

CampusExpense Manager

PERFORMED BY: GROUP 2

GROUP MEMBERS:

- VANG THI TUOI
- NGO KIM LOC
- NGUYEN DUY THANH
- VU DUY AN

CLASS: SE06204

SUPPORTING INSTRUCTOR: DINH VAN DONG

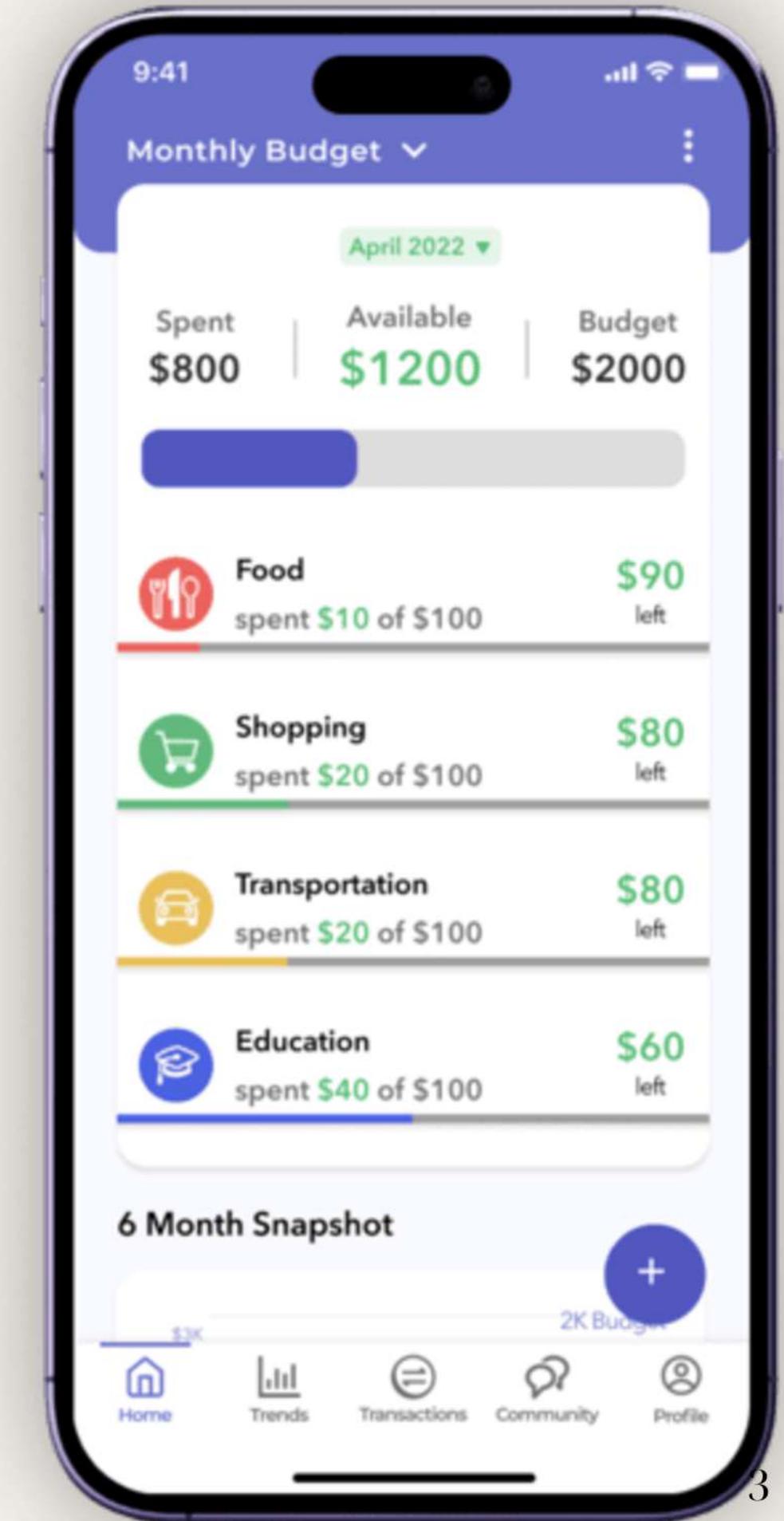
Table of contents

OF
DUY THANH

3	PROJECT OVERVIEW
4	Objectives of the App
5	User Requirements
6	Systems Investigation and Research
7	Project Scope and Constraints
8	Tools and Technologies Used
9	Development Methodology

PROJECT OVERVIEW

CampusExpense Manager is a mobile application designed by the BudgetWise Solutions development team, aiming to support university students in managing their expenses effectively. The application provides a simple, easy-to-use interface, helping to track and control budgets, suitable for all needs of every student.



Objectives of the App

- Simplify Expense Tracking
- Budget Management
- Financial Insights

- Accessibility
- Customization

User Requirements

- **Expense Tracking**
- **Budget Alerts**
- **Intuitive UI**
- **Categorization of Expenses**
- **Summary Reports**

Systems Investigation and Research

Research Summary:

- **Conducted surveys and interviews with university students to understand their financial challenges.**
- **Analyzed competitor apps like Mint and PocketGuard to identify strengths and gaps.**

Results obtained:

- Students prioritize simplicity and ease of use over complex features.
- Real-time budget updates and alerts are highly valued.
- Visual expense summaries (charts, graphs) enhance engagement.
- Categorization options tailored to student life (e.g., tuition, books, rent, dining) are essential.
- Many apps lack customization for unique student needs, presenting an opportunity for differentiation.

Project Scope and Constraints

Project Scope:

- Develop a mobile app for university students to manage expenses.
- Core features: expense tracking, budget alerts, categorization, summary reports.
- Emphasis on user-friendly design for students.

Constraints:

- Limited budget restricts advanced features and testing.
- Small team with limited mobile app development experience.
- Tight deadlines within the academic semester.

Design Influences:

- Prioritized essential features over advanced ones.
- Chose simple, scalable technologies matching the team's skills.
- Focused on usability and minimalism for quick user adoption.

Tools and Technologies Used

Android Studio

Android Studio is the official integrated development environment (IDE) for Android application development. It provides a complete set of tools from coding, testing to building and deploying applications.

Advantages

- Full integration with Android SDK.
 - Powerful development tools
- Audio AVD integration



Android Studio

Disadvantages

- Footprint and resources
- Complicated installation and initial setup

Tools and Technologies Used

SQLITE

SQLite is an administrative database built into Android for storing local data sets. It is used to manage user data such as spending accounts, financial transactions.

Advantages

- Reduced resource usage, suitable for low-end devices
- Cost effective with many open and free source codes
- No internet connection required

Disadvantages

Limited resources and features. Few management tools.



Development Methodology

Agile

Iterative and incremental development:

Software is developed in small iterations, each of which delivers a portion of the finished product. Each iteration improves the product based on feedback and continuous testing.

Flexibility:

Easily adapts to changes in project requirements or priorities.
Suitable for fast-changing environments such as technology projects.

Interaction and collaboration:

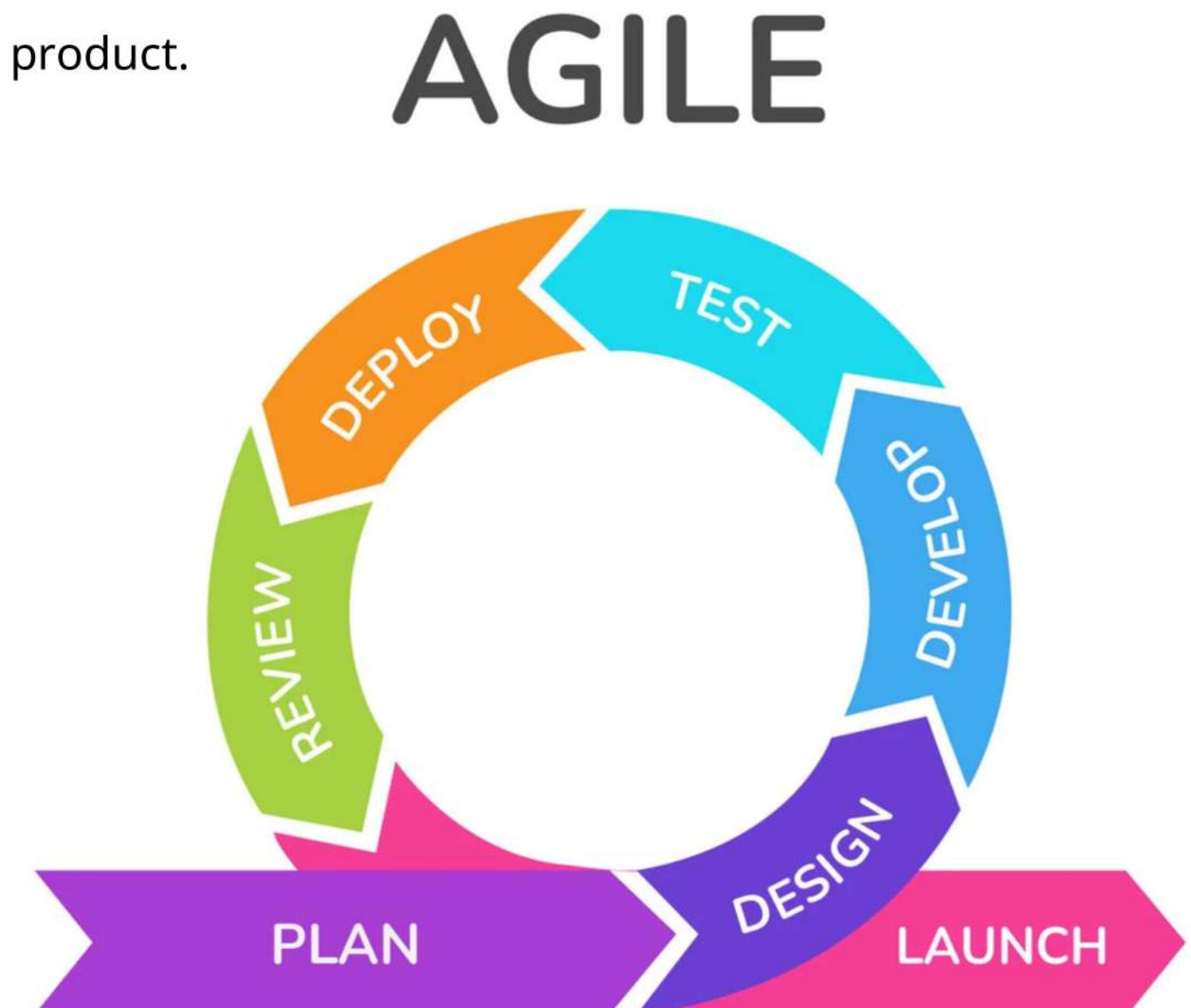
Enhances communication between team members and customers.
Focus on people and collaboration rather than rigid processes.

Deliver early and continuously:

Workable software is delivered after each iteration, helping customers see value early.

Responsive:

Leverage user feedback to improve the product as it develops.



Development Methodology

Agile

Why Agile Was Suitable:

Flexibility:

- Allowed the team to adapt to changes based on feedback from students and stakeholders.
- Iterative approach enabled prioritization of essential features.

Collaboration:

- Regular stand-ups and sprints facilitated effective communication within the team.
- Continuous involvement of stakeholders ensured alignment with project goals.

Incremental Development:

- Delivered the app in smaller, manageable iterations for frequent testing and review.
- Enabled the release of a Minimum Viable Product (MVP) to gather user feedback early.

Time Efficiency:

- Agile's time-boxed sprints helped manage the tight project timeline effectively.

Risk Mitigation:

- Frequent testing and integration reduced the risk of major issues late in development.

Initial Design and Prototyping

USER FEEDBACK:

USER SURVEY RESULTS

NUMBER OF PARTICIPANTS: 100 STUDENTS

1. USABILITY:

EASE OF USE:

1 (VERY DIFFICULT): 5%

2: 10%

3: 25%

4: 30%

5 (VERY EASY): 30%

60% RATED IT EASY TO USE (SCORES 4 AND 5)

2. REQUIRED FEATURES:

ENTER EXPENSES: 85%

CATEGORIZE EXPENSES: 75%

VIEW SPENDING OVERVIEW: 90%

RECEIVE BUDGET NOTIFICATIONS: 70%

3. USER INTERFACE:

INTUITIVE INTERFACE:

YES: 80%

NO: 20%

4. USER GUIDE:

NEED GUIDANCE:

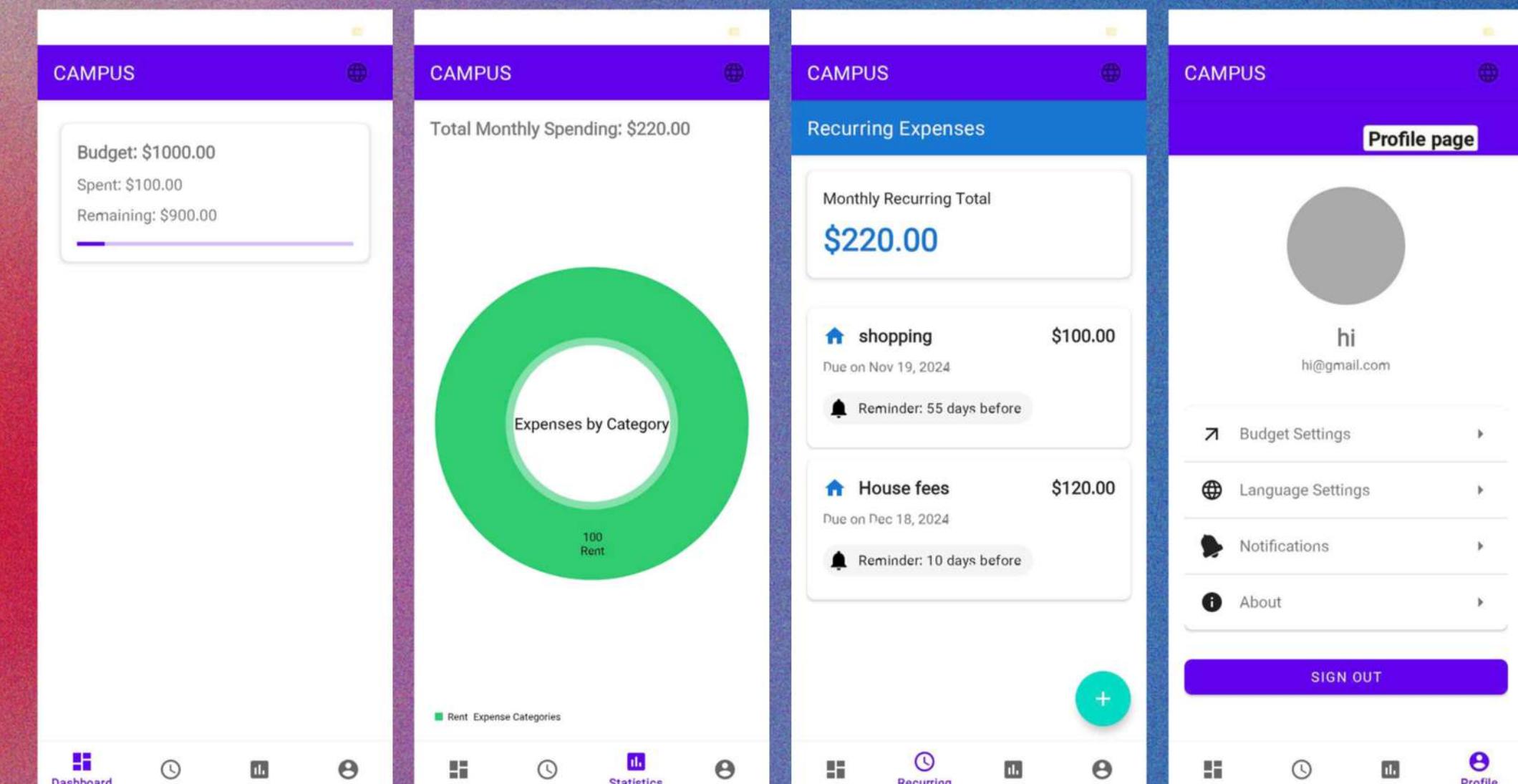
YES: 70%

NO: 30%

PREFERRED FORMAT: VIDEO (50%), STEP-BY-STEP GUIDE (40%)

5. OTHER COMMENTS:

SUGGEST ADDING A SPENDING ANALYSIS FEATURE.



User Interface (UI) Design

Key Design Decisions:

Simple: The interface is designed with a clear layout, minimizing clutter. Light and harmonious colors to create a comfortable feeling for the user.

Intuitive: Use easy-to-understand icons and simple language to make it easy for users to interact. Function buttons are placed in easy-to-see locations, allowing users to access quickly.

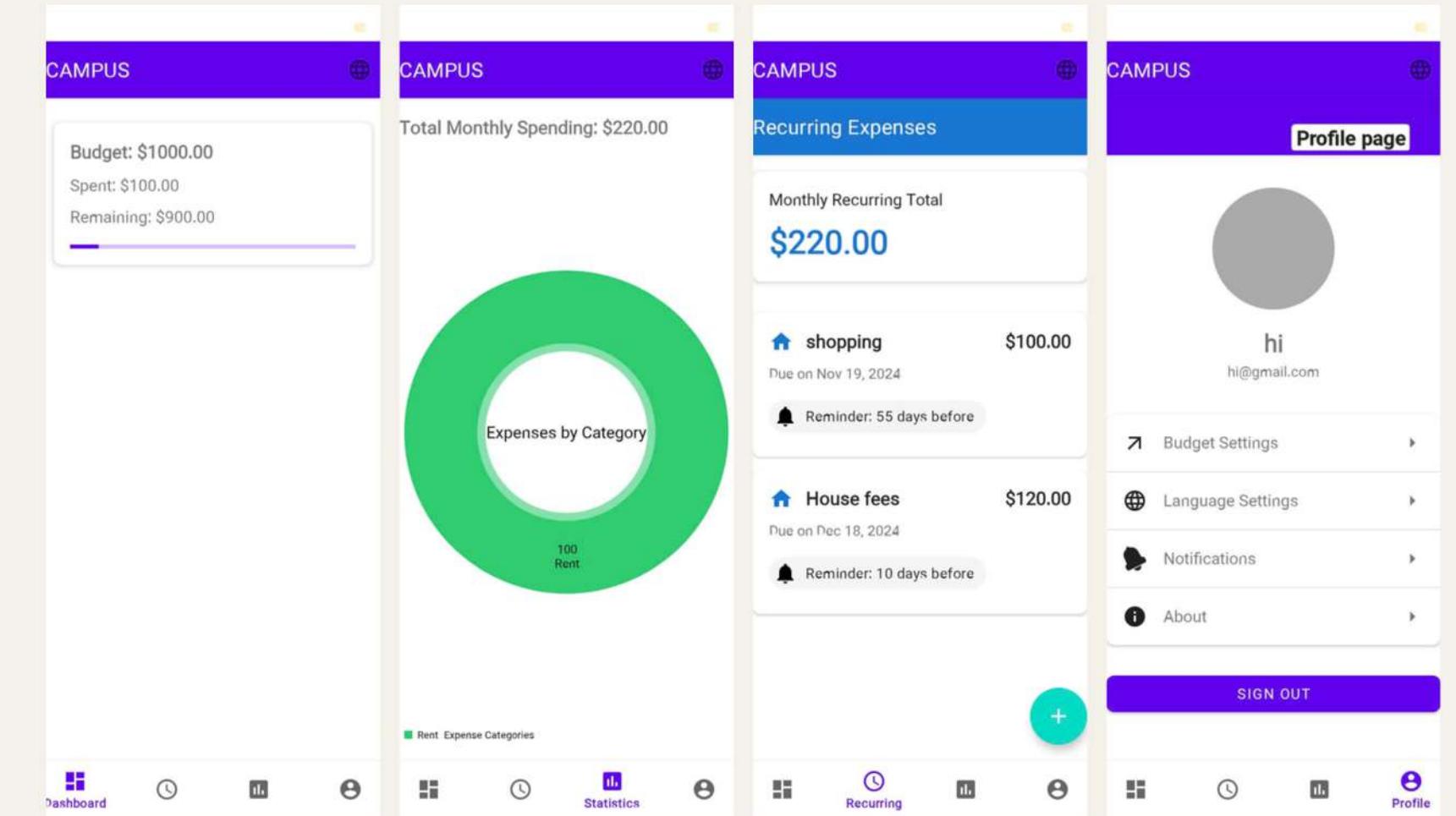
Design Basis Explained:

Design Principles: Focus on user experience (UX) to ensure users feel comfortable using the app.

Use consistent design elements to create brand recognition and ease of use.

User Feedback: The design was adjusted based on survey feedback, ensuring it meets user needs and expectations.

UI Design Screenshots:



Backend Architecture

SQLite:

Data Storage:

- SQLite is an embedded database, ideal for mobile applications with compact size.
- Data is stored locally on the device, helping to increase access speed and reduce latency.

Outstanding Features: No need to install a server, easy to integrate into the application. Supports standard SQL queries, helping to manage data effectively.



Core Features Implemented

1. Expense Logging

Functionality:

- Users can easily record daily expenses.
- Allows to enter details such as amount, date, and brief description.

Meeting Requirements:

- Helps users track expenses accurately and quickly.
- Provides detailed reports on expenses over time.

2. Category Selection

Functionality:

- Users can categorize expenses into different categories (e.g. dining, entertainment, bills).
- Provides the option to create custom categories according to individual needs.

Meeting Requirements:

- Helps users easily manage and analyze expenses by category.
- Enhances personal financial planning capabilities.

3. Summary View

Functionality:

- Provides an overview of expenses over a certain period of time.
- Display charts and statistics to help users quickly grasp the financial situation.

Meet the Requirements:

- Help users evaluate spending trends and adjust financial plans.
- Create an intuitive and easy-to-understand experience for users.

Additional Features

Manual Classification

Functionality:

- Allows users to categorize their expenses in their own way.
- Provides the option to add labels or tags to each expense.

Priority:

- Prioritized based on user feedback, indicating a high demand for personalization.
- Helps users feel more in control of their personal finances.

High Spending Notifications

Functionality:

- Warns users when spending exceeds a predetermined threshold.
- Provides notifications via app or email so users can make timely adjustments.

Priority:

- Highly rated by users, helping them become more aware of their financial situation.
- This feature not only helps users control their spending but also encourages them to plan their finances more effectively.

Data Management and Security

1. Data Management

Data Storage:

- User data is securely stored on the Firebase platform.
- The system automatically backs up data, ensuring no loss of important information.

Access Management:

- Provides flexible access rules, allowing access rights to be assigned to each user.
- Users can manage access rights for categories and sensitive information.

2. Data Security

User Authentication:

- Use Firebase Authentication to authenticate users through email, password, and other authentication methods (Google, Facebook, etc.).
- Ensure that only authenticated users can access personal data.

Data Encryption:

- Data is encrypted during transmission and storage, protecting information from external threats.
- Use modern encryption standards to ensure maximum security.

Monitoring and Intrusion Detection:

- Firebase provides monitoring and intrusion detection tools that help quickly identify and respond to unusual behavior.
- Users are notified immediately if there is suspicious activity related to their account.

Testing and User Feedback

1. Testing Process

- Conduct testing with a representative group of users.
- Use interviews and surveys to gather feedback.

2. Feedback Received

Positive:

Satisfied with expense logging and user-friendly interface.

Negative:

Difficulty finding features; workflow needs improvement.

3. Usability Changes

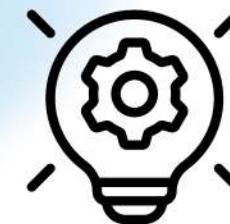
Adjust Button Location:

Move the "Record Expense" button to a more visible location.

Simplify the Process:

Combine similar steps into a single step.

Challenges Encountered



TECHNICAL CHALLENGES

Integration Challenges:

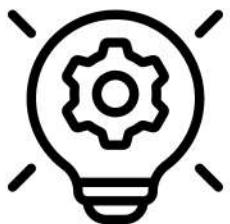
Difficulty integrating external APIs.

Solution: Conduct thorough research and test each step to ensure compatibility.

Performance Issues:

The application has difficulty handling large data.

Solution: Optimize the code and use more efficient storage methods.

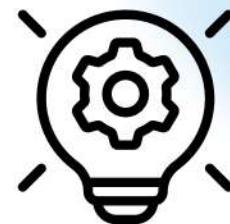


TIME CONSTRAINTS

Short Development Time:

Limited time to complete the project.

Solution: Plan in detail and divide the work among team members appropriately.



USER FEEDBACK

Requirement Changes:

Requirements change from users during the development process.

Solution: Maintain regular communication with users to make timely and flexible adjustments.

Risk Management

Identified Risks

Limited Mobile Experience

- Risk: Users may encounter usability challenges on mobile devices.
- Impact: Reduced user satisfaction and adoption.

Scope Creep

- Risk: Uncontrolled expansion of project requirements.
- Impact: Delayed timelines and increased costs.

Mitigation Strategies

- For Limited Mobile Experience:
 - Prioritized mobile-first design principles.
 - Conducted iterative user testing on mobile devices.
 - Engaged external mobile UX experts for review.

For Scope Creep:

- Established clear requirements and sign-off processes.
- Implemented a change management protocol.
- Regular stakeholder alignment meetings.

Project Management and Collaboration

Tool: GitHub

Purpose:

- Monitor development milestones through version control.
- Utilize GitHub Projects for Kanban-style tracking.
- Enable seamless integration with CI/CD pipelines to ensure smooth deployment.
- Provide a centralized and secure repository for project documents, codebases, and design assets.
- Facilitate version history to track changes and restore previous iterations when needed.
- Enable collaborative editing and pull request reviews.

Team Communication

Tool: Zalo

Purpose:

- Foster real-time communication for quick updates and collaboration.
- Create group chats for focused discussions.
- Share multimedia and voice messages to clarify complex issues.

Summary of User Requirements Fulfillment

Alignment with User Requirements and Functionality

Criteria:

- Does the app meet the defined user needs and functional requirements?
- Completeness and accuracy of core features.

Measurement Tools:

- Requirement traceability matrix.
- User acceptance testing (UAT) results.

User Satisfaction

Criteria:

- Ease of use, responsiveness, and overall user experience.
- Positive feedback from user surveys and interviews.
-

Measurement Tools:

- Net Promoter Score (NPS).
- Post-launch user feedback surveys.

Overall App Quality

Criteria:

- Performance, reliability, and scalability of the application.
- Frequency and severity of bugs or issues post-launch.
- Measurement Tools:
- Bug tracking reports.
- Load and stress testing results.

Lessons Learned

Insights into Mobile Development

- Importance of responsive design and cross-device compatibility.
- Strategies for performance optimization in resource-constrained environments.
- Leveraging user feedback to prioritize iterative improvements.
- Understanding of User-Centered Design
- Value of user personas in guiding design decisions.
- Conducting effective usability testing to identify pain points.
- Balancing aesthetics with functionality to deliver seamless user experiences.

Team Collaboration and Project Management

- Importance of clear communication and defined roles.
- Benefits of Agile methodology for adapting to evolving requirements.
- Maintaining flexibility while adhering to project timelines and goals.

Feedback Analysis and Future Improvements

Positive Feedback

- Ease of Use: Users appreciate the simplicity of tracking daily expenses and setting budgets.
- Comprehensive Reports: Detailed expenditure reports and analytics were highlighted as a key feature that users found very helpful.
- Budget Alerts: The notification feature for budget limits was well-received, helping users manage their finances better.

Constructive Feedback

- Mobile Performance: Some users reported that the app slowed down when managing large volumes of transactions.

User Feedback Integration

Added an in-app feedback form to gather user insights in real time, allowing us to prioritize future updates.

Suggested Improvements for Future Updates

Advanced Features

AI-Powered Budget Recommendations: Introduce AI-based recommendations for setting realistic budgets based on user spending patterns.

- Expense Categorization: Enhance automatic categorization of transactions to help users better track their spending habits (e.g., food, entertainment, utilities).

Thank you