CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-G5
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: WorthIt

Prepared By: Lerah A. Caones

Date of Submission: February 5, 2026

Version: 1

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to define the software requirements for the WorthIt budgeting system. It details the functional and non-functional requirements intended for developers, project managers, and stakeholders to ensure the final product meets the goal of simplifying personal finance management.

1.2. Scope

WorthIt is a unified, cross-platform financial tracking application designed to alleviate "budgeting burnout." The system categorizes expenses into two simple buckets: **Needs** (Essentials) and **Wants** (Extras).

- **Unified Experience:** Whether accessing via the **Mobile App** or **Web Browser**, users have full feature parity. A user can log transactions, view deep analytics, and manage settings from any device.
- **The "Filter" Strategy:** The system employs a logic layer that tracks all "Needs" for accuracy but segments "Wants" based on value (Micro-Treats vs. Splurges) to help users visualize their spending habits without friction.

1.3. Definitions, Acronyms, and Abbreviations

- **Need:** An essential expense required for daily living (e.g., Rent, Utilities, Food).
- **Want:** A discretionary expense (e.g., Entertainment, Dining out).
- **Micro-Transaction:** A "Want" expense below a user-defined threshold (e.g., < 200 PHP).
- **Splurge:** A "Want" transaction that is **above** a specific monetary threshold (e.g., > 200 PHP), representing a significant discretionary purchase.

2. Overall Description

2.1. System Perspective

WorthIt functions as a centralized client-server system.

- **Frontend:** React (Web) and Kotlin (Mobile).
- **Backend:** Spring Boot REST API.
- **Synchronization:** Any action taken on one platform (e.g., logging a "Want" on the phone) is instantly reflected on the other (e.g., the dashboard on the laptop).

2.2. User Classes and Characteristics

Guest User: An unregistered visitor who can view the landing page and "About" information but cannot store data.

Registered User: A financially conscious individual seeking a low-friction tracking method. They have full access to logging, editing, and viewing their financial analytics.

2.3. Operating Environment

Client (Mobile): Android

Client (Web): Modern web browsers (Chrome, Firefox, Safari, Edge).

Network: Requires internet connection for syncing; local storage handles temporary offline access.

2.4. Assumptions and Dependencies

Assumption: Users have basic digital literacy to operate a mobile app.

Assumption: Users enter accurate financial data (the system validates format, not truthfulness).

Dependency: The system relies on a stable internet connection for real-time syncing between Web and Mobile.

3. System Features and Functional Requirements

Feature 1: User Authentication & Profile Management

Description: Allows users to create a secure account, log in to access their private data, and manage their session across devices.

Functional Requirements:

- The system shall allow users to register using a unique Username, Email, and Password.
- The system shall validate that the Password meets security complexity standards (e.g., min 8 chars).
- The system shall authenticate users via Email and Password, issuing a JWT upon success.
- The system shall allow users to log out, invalidating their current session token.
- The system shall prevent access to Dashboard pages for unauthenticated users (Guest).

Feature 2: Transaction Logging

Description: The core input mechanism for tracking spending, designed for speed and ease on any device.

Functional Requirements:

- The system shall provide a "Quick Add" button visible on the home screen of both the Web and Mobile interface.
- The system shall accept any Amount and allow categorization as "Need" or "Want."

- The system shall allow users to input "Micro-Wants" (small amounts) without restriction.
- The system shall support editing or deleting past transactions from any device
- The system shall allow users to add an optional description text.
- The system shall automatically tag the transaction with the current Date and Time.

Feature 3: Expense Analysis Dashboard

Description: A visual breakdown of spending habits, available to view on the go (Mobile) or in detail (Web).

Functional Requirements:

- The system shall render a "Ratio Chart" (Needs vs. Wants) that is optimized for both touch screens (Mobile) and mouse pointers (Web).
- The system shall filter "Wants" into sub-groups (e.g., Under 200 vs. Over 200) for better insight.
- The system shall update charts in real-time as soon as a new transaction is added.

4. Non-Functional Requirements

4.1 Performance

- **Response Time:** API requests (login, save transaction) should complete within 500ms under normal load.
- **Sync Speed:** Data logged on Mobile should appear on the Web Dashboard within 2 seconds.

4.2 Security

- **Data Protection:** Passwords must be hashed using BCrypt before storage in the database.
- **Access Control:** All API endpoints (except Login/Register) must be protected by JWT authorization.

4.3 Usability

- **Consistency:** The color scheme, button placement, and icons must be identical across Web and Mobile to reduce user learning curve.
- **Responsiveness:** Dashboard charts must resize automatically to fit the screen width (Phone vs. Desktop) without breaking layout.
- **Clarity:** The distinction between "Needs" and "Wants" must be visually distinct (e.g., color-coded UI).

4.4 Reliability

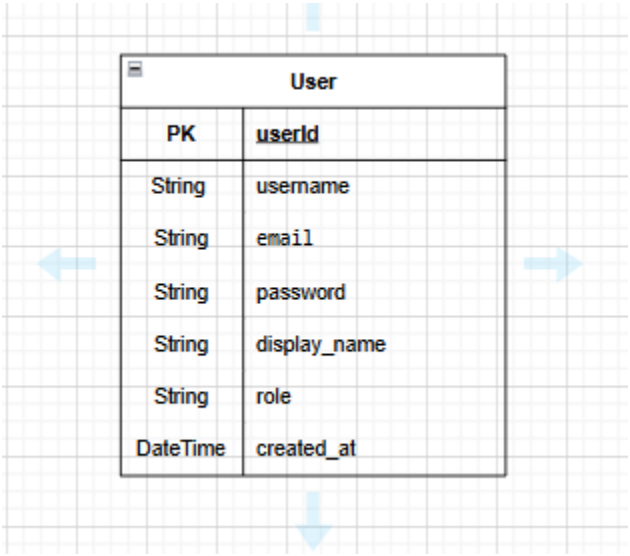
Availability: The system should be available 99% of the time during business hours.

Data Integrity: The system must ensure no data is lost during a failed sync attempt (e.g., if the internet cuts out).

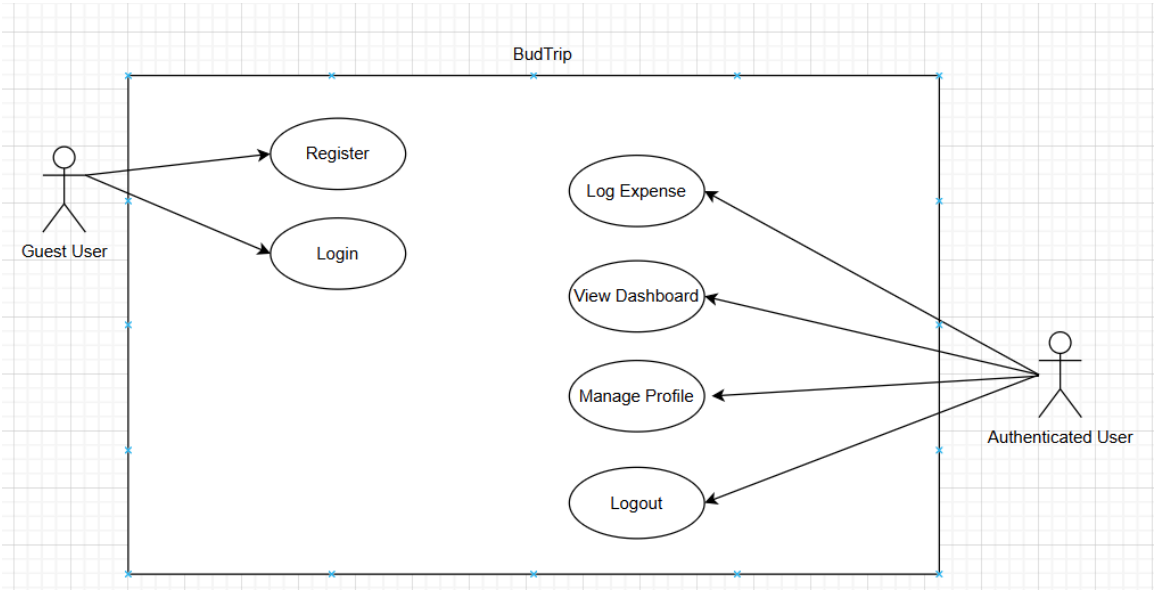
5. System Models (Diagrams)

Insert the necessary diagrams for the system:

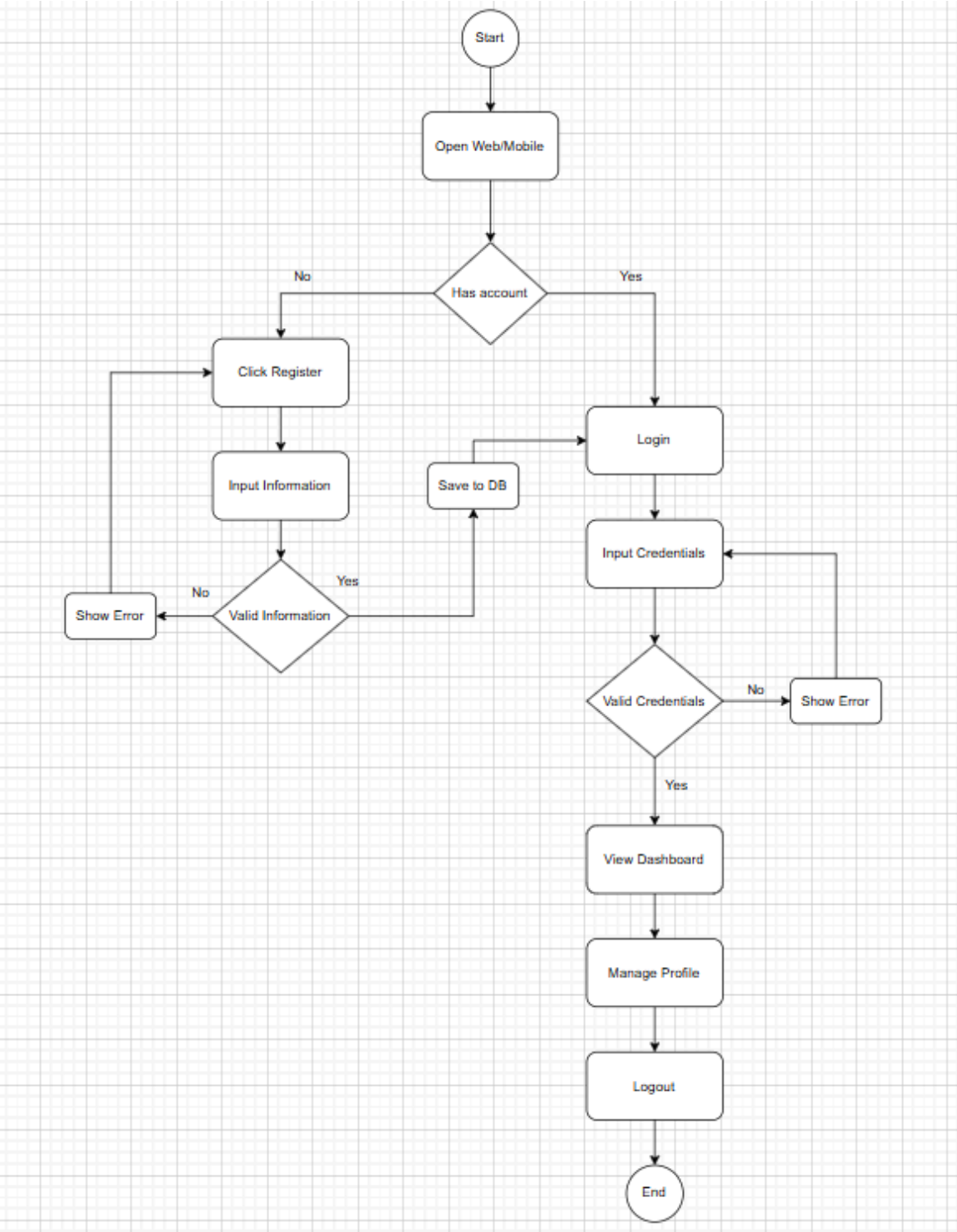
5.1. ERD



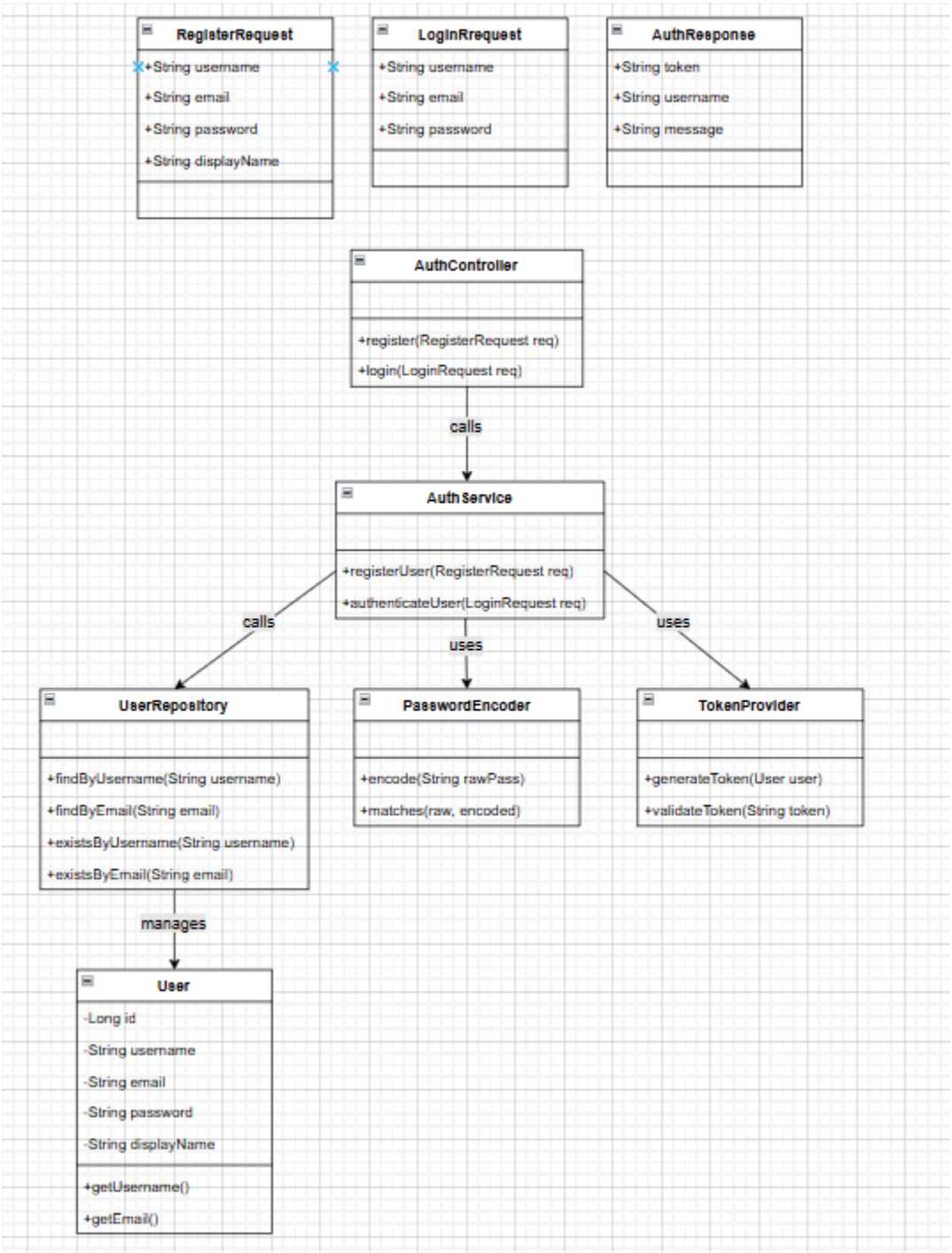
5.2. Use Case Diagram



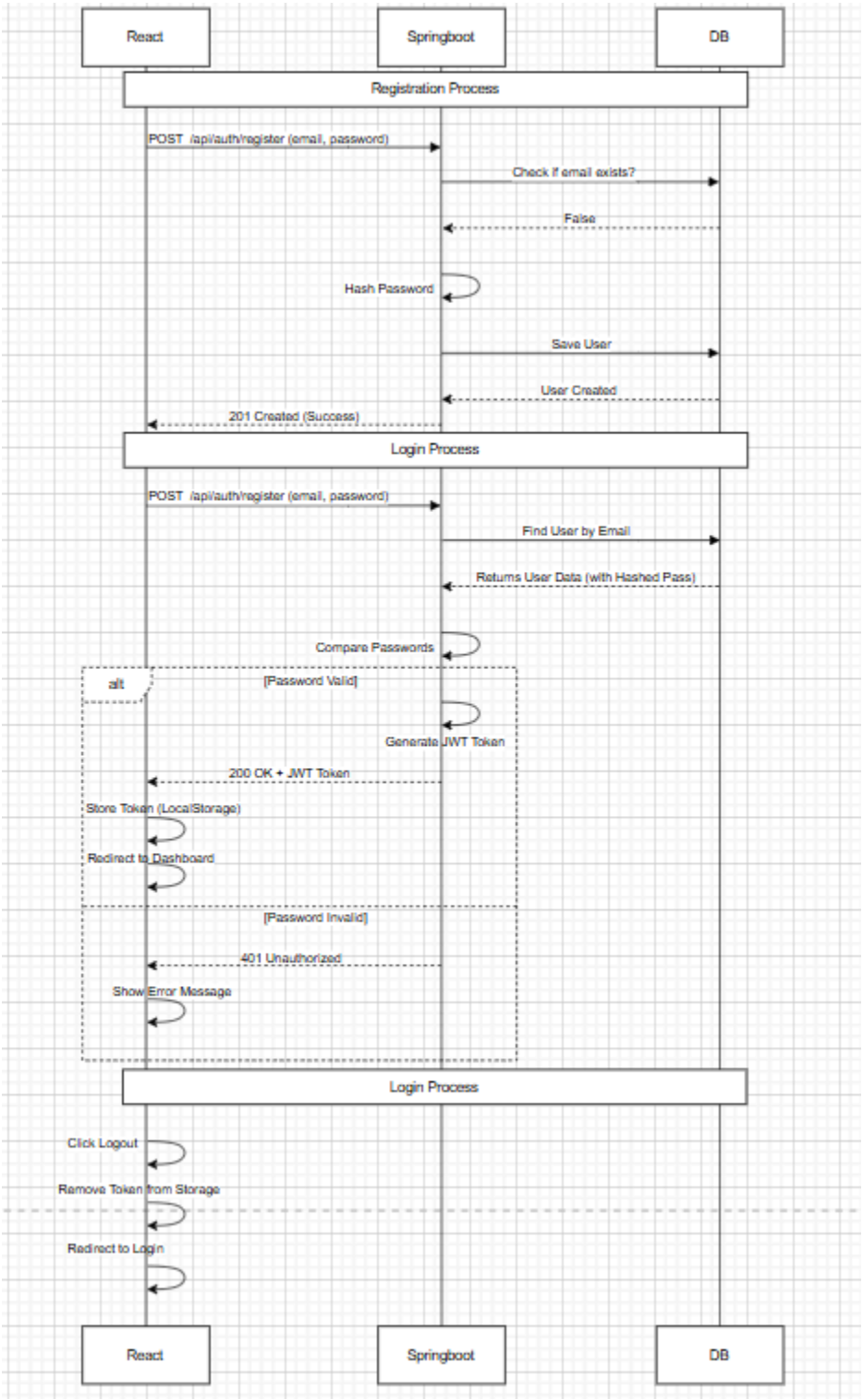
5.3. Activity Diagram



5.4. Class Diagram



5.5. Sequence Diagram

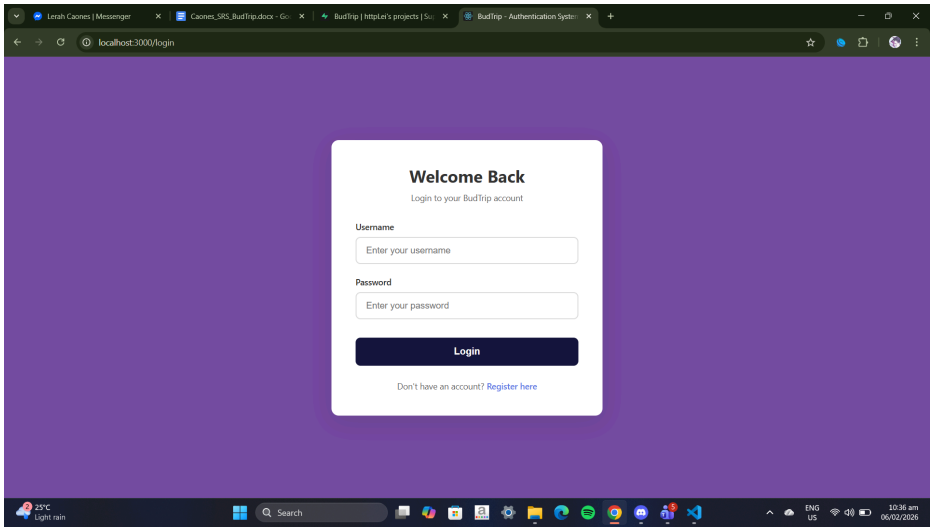


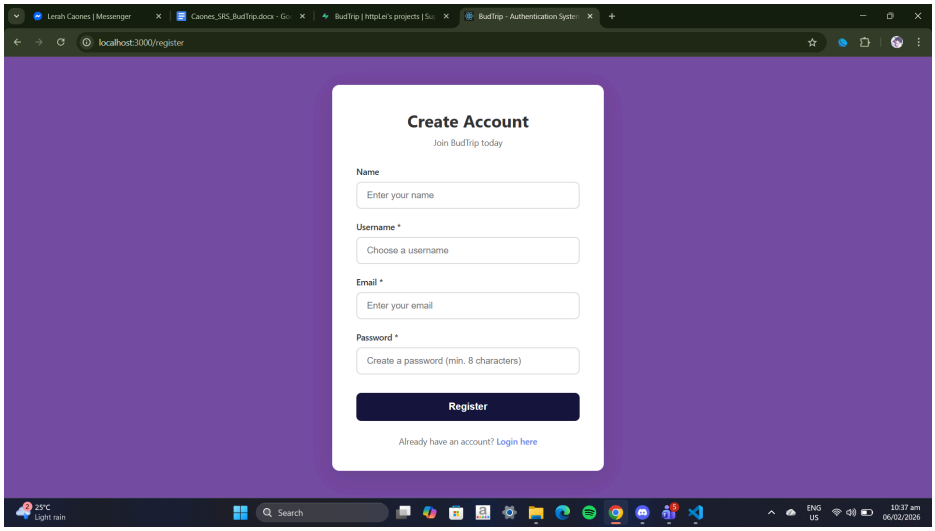
6. Appendices

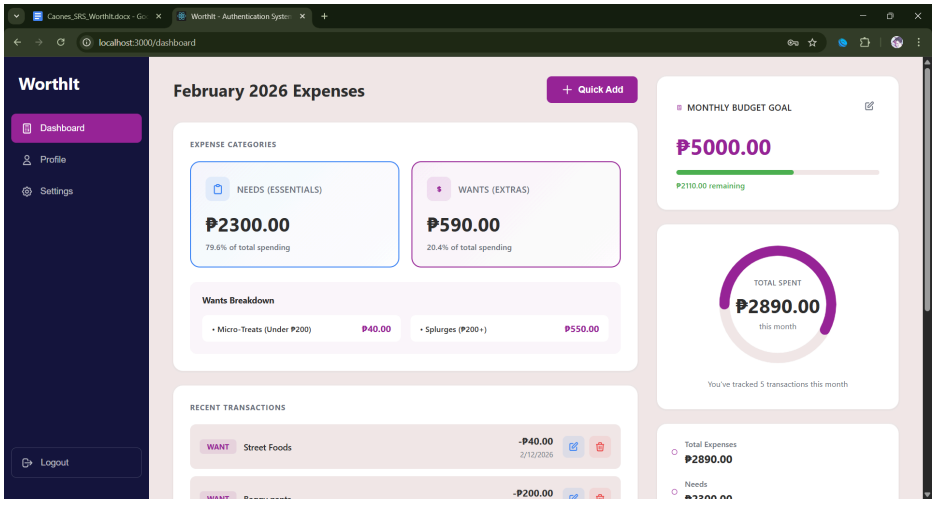
Include any additional information, references, or support materials.

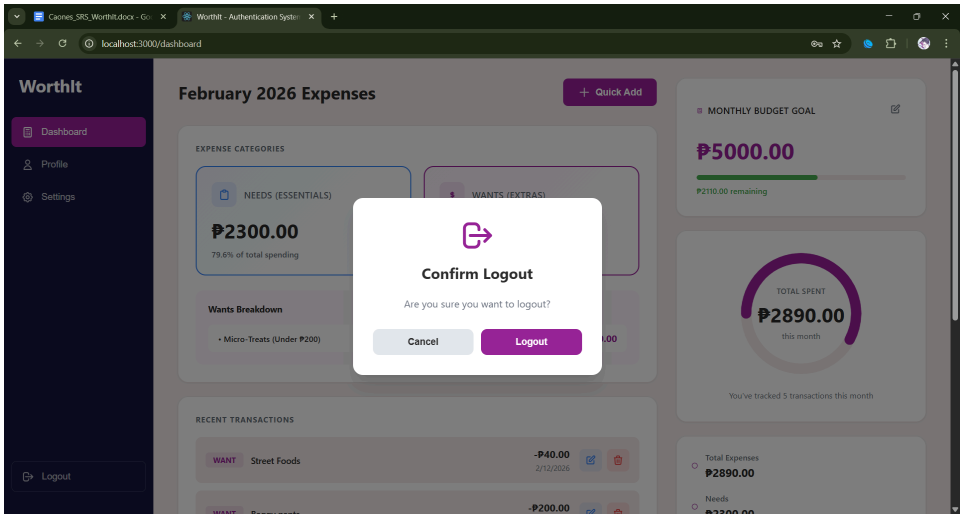
7. Website Screenshots

Register, Login, Dashboard/Profile, Logout

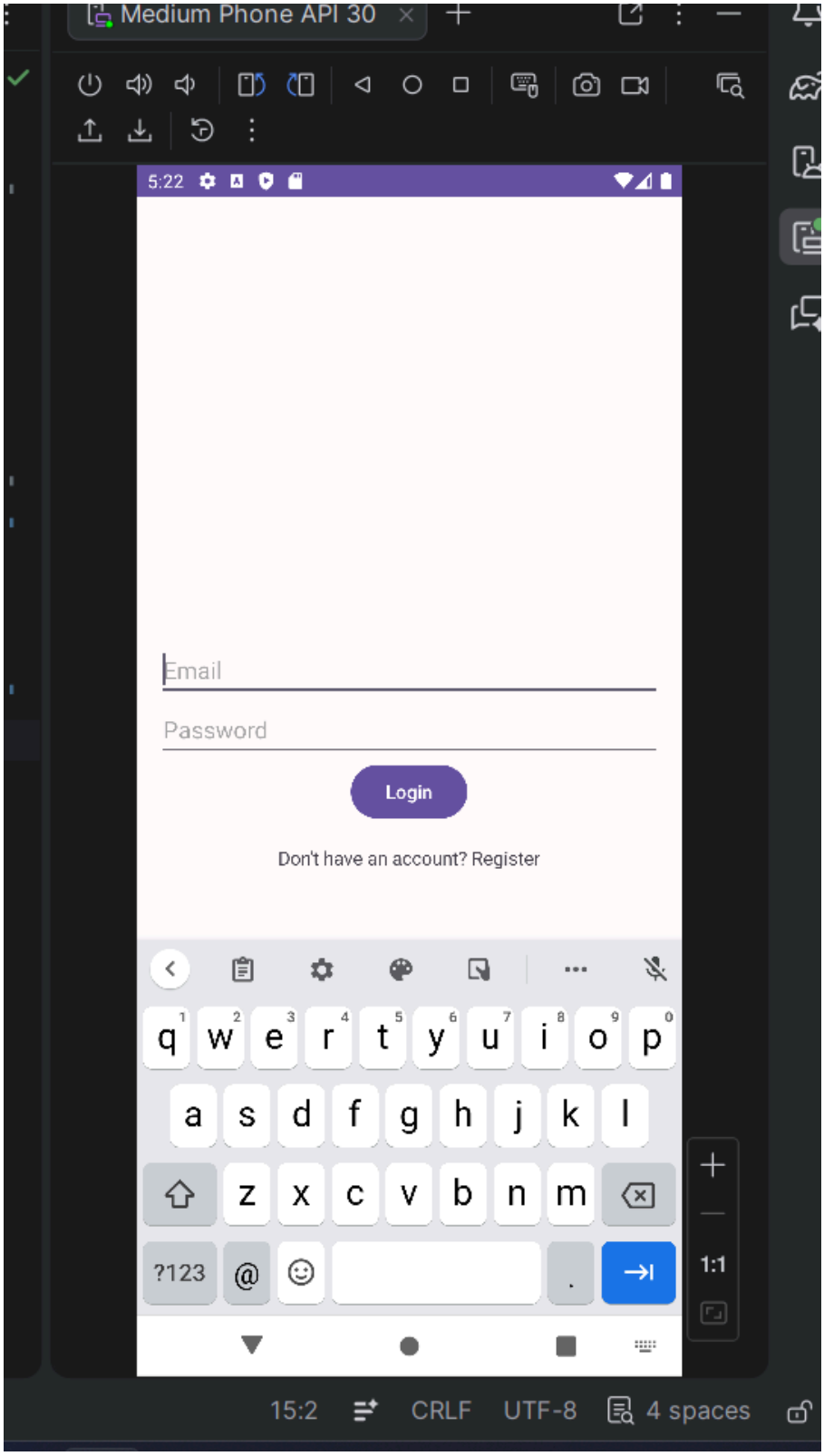


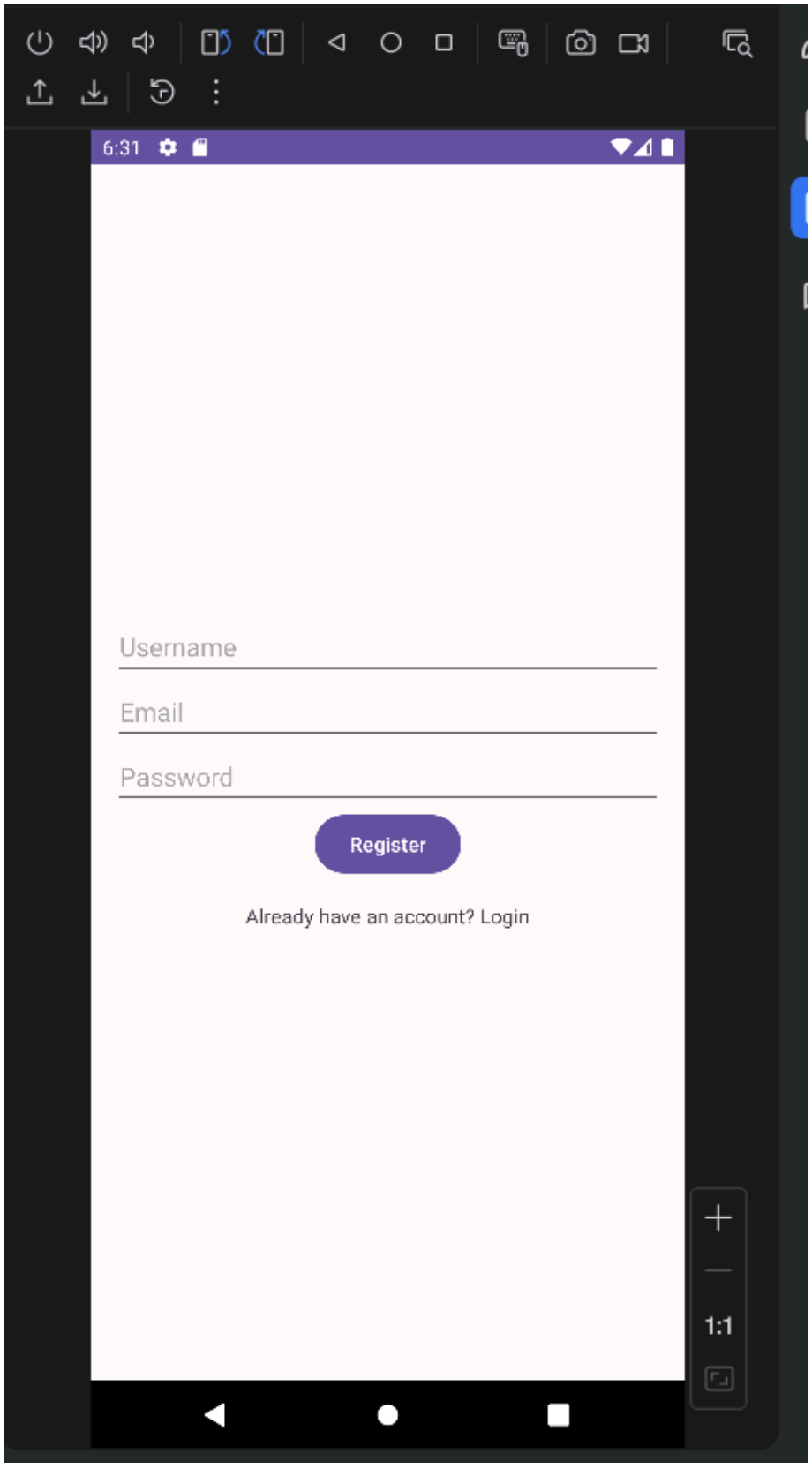


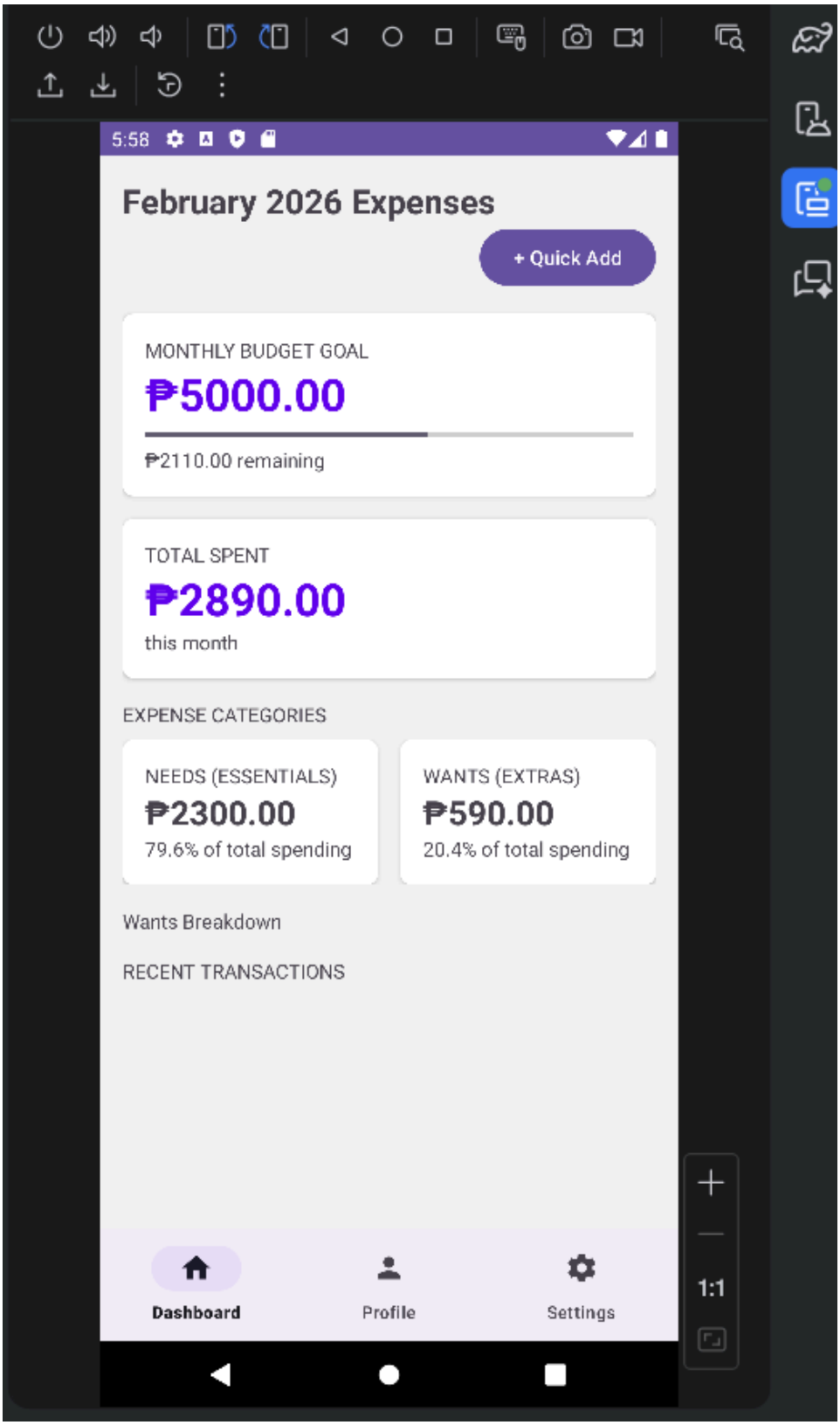


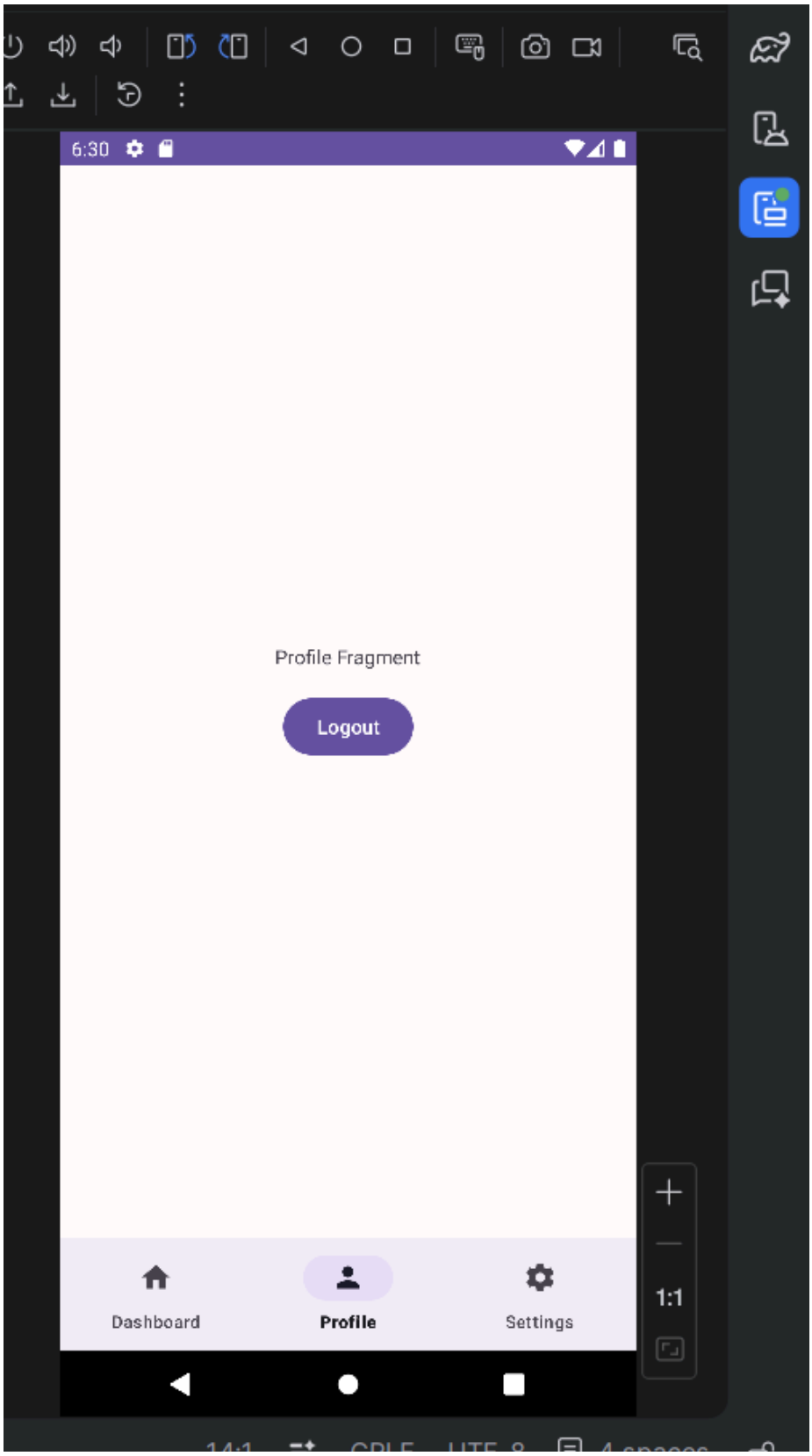

Caones_SRS_WorthIt.docx - Go
WorthIt - Authentication Syste
localhost:3000/dashboard
WorthIt
Dashboard
Profile
Settings
Logout
February 2026 Expenses
Quick Add
EXPENSE CATEGORIES
NEEDS (ESSENTIALS)
₱2300.00
79.6% of total spending
WANTS (EXTRAS)
Wants Breakdown
• Micro-Treats (Under ₱200)
Confirm Logout
Are you sure you want to logout?
Cancel
Logout
RECENT TRANSACTIONS
WANT
Street Foods
-₱40.00
2/12/2026
-₱200.00
MONTHLY BUDGET GOAL
₱5000.00
₱2110.00 remaining
TOTAL SPENT
₱2890.00
this month
You've tracked 5 transactions this month
Total Expenses
₱2890.00
Needs

8. Mobile Screenshots




6:31
Username
Email
Password
Register
Already have an account? Login
1:1


5:58
February 2026 Expenses
+ Quick Add
MONTHLY BUDGET GOAL
₱5000.00
₱2110.00 remaining
TOTAL SPENT
₱2890.00
this month
EXPENSE CATEGORIES
NEEDS (ESSENTIALS)
₱2300.00
79.6% of total spending
WANTS (EXTRAS)
₱590.00
20.4% of total spending
Wants Breakdown
RECENT TRANSACTIONS
Dashboard
Profile
Settings
1:1


6:30
Profile Fragment
Logout
Dashboard
Profile
Settings
1:1