The CampusExpense Manager Mobile Application

Agenda

- Stakeholders
- Functional and Non-functional Requirements
- Favourable and Challenges in Project Implementation
- Platforms and Essential Features of the Software
- Analyze potential risks and risks management

Budget

TOTAL PROJECT DURATION: 12 WEEKS

TOTAL BUDGET: $17,375

1. **DEVELOPMENT COSTS:** $8,000
   - JUNIOR DEVELOPER SALARIES: $4,500 (3 DEVELOPERS, 3 MONTHS)
   - UX/UI DESIGNER SALARY: $3,000 (1 DESIGNER, 3 MONTHS)
   - DEVELOPMENT SOFTWARE & TOOLS: $500 (OPTIONAL PLUGINS/LIBRARIES)

2. **TESTING AND QUALITY ASSURANCE (QA):** $1,250
   - INTERNAL QA TESTING: $750 (75 HOURS)
   - EXTERNAL TESTING WITH BETA USERS: $400 (20 TESTERS)
   - BUG FIXES AND POST-LAUNCH SUPPORT: $1,000

3. **MARKETING AND PROMOTION:** $2,125
   - SOCIAL MEDIA CAMPAIGN: $1,500
   - UNIVERSITY COLLABORATIONS: $500
   - APP STORE REGISTRATION FEES: $125

4. **TRAINING AND SKILL DEVELOPMENT:** $2,000
   - ONLINE COURSES FOR DEVELOPERS: $600
   - MENTORSHIP PROGRAM: $600
   - INTERNAL WORKSHOPS: $800

5. **DATA SECURITY AND COMPLIANCE:** $2,000
   - ENCRYPTION TOOLS: $1,000
   - PRIVACY COMPLIANCE CERTIFICATION: $1,000

6. **CONTINGENCY FUND:** $2,000

Project Timeline

12-WEEK PROJECT TIMELINE OUTLINE

WEEKS 1-2: PLANNING & INITIAL DESIGN

- PROJECT KICKOFF: DEFINE REQUIREMENTS, ASSIGN ROLES, AND FINALIZE SCOPE
- UX/UI DESIGN: WIREFRAMES FOR KEY SCREENS (REGISTRATION, EXPENSE ENTRY, BUDGET OVERVIEW)
- TECHNICAL SETUP: CHOOSE TECHNOLOGY STACK, SET UP VERSION CONTROL, AND INITIATE APP STRUCTURE IN THE FRAMEWORK
- APPROVAL: REVIEW DESIGN AND TECH SETUP WITH STAKEHOLDERS

WEEKS 3-6: CORE FEATURE DEVELOPMENT

- **WEEK 3:** USER AUTHENTICATION & REGISTRATION
  - IMPLEMENT SIGN-UP, LOGIN, AND PASSWORD RESET FUNCTIONALITIES
  - DEVELOP SECURITY MEASURES FOR AUTHENTICATION
- **WEEK 4:** EXPENSE TRACKING MODULE
  - CREATE FUNCTIONALITIES TO ADD, EDIT, CATEGORIZE EXPENSES WITH DATE AND AMOUNT
- **WEEK 5:** BUDGET SETTING & TRACKING MODULE
  - ALLOW USERS TO SET MONTHLY BUDGETS PER CATEGORY AND TRACK SPENDING
- **WEEK 6:** EXPENSE OVERVIEW DASHBOARD
  - BUILD A SUMMARY PAGE DISPLAYING SPENDING BREAKDOWN AND BUDGET STATUS

Project Timeline

WEEKS 7-8: ADVANCED FEATURE DEVELOPMENT

- **WEEK 7:** RECURRING EXPENSES & EXPENSE NOTIFICATIONS
  - ENABLE USERS TO SET RECURRING EXPENSES, ADD NOTIFICATIONS FOR BUDGET LIMITS
- **WEEK 8:** REPORT GENERATION
  - DEVELOP FEATURES FOR GENERATING MONTHLY/ANNUAL EXPENSE REPORTS BY CATEGORY

WEEKS 9-10: TESTING & QUALITY ASSURANCE

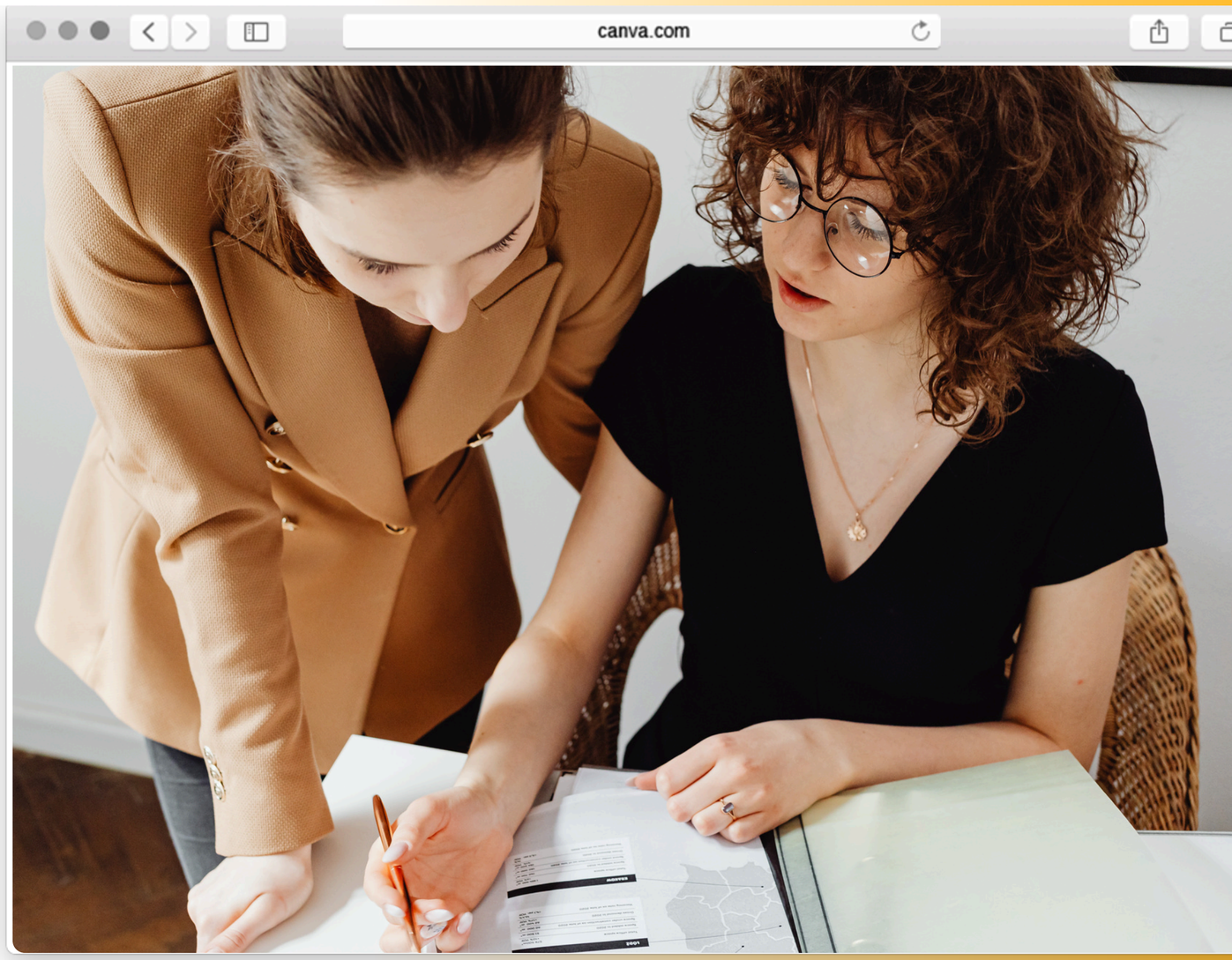
- **WEEK 9:** INTERNAL TESTING
  - MANUAL TESTING ON BOTH ANDROID AND IOS; FIX UI ISSUES AND BUGS
- **WEEK 10:** BETA TESTING WITH UNIVERSITY STUDENTS
  - GATHER USER FEEDBACK AND IMPLEMENT CRITICAL CHANGES
  - POST-BETA BUG FIXES: RESOLVE ISSUES DISCOVERED DURING BETA TESTING

WEEKS 11-12: FINAL TESTING, ADJUSTMENTS, & LAUNCH

- **WEEK 11:** FINAL TESTING AND PERFORMANCE OPTIMIZATION
  - ENSURE APP PERFORMS SMOOTHLY, ESPECIALLY WITH LARGE DATASETS
- **WEEK 12:** LAUNCH PREPARATION
  - PREPARE APP STORE LISTINGS, SUBMIT THE APP FOR REVIEW, AND INITIATE MARKETING CAMPAIGNS
- **LAUNCH DAY:** MAKE APP LIVE AND MONITOR PERFORMANCE FOR POST-LAUNCH SUPPORT

Stakeholder

and what they want in this project



Students

Student Representative: Nguyen Duc Khanh Huy

EFFECTIVE EXPENSE MANAGEMENT

It is important to track daily expenses and keep your budget on track.

SIMPLE AND USER-FRIENDLY FEATURES

The application should be easy to use, with an intuitive interface.

EXPENSE REPORT

Want to see an overview and spending trends over time

REMIND

Get notified when you're close to or over your budget

BudgetWise Solutions Team

Technical Team Leader: Tran Tuan Thanh
Mobile Developer: Tran Viet Anh
UI/UX Expert: Ngo Phuong Nam
Quality Assurance: Nguyen Manh Hung
Security Expert: Nguyen Tien Nam

COMPLETED ON TIME

Need to ensure application development and launch on schedule.

COST LIMIT

Must optimize resources and budget because the project has a limited budget

IMPROVE MOBILE APPLICATION DEVELOPMENT CAPABILITIES

Members need to learn and get familiar with new tools and technologies

USER FEEDBACK

Need a mechanism to collect user feedback to improve and develop more features

University Administrators

Principal: Mr. Vu Chi Thanh
Information Technology Department: Mr. Dinh Van Dong

ENCOURAGE STUDENTS TO BETTER MANAGE THEIR PERSONAL FINANCES

Want students to be conscious about spending and keep financial stability.

STUDENT SUPPORT TOOLS

This app could be introduced as part of student support services or in life skills courses.

Difficulties that BudgetWise Solutions may encounter when starting a project

Lack of experience in mobile application development

Time and budget constraints

Difficulty in ensuring security and regulatory compliance

Ensures good performance and user-friendly interface

Challenges in cross-platform testing and deployment

Functional and non-functional requirements in the system

Here’s a structured overview of the **functional** and **non-functional requirements** specifically aligned with the **CampusExpense Manager** mobile application system:

Functional Requirements

User Registration and Authentication

- **Account Creation:** Users can sign up by providing a username, email, and password.
- **Login:** Users can log in securely using their credentials.
- **Password Recovery:** Users can reset their passwords via email verification.

Expense Tracking

- **Add Expense:** Users can input new expenses with fields for description, date, amount, and category.
- **Edit Expense:** Users can modify existing expense entries.
- **Delete Expense:** Users can remove expenses from their records.
- **Categorization:** Users can categorize expenses (e.g., food, rent, entertainment).

Functional Requirements

Budget Setting

- **Set Budgets:** Users can define monthly budgets for different expense categories.
- **Adjust Budgets:** Users can modify budget amounts as needed.

Recurring Expenses

- **Setup Recurring Expenses:** Users can create recurring expenses with defined start and end dates.
- **Automatic Addition:** Recurring expenses should automatically be included in the monthly budget.

Expense Overview

- **Monthly Summary:** Users can view a summary of total spending, remaining budget, and breakdown by category.
- **Trends Visualization:** Users can see graphs or charts representing spending trends over time.

Expense Reports

- **Generate Reports:** Users can generate detailed expense reports for specific time periods (e.g., weekly, monthly, annually).
- **Categorized Breakdown:** Reports should display expenses categorized for easier analysis.

Functional Requirements

EXPENSE NOTIFICATIONS

- **BUDGET ALERTS:** THE APP SENDS NOTIFICATIONS WHEN USERS APPROACH OR EXCEED THEIR SET BUDGET LIMITS.
- **RECURRING REMINDERS:** USERS RECEIVE REMINDERS FOR UPCOMING RECURRING EXPENSES.

DATA BACKUP AND RESTORE

- **BACKUP:** USERS CAN BACK UP THEIR EXPENSE DATA TO A SECURE CLOUD SERVICE.
- **RESTORE:** USERS CAN RESTORE THEIR DATA FROM BACKUPS WHEN NEEDED.

Non-Functional Requirements

Non-Functional Requirements

Performance

- **Response Time:** The app should respond to user actions within 2 seconds.
- **Load Time:** The app should load within 3 seconds on a standard connection.

Platform Compatibility

- **Cross-Platform:** The app should function seamlessly on both Android and iOS devices.
- **Responsive Design:** The interface should adapt to various screen sizes and orientations.

User-Friendly Interface

- **Intuitiveness:** The UI should be intuitive, requiring minimal training for new users.
- **Accessibility:** The design should accommodate users with disabilities (e.g., screen reader compatibility).

Data Security

- **Encryption:** User data should be encrypted both in transit and at rest.
- **Compliance:** The app must comply with relevant data privacy regulations (e.g., GDPR, CCPA).

Non-Functional Requirements

Feedback and Support

- **In-App Feedback:** Users should have an easy way to submit feedback or report issues.
- **Support Response:** The development team should acknowledge and respond to user feedback within 48 hours.

Scalability

- **User Capacity:** The system should handle a growing number of users without performance degradation.
- **Data Volume:** The app should efficiently manage increasing amounts of expense data.

Offline Capability

- **Functionality Without Internet:** Users should be able to enter and view expenses without an internet connection.
- **Automatic Sync:** Data should sync automatically once the device is back online.

Maintainability

- **Code Quality:** The code should be modular and well-documented to simplify future updates and maintenance.
- **Error Handling:** The app should gracefully handle errors and provide informative messages to users.

Potential Challenges in Project Implementation

INEXPERIENCED STAFF

New team members may need significant training and mentoring to gain the necessary skills and knowledge.
Lack of experience can lead to mistakes, which can slow down the project and increase costs.
Inexperienced employees may take longer to complete tasks, affecting the overall efficiency of the project

NEW TEAM

A new team may not have clearly defined processes and workflows, leading to inefficiencies and misunderstandings.
New team members may have difficulty communicating effectively, hindering collaboration and teamwork.
The challenges associated with establishing a new team can negatively impact team morale and productivity.

POORLY CALCULATED BUDGET

Underestimating costs can lead to budget shortfalls, forcing difficult decisions like cutting features or delaying the project.
Unforeseen costs can arise, leading to budget overruns and the project potentially failing.
To stay within budget, the team may be forced to cut costs, affecting the quality of the final product.

To minimize these difficulties, consider the following strategies:

- Invest in comprehensive training programs for new team members.
- Establish effective communication channels and encourage open dialogue.
- Create detailed project plans with realistic timelines and milestones.
- Develop contingency plans to address unexpected challenges and budget overruns.
- Conduct regular project reviews to track progress and identify potential issues early.
- Foster a positive team culture through team-building exercises and social events.

Platforms and Essential Features of the Software

Take advantage
of the tools available.

Compare between Mobile vs Desktop

Feature/Aspect	Desktop Application	Mobile Application
Accessibility	Requires a specific location with a computer.	Accessible anytime, anywhere on smartphones.
User Engagement	Limited engagement; less frequent use.	High engagement with push notifications and reminders.
Target Audience	Primarily caters to users who work from a desk; less appealing to younger demographics.	Specifically targets university students, aligning with their mobile-first habits.
Offline Functionality	Generally requires internet access for most features.	Can function offline; allows users to log expenses without connectivity.

Compare between Mobile vs Desktop

Feature/Aspect	Desktop Application	Mobile Application
User Experience	Complex layouts for mouse and keyboard navigation.	Touch-optimized, simple, and intuitive design.
Platform Integration	Limited access to device features.	Seamless integration (camera, GPS, notifications).
Cost and Development	Higher costs and complexity in development.	More cost-effective with streamlined development.
Performance	Can be slower with larger datasets.	Quick access and responsiveness, optimized for mobile.
Security	Vulnerable to malware; less secure.	Enhanced security with biometric authentication.

The reason for choosing Mobile

- **Real-Time Tracking:** Students can log expenses instantly, promoting better financial management.
- **Convenience:** The ability to access the app anywhere enhances usability for students with busy schedules.
- **Engagement:** Mobile-specific features like notifications can help maintain user engagement and encourage regular usage.
- **Integration with Device Features:** Utilizing smartphone capabilities, such as cameras for receipt scanning and GPS for location-based expense entries, adds value to the app.

Platforms of the Software

The app should be developed for both Android and iOS platforms, which may require additional development effort and testing.

ANDROID

Development should target popular Android versions

IOS

The application must be developed for iOS devices (iPhone, iPad, etc.)

Application Implement

LANGUAGE

- Java/Kotlin: For Android development.
- Swift: For iOS development.
- React Native/Flutter: Considered for cross-platform development.

DATABASE

Firebase/SQLite: Used for storing user data and expenses.

USER INTERFACE

Responsive Design: Ensure the application works well on various screen sizes.

Essential Features of the Software

USER REGISTRATION AND AUTHENTICATION

- Account Creation: Allow users to create an account using a username and password.
- Secure Authentication: Ensure user information is protected during login.

EXPENSE MANAGEMENT

- Add, Edit, and Categorize Expenses: Enable users to add and edit expenses with categories such as rent, groceries, and transportation.
- Expense Details: Each entry should include a description, date, amount, and category.

BUDGET SETTING

- Monthly Budgets: Allow users to set budgets for different expense categories.
- Adjustable Budgets: Users should be able to modify budget amounts as needed.

EXPENSE OVERVIEW

- Monthly Summary: Provide information on total spending, remaining budget, and breakdown by category.
- Expense Trends: Users can analyze spending trends over time.

RECURRING EXPENSES

- Add Recurring Expenses: Allow users to set up recurring expenses (e.g., monthly rent).
- Automatic Addition: The app should automatically include these expenses in the monthly budget.

EXPENSE REPORTS

- Detailed Expense Reports: Enable users to generate reports for specific time periods (monthly, annually).
- Expense Notifications
- Reminders: Send notifications when users approach or exceed budget limits for specific categories.

Essential Features of the Software

USER-FRIENDLY INTERFACE

Intuitive Design: The interface should be easy to use and navigate.

DATA SECURITY

Secure Storage: User data must be encrypted and securely stored.

FEEDBACK AND SUPPORT

Feedback Form: Provide a tool for users to submit feedback or report issues.

Strategies to manage risks

Potential Risks

- Technical Risks
- Resource Risks
- Schedule Risks
- Budget Risks
- Data Security Risks
- User Acceptance Risks

Analyze potential risks

Technical Risks

New software development team members will have very little experience in mobile app development, which can lead to inefficient coding, delays in development timelines, and higher error rates. Inadequate testing due to time constraints can lead to undetected bugs or performance issues, which can lead to poor user experience, potentially damaging the app’s reputation and user retention.

Resource Humans

Ineffective communication between team members can lead to misunderstandings and inconsistent goals. Differences in perspectives or working styles can lead to conflict between team members. This can lead to duplication of effort, missed deadlines, and reduced team morale, and conflicts can create a toxic work environment that affects productivity and collaboration.

Analyze potential risks

Schedule Risks

The project has a development time of 12 weeks, which may not be enough to complete all the desired features. The development team is mainly programmers who are not experienced in mobile application development. Without enough time, some features may be omitted or developed to the quality requirements, affecting the user experience. Lack of experience may lead to extended development time due to errors in the setup process and development process.

Budget Risks

App development cost estimates can be inaccurate due to lack of information or experience. Personnel costs can increase due to the need to hire additional staff or pay higher salaries to experts. Actual costs can exceed budget, leading to feature cuts or longer completion times. Increased personnel costs can exceed the proposed budget, affecting the ability to invest in other features.

Analyze potential risks

Data Security Risks

DDoS attacks can cripple a service, making an application unavailable to users. Users cannot access the application, leading to dissatisfaction and loss of trust. Attackers can use vulnerabilities in the code to access sensitive data through SQL queries. Users' personal and financial data can be exposed or modified. Third-party APIs may not be secure, leading to data leaks. User data can be exposed through unsecured APIs. Using third-party software or libraries that may have security vulnerabilities. Attackers can exploit vulnerabilities to access sensitive data.

User Acceptance Risks

The app may have an unintuitive or difficult to use interface. Users may become frustrated and abandon the app. The app may run slowly or crash during use. Poor performance may cause users to become dissatisfied and abandon the app. Users may be concerned about their personal or financial data being leaked or being unsafe. Lack of trust may result in them not wanting to use the app, even if it has useful features. The app requests too many permissions to access the user's personal information or device. Users may refuse to install the app if they feel the permissions are unnecessary or excessive.

Risk Management Strategies

Technical Risks

Implement a mentorship program, conduct regular code reviews, and allocate time for training on best practices and tools. Develop a comprehensive testing plan that includes unit, integration, and user acceptance testing. Consider automated testing tools to streamline the process.

Resource Humans

Establish clear communication channels (e.g., Slack, regular meetings) and encourage open dialogue. Use project management tools (like Trello or Asana) to track progress and responsibilities. Foster a culture of respect and collaboration. Provide conflict resolution training and facilitate team-building activities to strengthen relationships.

Risk Management Strategies

Schedule Risks

Prioritize core features, implement Agile development to be flexible in planning and features. Provide training and support from experienced professionals, hold regular meetings and reviews to track progress.

Budget Risks

Conduct a detailed cost analysis and consult with industry experts for more accurate estimates. Plan your staffing appropriately and consider using temporary resources or freelancers when necessary.

Data Security Risks

Use DDoS protection services and have a rapid recovery plan to restore service. Use protections such as prepared statements and input validation to prevent SQL injection. Choose trusted API providers and ensure that all APIs are encrypted and secure. Perform regular security audits and update all software and libraries to the latest versions.

Risk Management Strategies

User Acceptance Risks

Conduct UX/UI testing with real users during the development phase to ensure the interface is easy to use and user-friendly. Optimize the source code and perform regular performance testing to ensure the application runs smoothly. Provide clear and transparent information about how data is protected, including security measures and privacy policies. Request only necessary access and explain the reason for each access so that users understand.


¡MUCHAS
GRACIAS!
THANK
YOU