



GROUP 3

THE CAMPUS EXPENSE MANAGER APPLICATION

MEMBERS

01

NGUYEN MINH HIEU

02

NGUYEN TIEN MANH

03

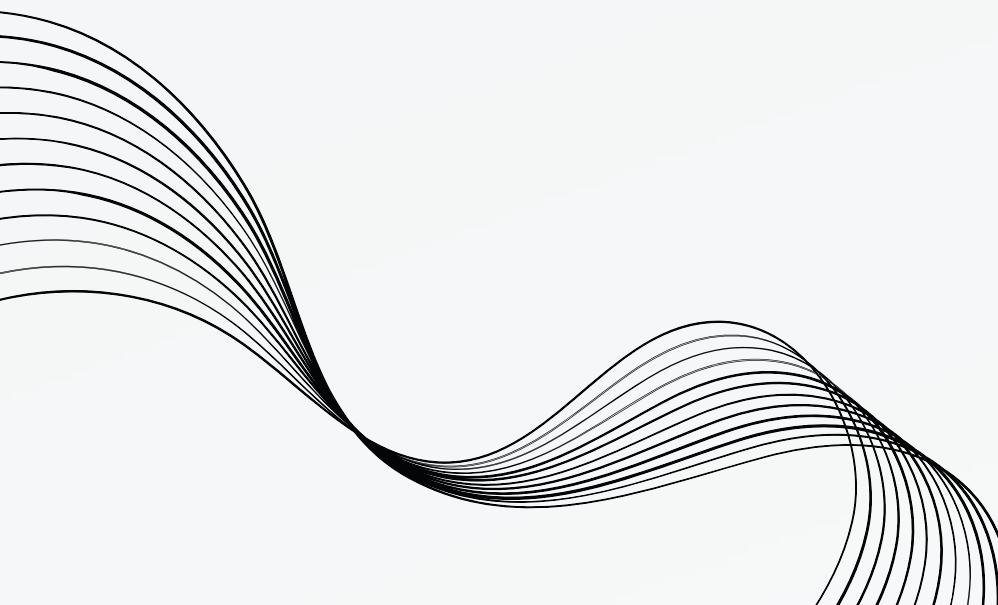
BUI HOANG LONG

04

PHAM XUAN BACH

PROJECT OVERVIEW

BudgetWise Solutions is a small development team with limited experience in mobile app development. They have taken on a project to create a mobile application called "CampusExpense Manager," tailored for university students. The goal is to develop an easy-to-use app that helps students manage their expenses effectively while staying within their budgets. The CampusExpense Manager aims to simplify expense tracking for students, whether they are living on or off-campus.



OBJECTIVES OF THE APP

Simplify Expense Tracking

- Easy input of expenses with categories and descriptions
- Quick search and filter functionality

Effective Budgeting

- Set monthly and category-specific budgets
- Receive real-time notifications on budget usage

Gain Financial Insights

- Visualize spending trends and patterns
- Generate detailed expense reports

Enhance User Experience

- Intuitive and user-friendly interface
- Reliable performance, even offline

USER REQUIREMENTS

Expense Tracking

- Ability to record daily expenses
- Categorization of expenses (e.g., food, rent, transportation)
- Search and filter functionality to analyze spending patterns

Budgeting

- Set monthly and category-specific budgets
- Receive timely budget alerts

Intuitive User Interface

- Easy-to-navigate design
- Clear and concise information display

Summary Reports

- Generate detailed expense reports
- Visualize spending trends and patterns

SYSTEMS INVESTIGATION AND RESEARCH

Research Phase:

To gain a comprehensive understanding of students' financial needs and preferences, we conducted a thorough research phase, including:

- **User Surveys and Interviews:** We directly engaged with university students to gather insights into their spending habits, budgeting challenges, and desired features in a financial management app.
- **Competitive Analysis:** We analyzed popular personal finance apps, such as Mint, PocketGuard, and YNAB, to identify their strengths, weaknesses, and unique selling points.

SYSTEMS INVESTIGATION AND RESEARCH

Key Insights from Research:

- **Simplicity and User-Friendliness:** Students desire a straightforward and easy-to-use interface.
- **Real-time Tracking:** The ability to track expenses in real-time, either manually or through automated imports.
- **Flexible Budgeting:** Customizable budgets that adapt to changing financial situations.
- **Offline Functionality:** The app should function without an internet connection, especially for students studying abroad or in areas with limited connectivity.
- **Security and Privacy:** Strong security measures to protect sensitive financial information.
- **Community Features:** Opportunities for peer-to-peer support, financial advice, and challenges.
- **Gamification:** Incorporating gamification elements to motivate users and make financial management fun.

PROJECT SCOPE AND CONSTRAINTS

Project Scope

The CampusExpense Manager app will provide the following core functionalities:

- Expense Tracking:
 - Manual input of expenses with categories and descriptions
 - Automatic categorization using machine learning (future enhancement)
- Budgeting:
 - Set monthly and category-specific budgets
 - Receive real-time budget alerts and notifications
- Reporting:
 - Generate detailed expense reports
 - Visualize spending trends and patterns
- User-Friendly Interface:
 - Intuitive design with clear navigation
 - Personalized dashboard

PROJECT SCOPE AND CONSTRAINTS

Project Constraints

- Limited Development Resources: The development team has limited experience in mobile app development and a small team size.
- Tight Timeline: The project has a strict 12-week timeline.
- Budget Constraints: Limited budget for development, testing, and marketing.
- Offline Functionality: The app must function offline to accommodate users in areas with limited connectivity.
- Platform Compatibility: The app must be compatible with both iOS and Android platforms.
- Data Security: Robust security measures must be implemented to protect user data.
- Performance Optimization: The app must perform efficiently, even on low-end devices.

TOOLS AND TECHNOLOGIES USED

01

02

03

04

SQLITE

Local database for offline data storage

ANDROID STUDIO

Integrated development environment for Android app development

JAVA FOR ANDROID

JAVA is a programming language which is used in Android App Development. The primary goals of JAVA is to be simple, object-oriented, robust, secure and high level.

FIGMA

A powerful design tool for creating user interfaces and prototyping

DEVELOPMENT METHODOLOGY

Agile Development has been chosen as the primary methodology for this project. This iterative approach aligns well with the dynamic nature of software development and the evolving requirements of our target audience.

Key Benefits of Agile Development:

- **Flexibility:** Agile allows for adaptability to changing requirements and unforeseen challenges.
- **Rapid Iteration:** Frequent iterations enable quick delivery of features and early feedback.
- **Collaboration:** Strong collaboration between the development team and stakeholders.
- **Continuous Improvement:** Continuous learning and improvement throughout the development process.

INITIAL DESIGN AND PROTOTYPING

User Research: Conducted interviews and surveys to understand user needs and pain points.

Wireframing: Created low-fidelity wireframes to visualize the app's basic structure and layout.

Prototyping: Developed high-fidelity prototypes using Figma to simulate the app's user interface and interaction.

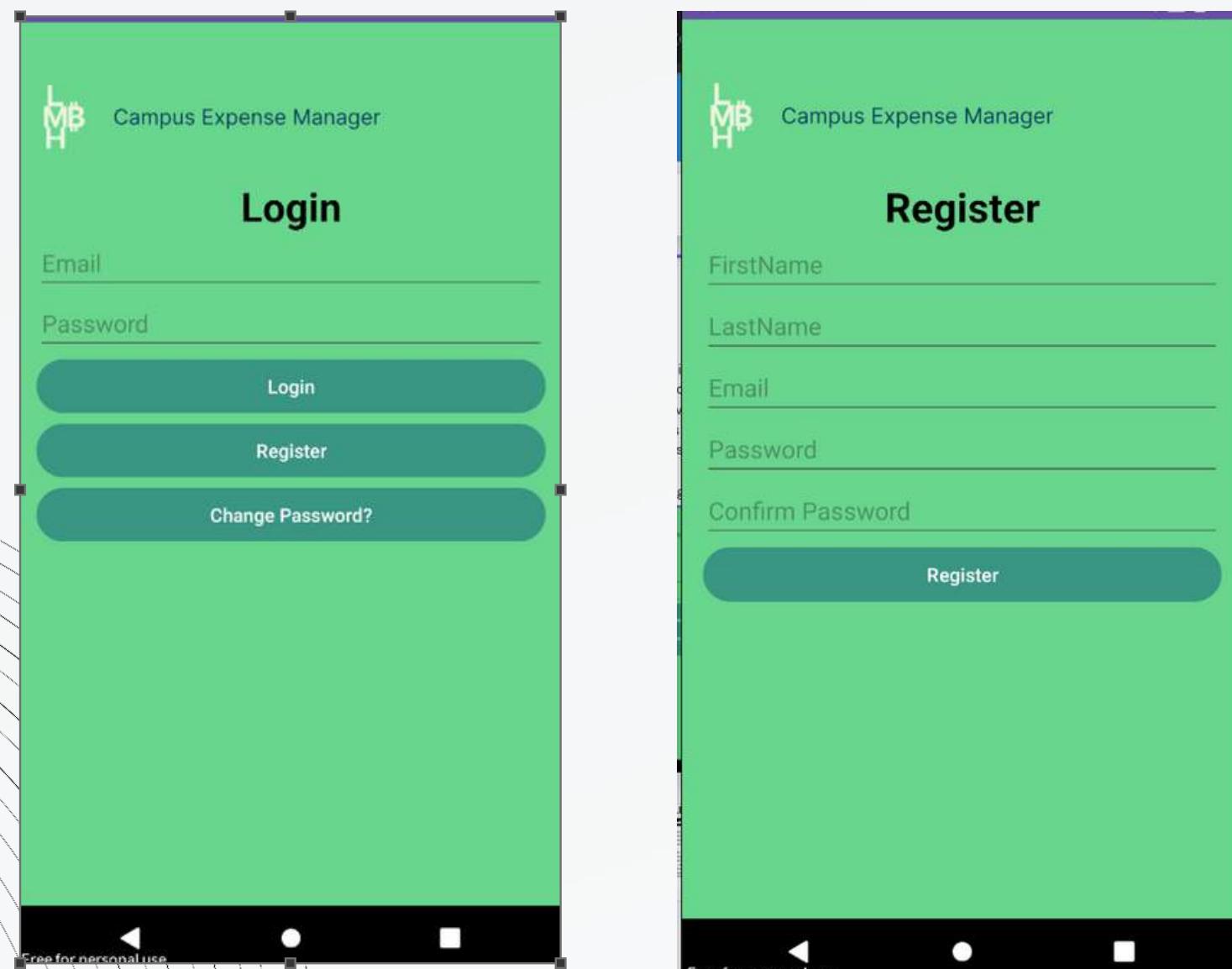
- User Feedback:
 - Conducted usability testing with a small group of students to gather feedback on the prototype.
 - Key feedback points:
 - Simplify the expense categorization process.
 - Improve the clarity of budget visualizations.
 - Enhance the overall app's visual appeal.

USER INTERFACE (UI) DESIGN

Design Principles:

- Simplicity: A clean and minimalist design to reduce cognitive load.
- Intuitiveness: Self-explanatory interface elements and a logical flow.
- Consistency: Consistent use of typography, color palette, and iconography.
- Accessibility: Adherence to accessibility guidelines to ensure inclusivity.

(UI) Design

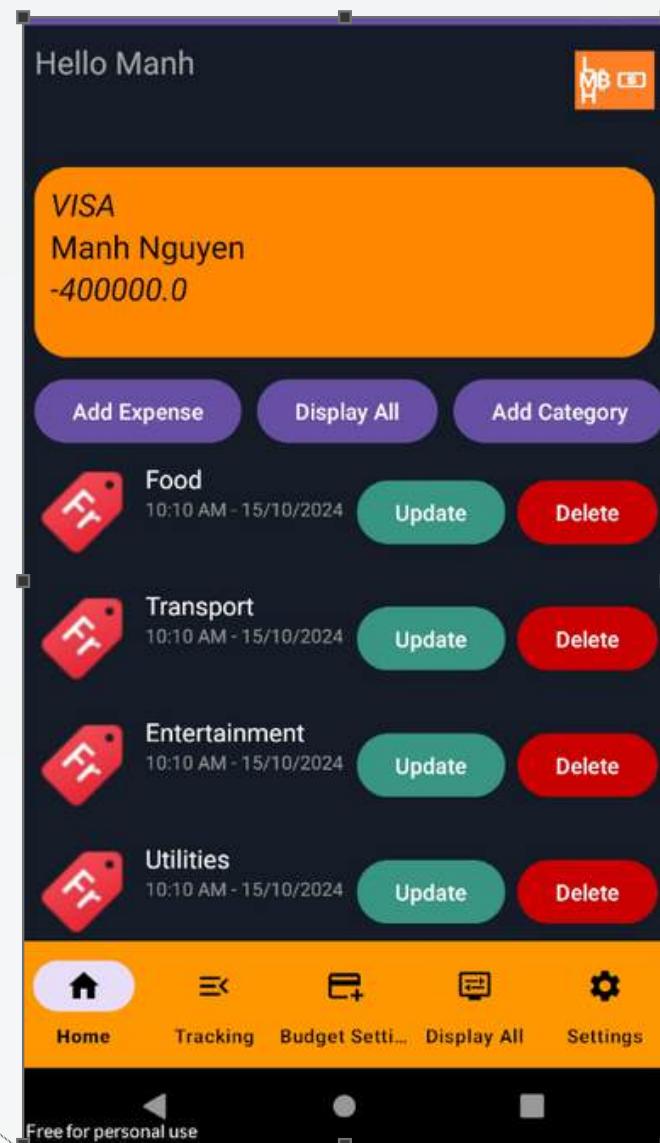


Design Rationale:

- Minimalistic Design: A clean and uncluttered interface reduces cognitive load and improves user experience.
- Clear Call-to-Action Buttons: Prominent buttons encourage user interaction.
- Error Handling: Informative error messages guide users to correct input mistakes.

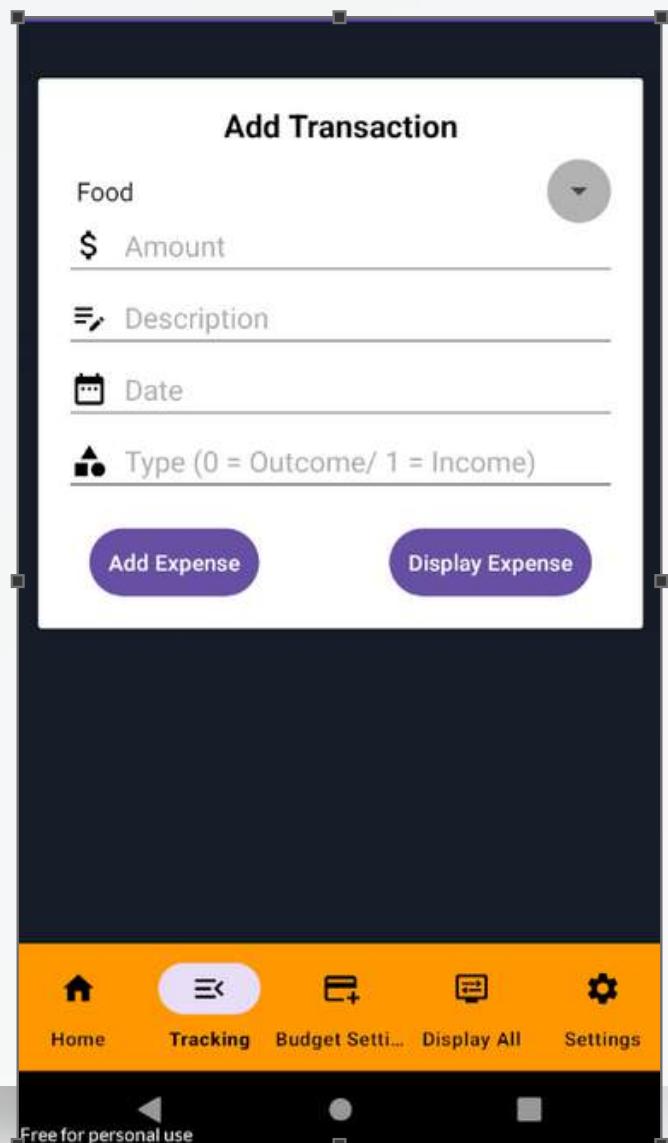
USER INTERFACE (UI) DESIGN

Home



- Clear display of current budget and spending.
- Quick access to expense tracking, budgeting, and reporting features.

Tracking expenses



- Simple form for inputting expenses.
- Categorization options for easy organization.

BACKEND ARCHITECTURE

OVERVIEW OF BACKEND CHOICES

The application uses SQLite as the primary data storage solution on mobile devices. SQLite is a standalone, serverless, high-performance SQL database engine, renowned for its simplicity and ease of integration. Designed to be embedded in applications, SQLite eliminates the need for separate database server processes and complex configurations.

Choosing SQLite helps minimize reliance on external services, especially when the application needs to store and process data directly on the user's device

SQLITE FOR USER DATA STORAGE

Local data storage: Spending data and related information such as spending categories, budgets, and reports will be stored directly on the mobile device in an SQLite database.

SQLite can store data in interconnected tables, supporting complex SQL queries to effectively manage user spending data.

Simple data management: SQLite is lightweight and easy to use, does not require a separate database server, which simplifies deployment and maintenance.

SCALABILITY SUPPORT

SQLite supports local data storage on mobile devices, helping applications operate smoothly even without an internet connection.

Expenditure data and related information will be stored in an SQLite database, and querying and updating data can be done quickly and effectively.

With the ability to process a moderate amount of data on the device, SQLite is very suitable for applications that need to store user information without requiring large-scale scalability or online synchronization.

CORE FEATURES IMPLEMENTED

User Registration and Authentication

Functional requirements:
User can create account using login and password name.

Fulfilling User Needs: The application provides an interface for authorized users to register and log in to personal accounts. The authentication process must be safe and secure, with users only able to access their spending data.

Expense Tracking

Functionality requirements:
Users can add, edit, and categorize expenses (e.g., rent, food, transportation).

Meet user needs: Each expense must include information such as description, date, amount, and type of expense. Users must be able to easily edit information if needed.

Budget Setting

Functional Requirement:
Users can set up monthly budgets for different spending categories (e.g., food, shopping, bank payment, service bills, education).

Fulfilling User Needs: The application allows users to adjust the budget amount for each spending category and updates when there are changes.

CORE FEATURES IMPLEMENTED

Expense Overview

Functional Requirement: The app provides a monthly spending summary, including total spending, remaining budget, and spending allocation by category.

Fulfilling User Needs: Users can view total spending, compare it with the set budget, and check spending details by each category. Can also show spending trends over time

Recurring Expenses

Functional Requirement: Users can add recurring expenses (for example, monthly rent) with start and end dates.

Fulfilling User Needs: Recurring expenses are automatically added to the user's budget every month, making expense management stable and easy.

Expense Reports

Functional Requirement: Users can create detailed spending reports for specific time periods (e.g. monthly, yearly).

Fulfilling User Needs: The spending report will include spending allocation by categories, helping users easily evaluate their personal financial situation.

CORE FEATURES IMPLEMENTED

Expense Notifications

Functional Requirement: The application will send reminders or notifications when the user nears or exceeds the budget limit of each category.

Fulfilling User Needs: The notification feature helps users control spending more effectively, avoiding spending beyond the established budget.

ADDITIONAL FEATURES

EXPLANATION OF EXTRA FEATURES

MANUAL CATEGORIZATION

Description: Users can create and categorize expenses into the categories they want, helping them customize spending tracking according to their personal needs.

Benefits: This feature helps users more closely control and classify spending easily, thereby making reasonable financial decisions.

NOTIFICATIONS FOR HIGH SPENDING

Description: The app sends notifications when users near or exceed budget limits in important spending categories.

Benefits: Helps users identify and adjust spending promptly, thereby avoiding exceeding the budget and maintaining a stable financial situation.

PRIORITIZATION BASED ON USER FEEDBACK

These additional features were prioritized for development based on direct feedback from users during testing.

Manual Categorization and Notifications for High Spending are two special features most requested by users because they help them control spending and adjust budgets proactively.

These features not only meet current needs but also assist users in optimizing their budget and managing long-term personal finances.

DATA MANAGEMENT AND SECURITY

DESCRIPTION OF HOW DATA IS
MANAGED AND SECURED

DATA MANAGEMENT

- User Data Management: User data, including personal information and expenses, is efficiently stored and managed in an SQLite database, ensuring quick processing and availability on mobile devices.
- Data Synchronization: Firebase provides data synchronization across devices, ensuring users can access and update their information anytime and anywhere without losing data.

DATA SECURITY

- Authentication and Security with Firebase: Firebase Authentication is used to securely authenticate users. Each user can only access their own data after successfully logging in.
- Data Encryption: Sensitive data, such as passwords and payment information, is encrypted to ensure that even if data is accessed improperly, personal information remains protected.

DATA MANAGEMENT AND SECURITY

DESCRIPTION OF HOW DATA IS
MANAGED AND SECURED

FIREBASE SECURITY FEATURES

- Role-based Access Control: Firebase uses security rules based on roles, granting users access only to the resources they are authorized to view.
- Data Protection During Transmission: Firebase ensures that data is encrypted during transmission, safeguarding user information from network attacks.

DATA PRIVACY AND COMPLIANCE

- Regulatory Compliance: The app complies with data protection regulations like GDPR and other security standards to ensure user privacy.
- User Rights for Data Access and Management: Users have the right to request access, modify, or delete their personal data at any time.

USER TESTING AND FEEDBACK

Summary of User Testing

Vision



The app was tested by a diverse group of university students, ranging from first-year to fourth-year, to ensure varied perspectives and use cases. A total of 40 students participated in the testing phase. They explored the app's key features, including user registration, expense tracking, budget setting, and notifications, to evaluate its usability and functionality.

Students were assigned specific scenarios, such as registering an account, adding expenses, generating reports, and responding to budget notifications. These scenarios allowed for comprehensive testing of the app's functional requirements.

USER TESTING AND FEEDBACK

Feedback Overview

Positive Feedback:

- **Expense Tracking:** Most students appreciated the straightforward process of logging expenses. The ability to edit and categorize expenses made the feature highly practical for everyday use.
- **Budget Setting:** The budget feature was praised for helping students better manage their spending habits. They particularly liked the flexibility to adjust budget amounts as needed.
- **Expense Overview:** The summary dashboard received positive remarks for its clear presentation of spending trends and category breakdowns, which made it easier to track finances.

Negative Feedback:

- **Recurring Expenses:** Several students found the recurring expense setup confusing, especially when selecting start and end dates.
- **Notifications:** While notifications were helpful, some students found them too frequent, which became annoying over time.
- **Navigation:** First-year students struggled with navigating between features. They noted that the interface could be more beginner-friendly.

USER TESTING AND FEEDBACK

Usability Changes Based on Feedback

To address the issues raised during testing, several improvements were made:

1. Recurring Expenses: The recurring expense form was simplified by introducing preset durations (e.g., monthly, yearly), reducing the need for manual date selection.
2. Notification Frequency: A customizable notification setting was added, allowing users to control how often they receive alerts about budget thresholds.
3. Improved Navigation: Onboarding tutorials were introduced to guide first-time users through the app's key features. Additionally, clearer icons and labels were added to improve overall navigation.
4. User Interface Enhancements: Key action buttons, such as "Add Expense" and "Set Budget," were repositioned for easier access. The interface was also decluttered by minimizing redundant options and prioritizing essential features.

CHALLENGES FACED

TECHNICAL DIFFICULTIES

Challenge:
Integrating a new system with these existing applications
can be complex and may lead to data compatibility issues.

Solution:
Custom integration required, including data mapping and transformation processes.
Using middleware solutions helped facilitate smoother integration.

TIME CONSTRAINTS

Challenge:
Increasing the risk of bugs and user dissatisfaction.

Solution:
Breaking the project into smaller, manageable sprints.

BUDGET CONSTRAINTS

Challenge:
Limited budgets can restrict the resources available for development, testing, and implementation.

Solution:
They sought feedback to identify which features provided the most value, allowing for better allocation of resources.

QUALITY ASSURANCE AND TESTING

FUNCTIONAL TESTING

Objective: To verify that each feature of the application functions according to the specified requirements.

Approach:

This phase is usually conducted by developers.

This helps identify issues arising from the combination of different modules.

This includes testing all functionalities in a staging environment that mimics production.

The changes do not introduce new bugs or affect existing functionalities.

USABILITY TESTING

Objective: To evaluate the application's user interface and overall user experience.

Approach:

Feedback is collected on ease of use, navigation, and overall satisfaction.

Identify potential usability issues.

which performs better in terms of user satisfaction and effectiveness.

QUALITY ASSURANCE AND TESTING

KEY RESULTS THAT ENSURED APP RELIABILITY AND STABILITY.

Improved Functionality

This ensured that all features operated seamlessly, significantly enhancing the application's reliability.

Enhanced User Experience

Usability testing revealed that users found certain navigation paths confusing. Based on this feedback, the team made significant adjustments to the user interface, resulting in a more intuitive experience. Post-implementation surveys showed a 30% increase in user satisfaction scores.

QUALITY ASSURANCE AND TESTING

KEY RESULTS THAT ENSURED APP RELIABILITY AND STABILITY.

Higher Stability

Regression testing helped identify vulnerabilities that emerged during the addition of new features. These issues were addressed before deployment, leading to a more stable application with fewer post-launch defects.

Enhanced User Experience

Usability testing revealed that users found certain navigation paths confusing. Based on this feedback, the team made significant adjustments to the user interface, resulting in a more intuitive experience. Post-implementation surveys showed a 30% increase in user satisfaction scores.

RISK MANAGEMENT

IDENTIFIED RISKS

1. LIMITED MOBILE EXPERIENCE

- DESCRIPTION: RISK OF SUBOPTIMAL PERFORMANCE ON MOBILE DEVICES, LEADING TO USER DISSATISFACTION.
- IMPACT: COULD REDUCE ADOPTION RATES AND EFFICIENCY FOR MOBILE USERS.

2. SCOPE CREEP

- DESCRIPTION: UNCONTROLLED EXPANSION OF PROJECT SCOPE DUE TO NEW FEATURE REQUESTS.
- IMPACT: MAY CAUSE DELAYS AND INCREASED COSTS, COMPROMISING PROJECT OBJECTIVES.

RISK MANAGEMENT

MITIGATION STRATEGIES

1. LIMITED MOBILE EXPERIENCE

- USER-CENTERED DESIGN: INVOLVED USERS IN THE DESIGN AND TESTING PHASES TO ENHANCE MOBILE USABILITY.
- RESPONSIVE DESIGN: DEVELOPED THE APP TO ADAPT TO VARIOUS SCREEN SIZES, ENSURING FUNCTIONALITY ACROSS DEVICES.
- PROTOTYPING AND TESTING: CREATED MOBILE PROTOTYPES AND CONDUCTED USABILITY TESTS TO IDENTIFY AND RESOLVE ISSUES EARLY.
- PERFORMANCE OPTIMIZATION: FOCUSED ON REDUCING LOAD TIMES AND OPTIMIZING RESOURCE USE FOR MOBILE ACCESS.

1. SCOPE CREEP

- CLEAR REQUIREMENTS DEFINITION: ESTABLISHED DETAILED PROJECT REQUIREMENTS TO GUIDE DEVELOPMENT.
- CHANGE CONTROL PROCESS: IMPLEMENTED A FORMAL PROCESS FOR DOCUMENTING AND APPROVING CHANGES TO THE PROJECT SCOPE.
- REGULAR STAKEHOLDER MEETINGS: CONDUCTED FREQUENT CHECK-INS TO ALIGN WITH STAKEHOLDERS AND MANAGE EXPECTATIONS.
- FEATURE PRIORITIZATION: FOCUSED ON DELIVERING THE MOST CRITICAL FUNCTIONALITIES FIRST TO KEEP THE PROJECT ON TRACK.

PROJECT MANAGEMENT AND COLLABORATION

TOOLS USED

1. PROJECT MANAGEMENT SOFTWARE

- TRELLO/JIRA: USED FOR TASK MANAGEMENT, WHERE TASKS ARE ORGANIZED INTO BOARDS OR SPRINTS. TEAM MEMBERS CAN ASSIGN, PRIORITIZE, AND TRACK TASK PROGRESS IN REAL TIME.
- ASANA: EMPLOYED FOR TRACKING PROJECT MILESTONES AND DEADLINES, ALLOWING FOR EASY VISIBILITY INTO ONGOING TASKS AND RESPONSIBILITIES.

1. COMMUNICATION TOOLS

- SLACK: UTILIZED FOR REAL-TIME COMMUNICATION, ENABLING QUICK DISCUSSIONS, FILE SHARING, AND INTEGRATION WITH OTHER TOOLS FOR NOTIFICATIONS AND UPDATES.
- MICROSOFT TEAMS: USED FOR VIDEO CONFERENCING AND TEAM MEETINGS, FACILITATING COLLABORATION ACROSS DIFFERENT LOCATIONS.

PROJECT MANAGEMENT AND COLLABORATION

VERSION CONTROL SYSTEM

GIT/GITHUB: IMPLEMENTED FOR SOURCE CODE MANAGEMENT, ALLOWING TEAM MEMBERS TO COLLABORATE ON CODE, TRACK CHANGES, AND MANAGE DIFFERENT VERSIONS EFFECTIVELY.

DOCUMENTATION TOOLS

- CONFLUENCE/GOOGLE DOCS: USED FOR CREATING AND SHARING PROJECT DOCUMENTATION, MEETING NOTES, AND DESIGN SPECIFICATIONS, ENSURING THAT ALL TEAM MEMBERS HAVE ACCESS TO UP-TO-DATE INFORMATION.

PROJECT MANAGEMENT AND COLLABORATION

STRATEGIES FOR EFFECTIVE COLLABORATION

1. AGILE METHODOLOGY

- THE TEAM ADOPTED AGILE PRACTICES, BREAKING THE PROJECT INTO SMALLER SPRINTS. THIS ALLOWED FOR ITERATIVE DEVELOPMENT, FREQUENT REASSESSMENT OF PROGRESS, AND ADAPTATION TO CHANGES BASED ON FEEDBACK.

2. REGULAR STAND-UP MEETINGS

- DAILY STAND-UP MEETINGS WERE HELD TO DISCUSS PROGRESS, ADDRESS ANY ROADBLOCKS, AND ALIGN ON PRIORITIES. THIS QUICK CHECK-IN HELPED MAINTAIN FOCUS AND ACCOUNTABILITY.

PROJECT MANAGEMENT AND COLLABORATION

STRATEGIES FOR EFFECTIVE COLLABORATION

3. CLEAR ROLE DEFINITIONS

EACH TEAM MEMBER'S ROLE AND RESPONSIBILITIES WERE CLEARLY DEFINED FROM THE OUTSET, REDUCING CONFUSION AND ENSURING ACCOUNTABILITY FOR SPECIFIC TASKS.

4. COLLABORATIVE TOOLS INTEGRATION

INTEGRATION OF TOOLS (E.G., LINKING TRELLO WITH SLACK) ENABLED SEAMLESS UPDATES AND NOTIFICATIONS, KEEPING EVERYONE INFORMED ABOUT TASK STATUSES WITHOUT EXCESSIVE MANUAL CHECKING.

PROJECT MANAGER PERFORMANCE EVALUATION

CRITERIA MANAGEMENT AND COLLABORATION

ALIGNMENT WITH USER REQUIREMENTS

REQUIREMENT COVERAGE: MEASURED BY THE PERCENTAGE OF USER REQUIREMENTS THAT WERE SUCCESSFULLY IMPLEMENTED. THIS INCLUDES BOTH FUNCTIONAL AND NON-FUNCTIONAL

REQUIREMENTS.

VALIDATION TESTING: CONDUCTED TO VERIFY THAT THE APPLICATION MEETS SPECIFIED REQUIREMENTS THROUGH USER ACCEPTANCE TESTING (UAT) AND FEEDBACK SESSIONS.

FUNCTIONALITY

FEATURE COMPLETENESS: EVALUATED BASED ON THE NUMBER OF IMPLEMENTED FEATURES COMPARED TO THE INITIAL PROJECT SCOPE. THIS INCLUDES CORE FUNCTIONALITIES LIKE EXPENSE SUBMISSION, APPROVAL WORKFLOWS, AND REPORTING CAPABILITIES.

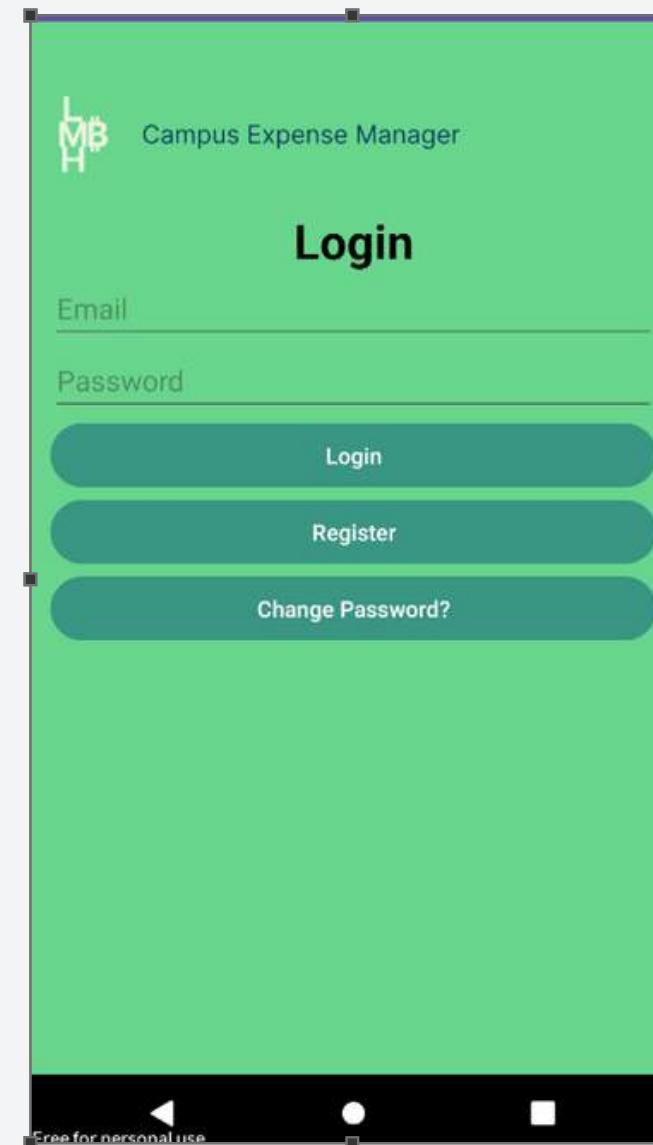
ERROR RATE: MONITORED BY TRACKING THE NUMBER OF BUGS REPORTED DURING TESTING PHASES AND POST-LAUNCH. A LOWER ERROR RATE INDICATES HIGHER FUNCTIONALITY RELIABILITY.

SUMMARY OF USER REQUIREMENTS FULFILLMENT

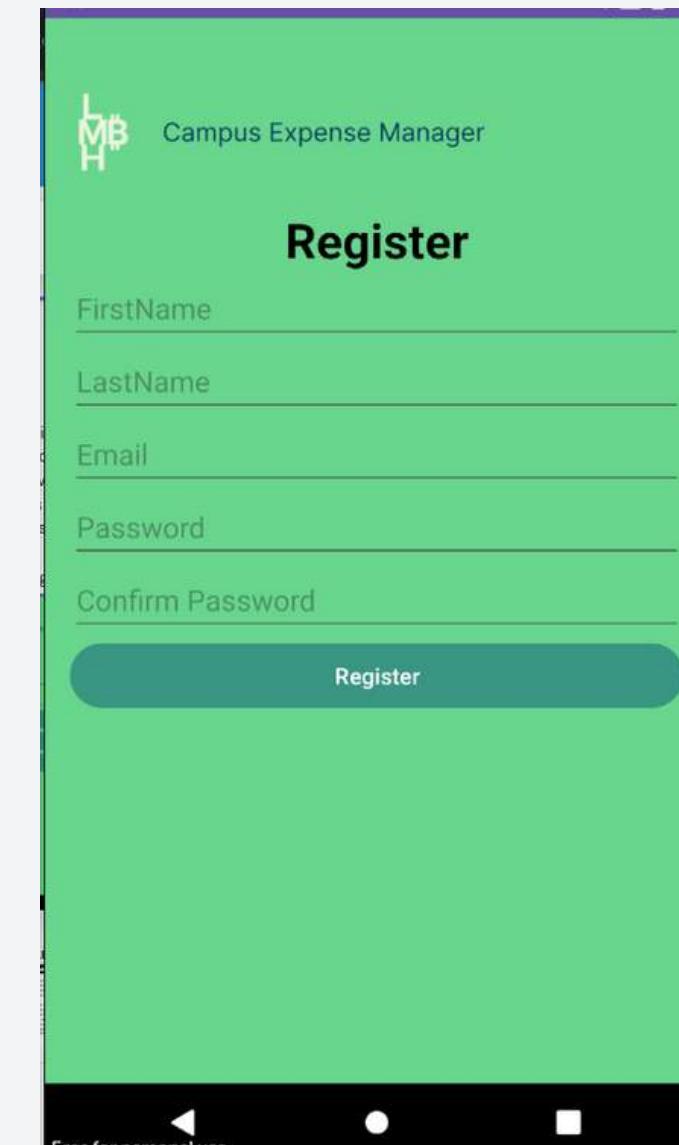


User Registration & Authentication

Deployment Feature: Create an account and securely log in to access users' personal expense data



The login screen for the Campus Expense Manager. It features a logo with 'L', 'M', and 'B' in a stylized font, followed by the text 'Campus Expense Manager'. Below this is a 'Login' button. The main area contains two input fields: 'Email' and 'Password', each with a corresponding label above it. Below the password field is a 'Login' button. At the bottom of the screen are three buttons: 'Register', 'Change Password?', and a small link that says 'Free for personal use'.



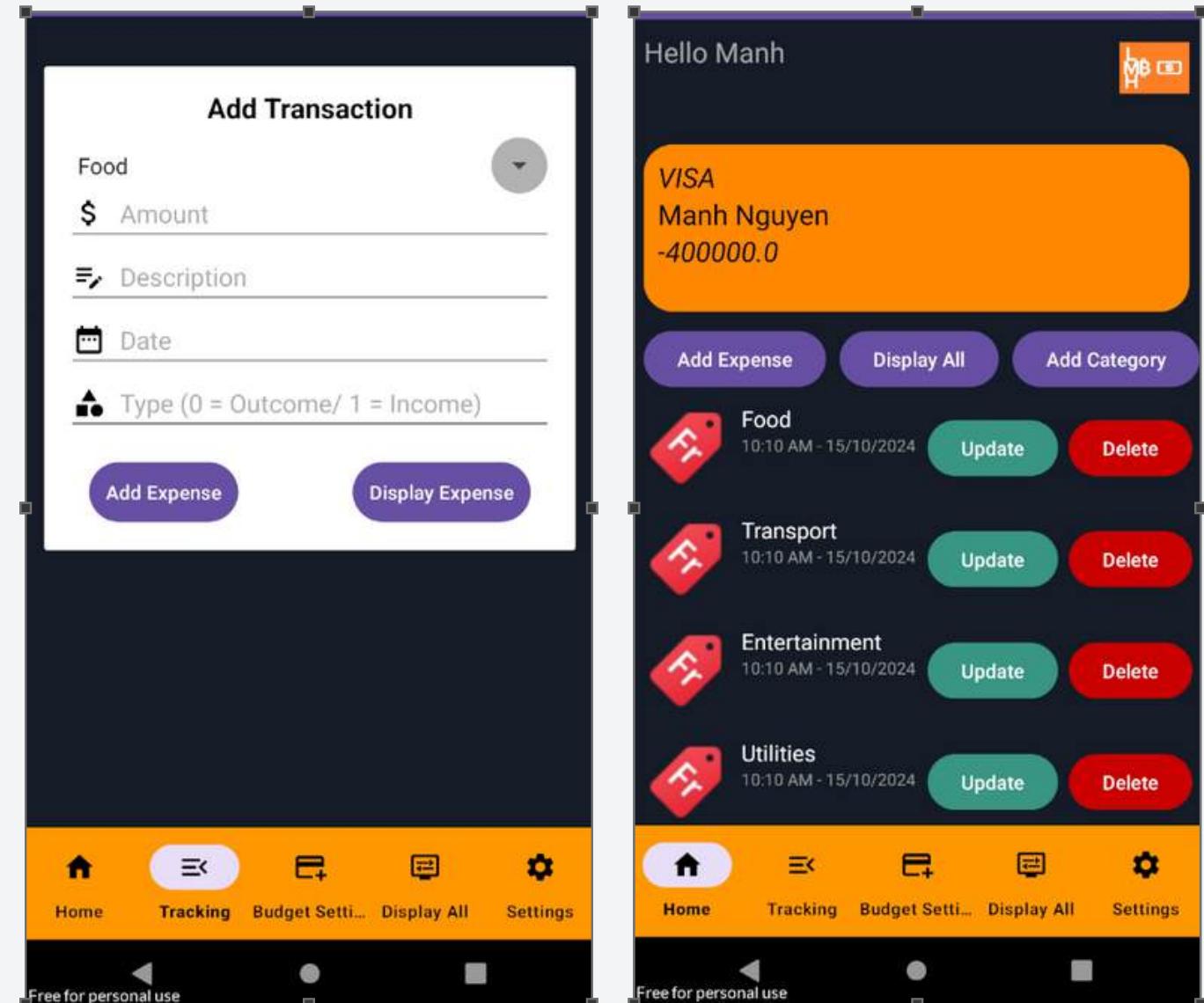
The register screen for the Campus Expense Manager. It features a logo with 'L', 'M', and 'B' in a stylized font, followed by the text 'Campus Expense Manager'. Below this is a 'Register' button. The main area contains four input fields: 'FirstName', 'LastName', 'Email', and 'Confirm Password', each with a corresponding label above it. Below the 'Confirm Password' field is a 'Register' button. At the bottom of the screen are three buttons: 'Login', 'Change Password?', and a small link that says 'Free for personal use'.

Users enter information fields such as email, password, firstname, lastname to register an account.

Users log in with their registered email and password to log into the system.

Expense Tracking

Deployment Feature: Set and adjust monthly budgets for categories (e.g., food, entertainment).



Users enter item quantity, description, date, type, email to add and adjust monthly budgets for categories (e.g. food, entertainment).

The spending items added by the user will be displayed on the home screen.

Lessons Learned

1. Skills Acquired

Technical Skills: Improved proficiency in mobile development frameworks and tools.

Design Skills: Gained experience in creating user-centered designs that prioritize usability.

2. Insights into Mobile Development

User Feedback: The importance of incorporating user feedback into the development process to refine features.

Testing: Understanding the necessity of thorough testing to ensure app stability and performance.

3. Understanding User-Centered Design

Empathy in Design: Recognizing the value of designing with the user's perspective in mind.

Iterative Process: Learning that design is an iterative process, where continuous improvement is essential based on user interaction and feedback.

Feedback Analysis and Future Improvements

1. Feedback Review

Positive Feedback:

Users appreciated the intuitive design and ease of navigation.

The budgeting feature was highlighted as particularly useful.

Constructive Criticism:

Some users had trouble syncing data across multiple devices.

Requested more detailed analytics and reporting options.

2. Actions Taken Based on Feedback

Data Sync Improvements:

Implemented improvements to ensure seamless data syncing across all devices.

Advanced Reporting Features:

Introduced new analytics options to provide users with deeper insights into their spending habits.

Feedback Analysis and Future Improvements

3. Suggested Improvements for Future Updates

User Education:

Develop instructional videos or tutorials to help users maximize the app's features.

Customization Options:

Allow users to customize their dashboards to display information that best suits their needs.

Gather regular feedback:

Set up a system for ongoing user feedback to continually improve the app based on user needs.

Conclusion

In conclusion, our project successfully launched an app that meets all initial user requirements, featuring well-received elements like expense tracking, budgeting, and reporting that enhance user experience.

Our team's exceptional collaboration, commitment, and resilience allowed us to overcome challenges and adapt to feedback, ensuring a high-quality product.

We are grateful for the opportunity to present this project and extend our thanks to all stakeholders, mentors, and users for their valuable insights and support throughout this journey.



Thank for watching