extra time and effort.

<https://marketplace.flutterflow.io/item/nbP3sGtlb2qdl6TNorKf>

- **Backend Adjustments:**

Some aspects of the backend data structure were not fully aligned with the initial design, leading to additional modifications as new requirements emerged.

These experiences highlighted the importance of dedicating more time to **planning the data structure, relationships, and page design before diving into development**.

#### 5.1.2 Turning Challenges into Opportunities

Despite the inefficiencies, the adjustments provided opportunities for improvement:

- **Code Refactoring**

**Revisiting and reorganizing our code during redesigns** allowed us to **optimize the overall structure**, making it cleaner and more maintainable.

We identified redundancies and implemented best practices to improve performance and readability.

- **Improved Collaboration**

The need for revisions encouraged us to communicate more effectively within the team, ensuring everyone was aligned with the updated structure and design.

Documentation improved as we realized its importance for smoother transitions between tasks and members.

#### 5.1.3 Development Process and Key Takeaways

- **Core to Extended Features**

We recognized the importance of focusing on core functionalities first to establish a stable foundation before adding additional features.

- **Documentation and Versioning**

Consistently tagging files and maintaining clear documentation became essential as the project evolved, especially during the redesign phases.

- **Testing and Feedback**

Testing helped identify areas for improvement, and feedback from team members during the redesigns contributed to a better overall structure.

## **5.2 Final Reflections**

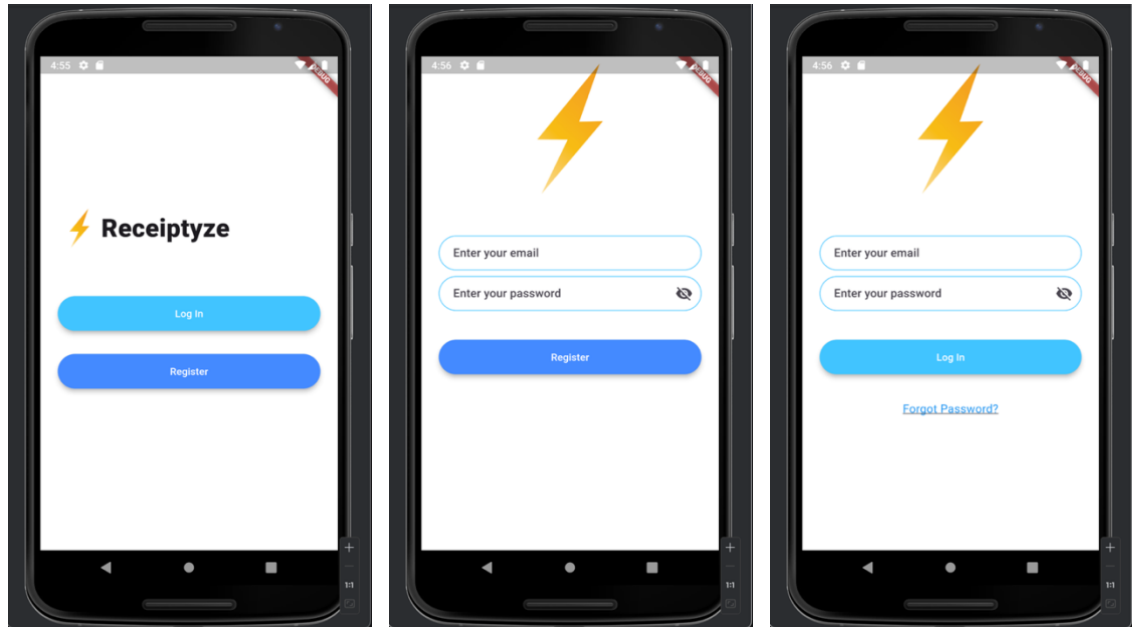
This project taught us the importance of detailed planning and the value of **flexibility** when facing unexpected challenges. While the initial inefficiencies led to extra work, they provided opportunities to optimize our codebase and improve our teamwork. Moving forward, we aim to allocate more time for **planning and design** while maintaining the adaptability we demonstrated during this project.

These experiences will serve as valuable lessons for future projects, allowing us to balance careful preparation and the flexibility to adapt as needed.

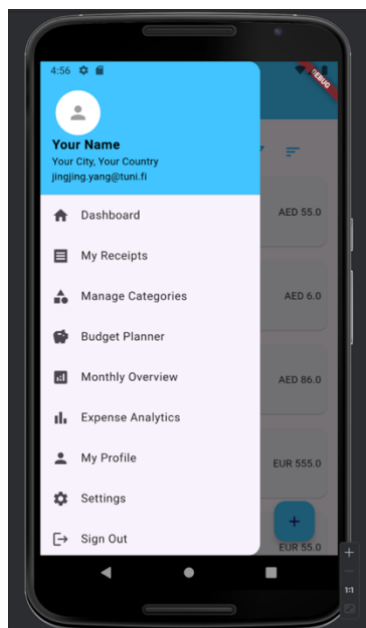
## APPENDICES

### Appendix 1. UI version 1 demo

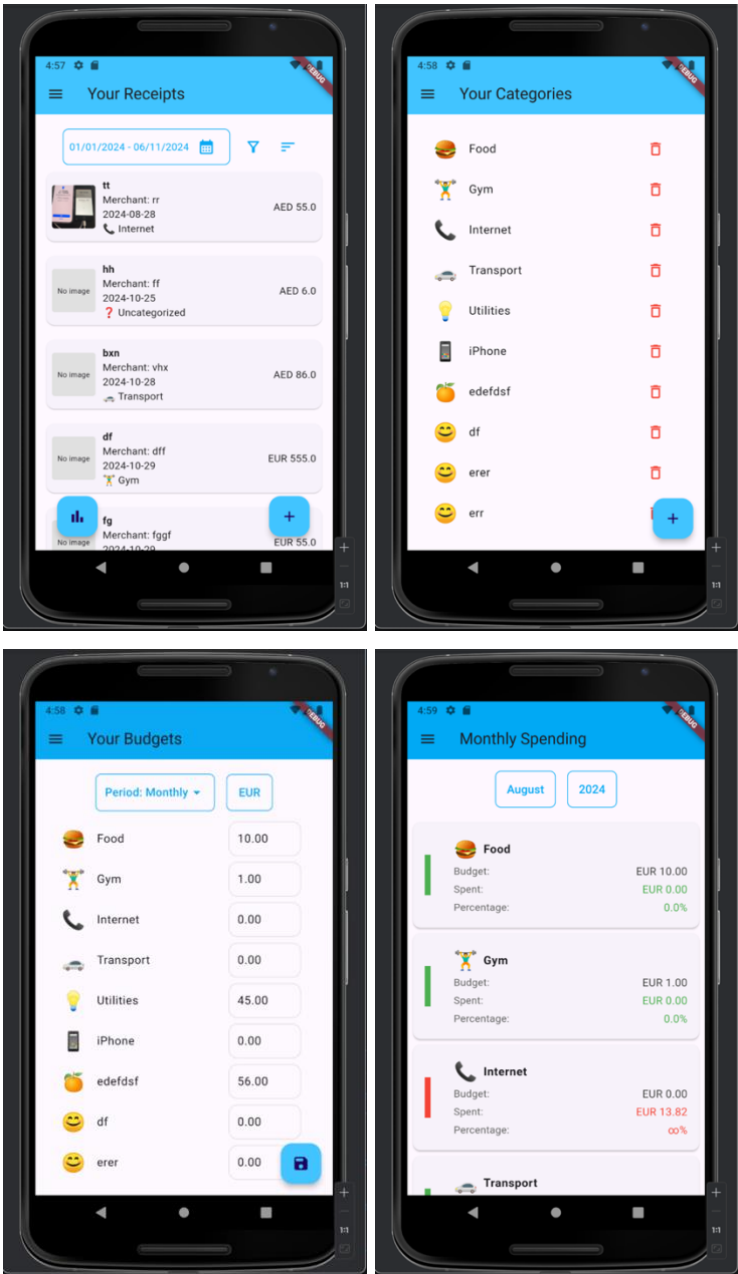
#### 1. Register and login



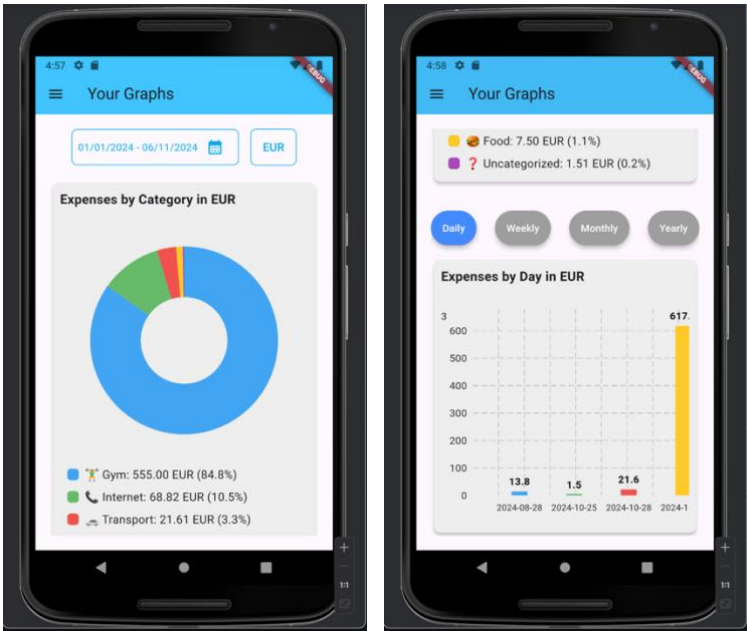
#### 2. Menu



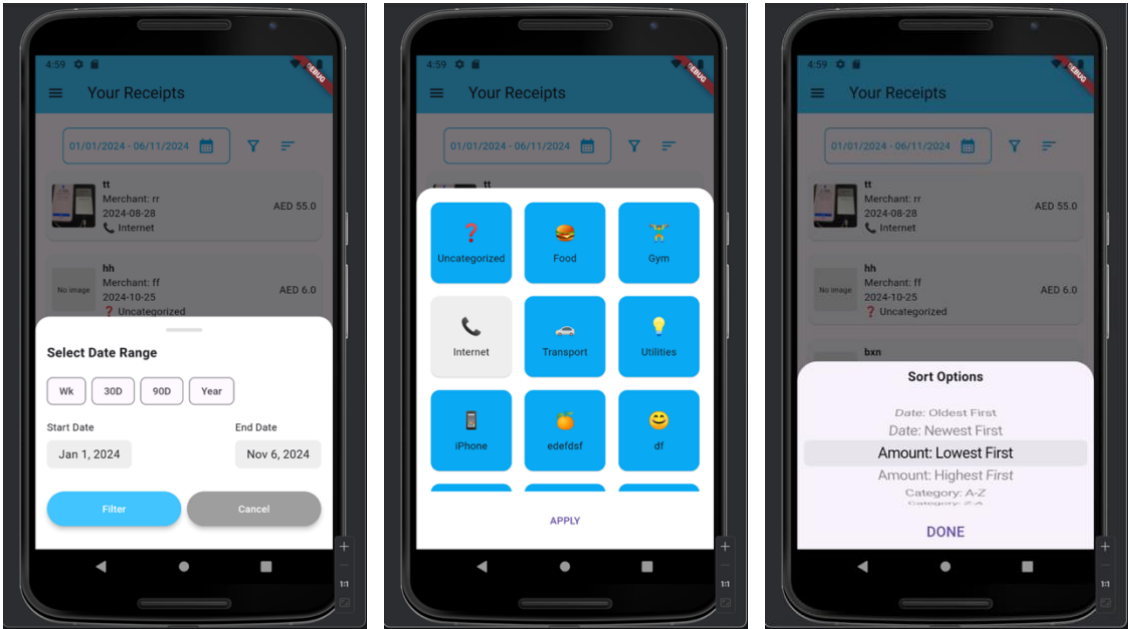
#### 3. Receipts, categories, budgets, monthly overview



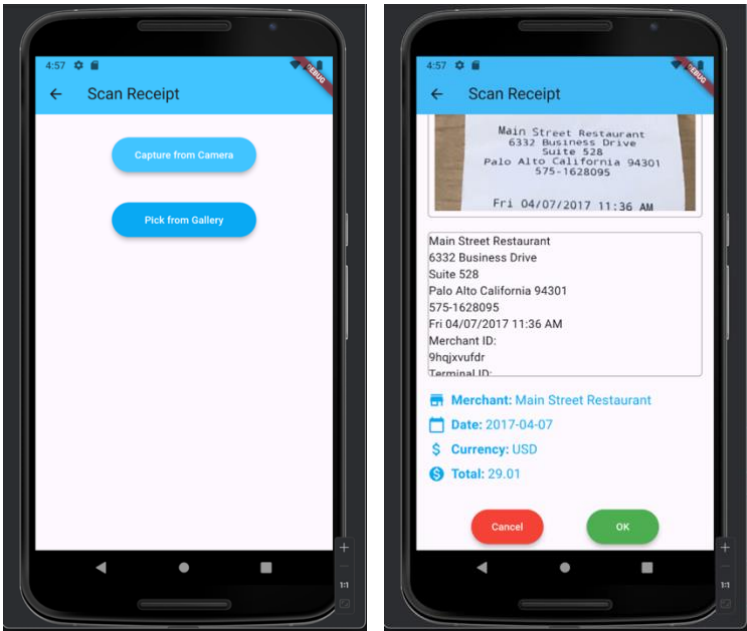
4. Charts



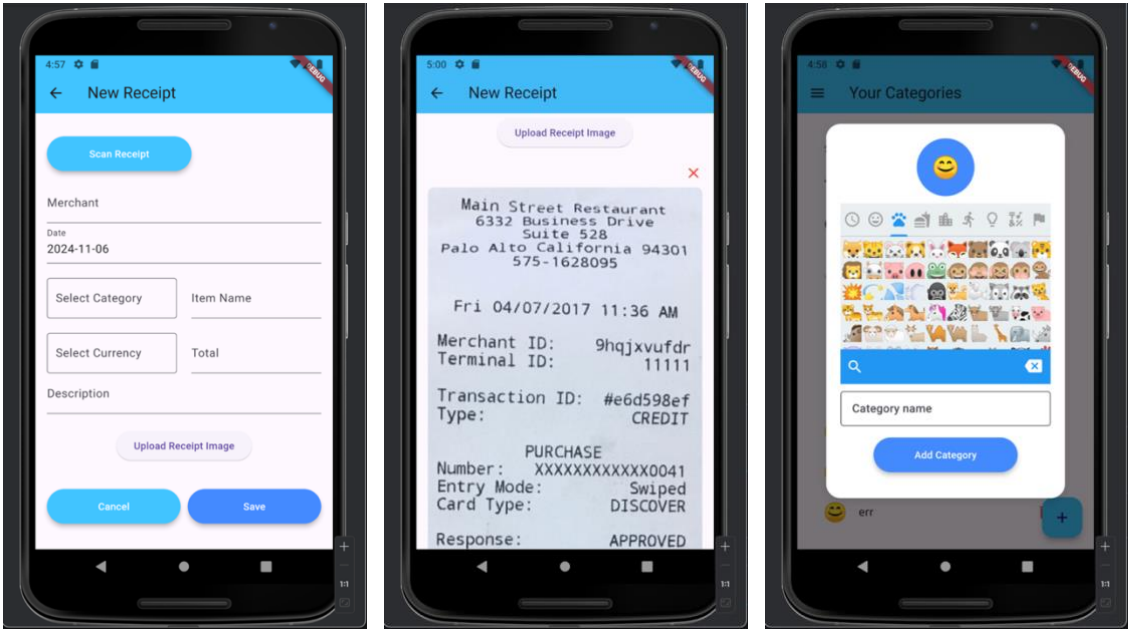
5. Filtering and Sorting



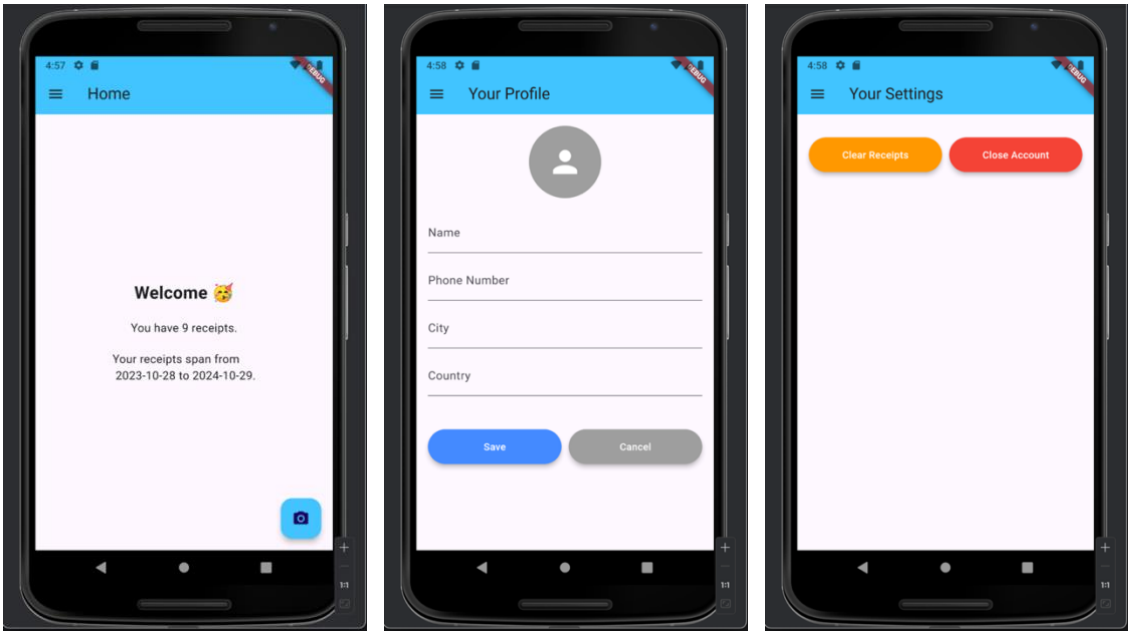
6. OCR



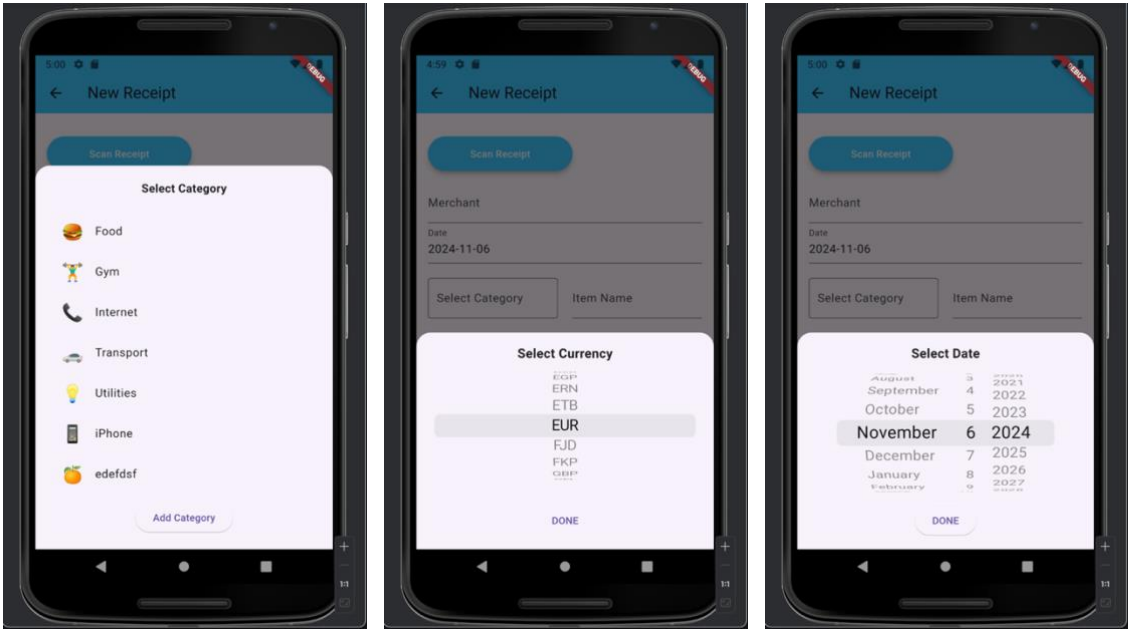
7. Add new receipt, new category



8. Home, Profile, Setting



9. Some picker and popup



## Appendix 2. Summary of Tests and Expected Results

### 1. Authentication

#### A. Login:

- a. **Test:** Valid and invalid credentials.
- b. **Expected Result:** Successful login navigates to the app; invalid login shows an error message.

#### B. Forgot Password:

- a. **Test:** Submit an email for password reset.
- b. **Expected Result:** Email is sent to the user; invalid email shows an error.

#### C. Register:

- a. **Test:** Register a new user with valid, missing, and invalid details.
- b. **Expected Result:** Successful registration logs in the user and initializes profile data; errors for invalid details.

### 2. Receipts

#### A. Empty Receipt List:

- a. **Test:** Load with no receipts in the database.
- b. **Expected Result:** Pages display a "No data available" message on the receipt list, summary, and report pages.

#### B. Add Receipt:

- a. **Test:** Add a receipt and view it in the receipt list, summary, and reports.
- b. **Expected Result:** New receipt appears in the receipt list, updates totals and charts in the summary and report pages.

#### C. Update Receipt:

- a. **Test:** Edit details of an existing receipt.
- b. **Expected Result:** Changes reflect in the receipt list, summary, and report pages.

#### D. Remove Receipt:

- a. **Test:** Delete a receipt.
- b. **Expected Result:** Receipt is removed, and the receipt list, summary, and report pages update accordingly.

### 3. Categories



**A. Empty Category List:**

- a. **Test:** Load with no categories in the database.
- b. **Expected Result:** Receipts without categories are marked as "Uncategorized," and the uncategorized section is visible.

**B. Add Category:**

- a. **Test:** Add a new category and assign it to receipts.
- b. **Expected Result:** The new category appears in filters, receipt details, and is reflected in the summary and report charts.

**C. Update Category:**

- a. **Test:** Change the name or icon of a category.
- b. **Expected Result:** Updates are reflected across all pages (receipt list, summary, and report charts).

**D. Remove Category:**

- a. **Test:** Delete a category.
- b. **Expected Result:** Receipts in the deleted category are marked as "Uncategorized," and all pages update accordingly.

**4. Budgets****A. Empty Budget List:**

- a. **Test:** Load with no budgets in the database.
- b. **Expected Result:** Budget-related sections display "No data available" in the summary page.

**B. Add Budget:**

- a. **Test:** Add a budget for a category.
- b. **Expected Result:** Budget appears in the summary page, and spending comparisons update.

**C. Update Budget:**

- a. **Test:** Edit an existing budget.
- b. **Expected Result:** Changes reflect in the budget summary.

**D. Remove Budget:**

- a. **Test:** Delete a budget.
- b. **Expected Result:** Budget is removed, and the summary page updates accordingly.

**5. User Profile****A. Currency Change:**

- a. **Test:** Change the user's currency.
- b. **Expected Result:** All amounts (receipts, summaries, charts) are converted and displayed in the new currency.

**B. Edit User Name:**

- a. **Test:** Change the user's name.
- b. **Expected Result:** New name appears on the profile and settings pages.

**C. Edit Profile Image:**

- a. **Test:** Update the user's profile picture.
- b. **Expected Result:** New image is displayed in the profile and settings sections.

**D. Empty Profile Data:**

- a. **Test:** Load a user with no profile data.
- b. **Expected Result:** Default placeholders (e.g., "Your Name") are displayed without breaking functionality.

**6. Filters and Search**

**A. Filter Change (Category, Payment Method):**

- a. **Test:** Apply category and payment filters (e.g., one selected, all selected, none selected).
- b. **Expected Result:** Receipts, summaries, and charts update to reflect the selected filters.

**B. Search Bar:**

- a. **Test:** Search receipts by merchant, category, or other fields.
- b. **Expected Result:** Search results update dynamically based on the query and selected filters.

**C. Date Range Change:**

- a. **Test:** Update the date range filter.
- b. **Expected Result:** Receipts and charts update to display data within the selected range.

**D. All Filters Selected:**

- a. **Test:** Select all filters.
- b. **Expected Result:** Full dataset is displayed across all pages.

**E. No Filters Selected:**

- a. **Test:** Deselect all filters.

- b. **Expected Result:** No data is displayed with a "No data available" message.

## 7. Charts

### A. Pie Chart (Report Page):

- a. **Test:** Display expenses grouped by category.
- b. **Expected Result:** Pie chart sections and legends update dynamically based on the filtered receipts.

### B. Bar Chart (Report Page):

- a. **Test:** Display expenses grouped by time interval.
- b. **Expected Result:** Bar chart updates dynamically to reflect the selected interval and filters.

### C. Line Chart (Report Page):

- a. **Test:** Display expenses trend by category over time.
- b. **Expected Result:** Line chart shows trends accurately, with categories reflected in the legend.

## 8. Global Reflections

### A. Consistency Across Pages:

- a. **Test:** Make changes (e.g., add receipt, update filters) and navigate between pages.
- b. **Expected Result:** Changes reflect instantly on all relevant pages (receipt list, summary, and reports).

### B. Data Sync on Reload:

- a. **Test:** Refresh the app or logout and log back in.
- b. **Expected Result:** All data persists and reflects correctly based on the saved database state.

### C. Error States:

- a. **Test:** Simulate database errors or connectivity issues.
- b. **Expected Result:** User-friendly error messages are displayed, and the app remains functional.