



GROUP 3

# THE CAMPUS EXPENSE MANAGER APPLICATION

# MEMBERS

**01**

NGUYEN MINH HIEU

**02**

NGUYEN TIEN MANH

**03**

BUI HOANG LONG

**04**

PHAM XUAN BACH

# CONTENT

**01**

STAKEHOLDERS

**02**

FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

**03**

DIFFICULTIES OF THE PROJECT

**04**

PLATFORM USED AND FEATURES

**05**

ANALYZE HIDDEN RISKS

# INTRODUCTION

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 specifically for university students.

The CampusExpense Manager application aims to support college students in effective financial management. By simplifying the spending tracking and budgeting process, the app hopes to help students make informed financial decisions and maintain financial stability throughout their studies.

# STAKEHOLDERS



Students



Budgetwise Solution team



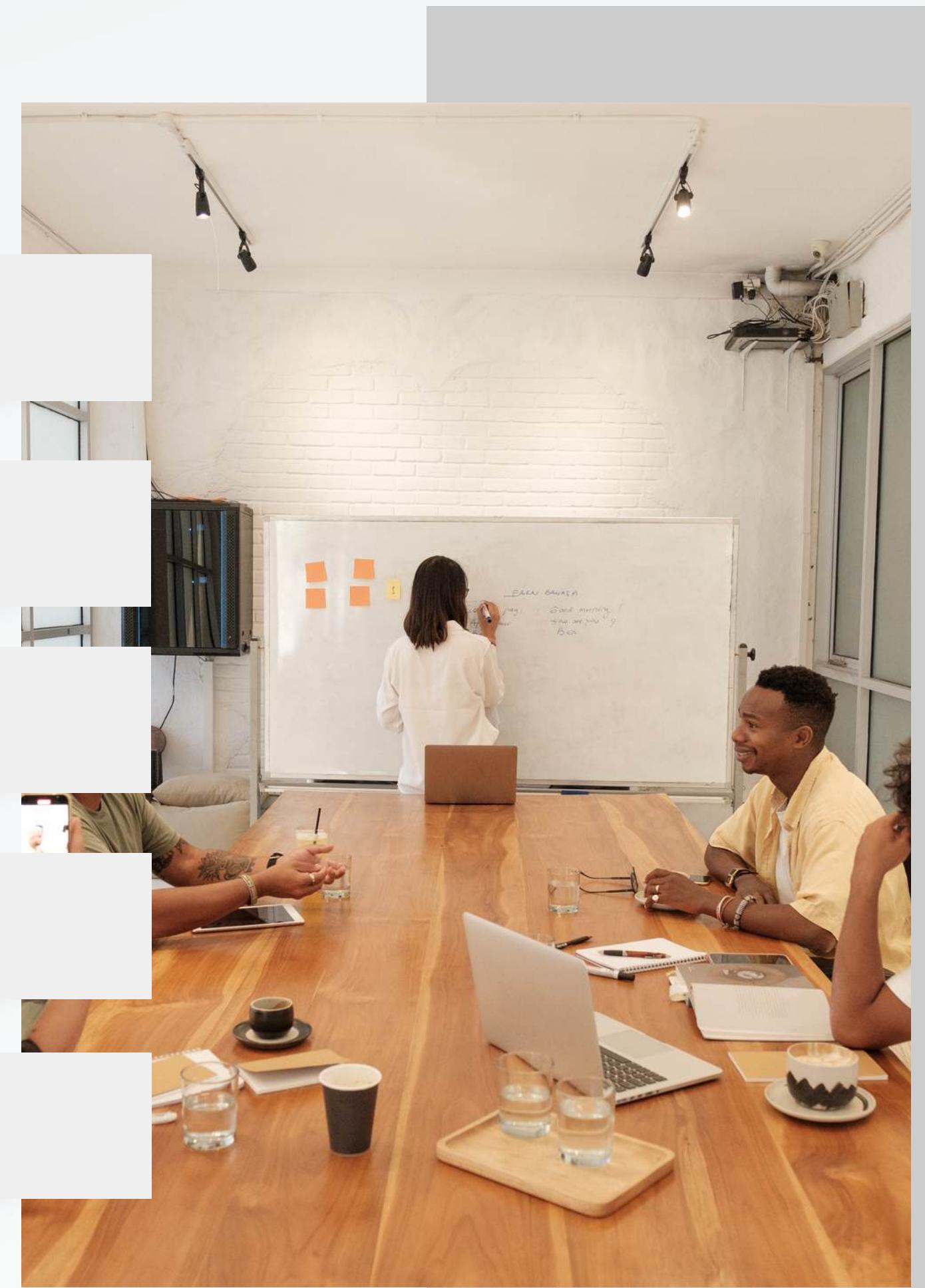
University administration



Investor



Students



# Students

Role: Students will be the main users of the application, using it to track and manage their daily expenses throughout their studies. They will interact directly with the app's features to ensure that it meets their needs effectively.

Goals and needs: The main goal of students is to manage spending appropriately so as not to fall into debt during their studies. They need a tool that helps them budget, track spending, and make smart financial decisions to maintain financial stability.

Contributions: Students can provide feedback and ideas during the app testing process, helping the development team better understand what they need and want. Their participation in testing and providing feedback will be invaluable in improving the quality of the product.

Challenges: Some students may have difficulty accessing technology or lack time to use the application due to their busy study schedule. Additionally, they may feel hesitant to change their financial management habits.



# Budgetwise Solution team

Role: The BudgetWise Solutions team consists of approximately 5-6 members, each taking on a specific role in the application development process. Members may include programmers, user interface designers, testers, and project managers, all working toward the common goal of creating a high-quality product for the customer. student.

Goals and needs: The team's goal is to develop an effective expense management application that meets students' needs in tracking and managing personal finances. They need to have a clear understanding of the end user's needs and challenges to design relevant and useful features in the application.

Contribution: Each team member will contribute their own skills and expertise to the project. Programmers will develop the source code, designers will create attractive user interfaces, and testing experts will ensure that the application works smoothly and without errors before launch.

Challenges: The team may face challenges such as time and budget constraints, as well as inexperience in mobile application development. They also need to manage effective communication between members to ensure that everyone is on track and in sync with the development process.



# Divide the work of the BudgetWise team

Team Member	Role	Tasks	Deadline
Hieu	Frontend Developer	- Design user interface (UI) - Implement expense tracking features - Develop budget setting UI	Week 6
Manh	Backend Developer	- Set up database schema - Implement user authentication - Develop API for expense management	Week 5
Long	QA Tester	- Create test cases for all features - Conduct usability testing - Report and track bugs	Week 8
Bach	Database Designer	- Design and optimize database schema - Ensure data integrity and security - Collaborate with backend developer for database integration	Week 4

# University administration

**Role:** The Board of Directors, including the principal and vice principals, is responsible for managing and operating the university's activities. They will serve as key stakeholders in approving and supporting the "CampusExpense Manager" application development project, ensuring that the application is aligned with the school's educational and financial goals.

**Goals and needs:** The main goal of the Board of Directors is to enhance the learning experience for students and support them in personal financial management. They needed an app that would help students track spending and budget, thereby easing financial pressure and facilitating more effective learning.

**Contribution:** The School Board can provide financial resources and policy support for the project. They can also connect the development team with experts in the fields of finance and education to ensure that the application properly meets the needs of students.

**Challenges:** One of the major challenges the Board of Directors may face is balancing a limited budget with the growing needs of students. They also need to ensure that the application complies with applicable regulations and educational standards.



# Investor

Role: Mr. Long Ma is a strategic investor who will provide the necessary capital for the application development project "CampusExpense Manager". With extensive experience in the field of technology investment, he will play an important role in directing and financially supporting the development team.

Objectives and needs: Mr. Long's goal is to invest in projects with high growth potential, especially applications serving the needs of students. He looks for investment opportunities that can bring sustainable profits and contribute positively to the education community.

Investment: Mr. Long decided to invest 250 million VND in the project. This investment will be used for application development, marketing, and other support activities to ensure the success of the project.

Contribution: In addition to providing capital, Mr. Long can also share his experience and knowledge in the fields of startups and product development. He can connect the development team with potential partners and markets, helping to expand the project scale.

Challenges: One of the challenges that Mr. Long may face is accurately assessing the application's potential in a competitive market. He needs to consider factors such as market trends, user needs and application competitiveness.



Mr. Long Ma

# Estimated budget for the CampusExpense Manager Application project

Item	Amount (VND)	Notes
Application Development	120,000,000	Costs for the software development team, UI/UX design.
Marketing and Advertising	70,000,000	Costs for promotional campaigns, social media, and launch events.
Operational Costs	30,000,000	Costs for server maintenance, hosting services, and technical support.
Market Research	15,000,000	Costs for user surveys, market analysis, and competitor studies.
Administrative Expenses	10,000,000	Costs for office supplies, phone, internet, and other administrative expenses.
Contingency	5,000,000	Reserve for unforeseen expenses.
<b>Total</b>	<b>250,000,000</b>	Total proposed budget for the entire project.

# **Estimated budget for the CampusExpense Manager Application project**

## **Detailed Explanation**

**Application Development:** This is the largest expense, covering salaries for developers, designers, and IT specialists. This amount ensures that the application is developed with high quality and meets user needs.

**Marketing and Advertising:** To attract users, an effective marketing campaign is essential. This budget will be used to promote the application on social media platforms and organize events to introduce the app to students.

**Operational Costs:** After the application is launched, there will be ongoing costs to ensure that it runs smoothly and provides technical support to users.

**Market Research:** Conducting surveys and research is crucial to understand user needs and preferences.

**Administrative Expenses:** This includes daily expenses to maintain the operations of the development team and office.

**Contingency:** It is always necessary to have a reserve for unexpected situations that may arise during the development and deployment of the application.

# Functional Requirements

## User Registration & Authentication

Secure account creation and login to access personal expense data.

## Expense Tracking

Add, edit, and categorize expenses with details (description, date, amount, category).

## Budget Setting

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

## Expense Overview

Summarize monthly expenses, remaining budget, and category breakdowns.  
Display expense trends over time.

# Functional Requirements

## Recurring Expenses

Add recurring expenses with start/end dates, automatically included in monthly budget.

## Expense Reports

Generate detailed reports by time period (e.g., monthly, annually) with category breakdown.

## Expense Notifications

Send reminders when approaching or exceeding budget limits.

# Non-Functional Requirements

## Performance

Smooth, responsive user experience, even with extensive data.

## User-Friendly Interface

Intuitive design with clear labels and easy navigation.

## Platform Compatibility

Available on both Android and iOS.

# Non-Functional Requirements

## Data Security

Secure storage with encryption, adherence to data privacy standards.

## Feedback and Support

In-app feedback form; team actively monitors and addresses user feedback.

## Monetization (Optional)

Future updates may include in-app ads or premium features; prioritize core functionality first.

# Project difficulties how to overcome expense management system

## Limited Experience

- Description: The development team has limited experience in mobile app development, which can lead to challenges in understanding best practices and optimizing the development process.
- Impact: This may result in longer development times, increased errors, and potential rework.

## Scope Creep

- Description: As stakeholders provide feedback, there may be a tendency to add new features or change requirements.
- Impact: This can extend the timeline and strain resources, making it difficult to meet the original project objectives.

## Tight Timeline

- Description: The project has a strict 12-week timeline from concept to launch.
- Impact: This may lead to rushed development, inadequate testing, and potential compromises on quality if not managed effectively.

## Technical Challenges

- Description: Integrating various technologies and ensuring compatibility between iOS and Android platforms can be complex.
- Impact: Technical issues may arise, leading to delays and requiring additional resources for troubleshooting.

# Difficulties of the project

## Data Security and Privacy

- Description: Ensuring the security of user data and compliance with data protection regulations (e.g., GDPR) is critical.
- Impact: Implementing adequate security measures may require additional time and expertise, potentially impacting the project timeline.

## Offline Functionality

- Description: Ensuring the app functions effectively without an internet connection adds complexity to the design and development.
- Impact: This may require additional development effort and testing to ensure data synchronization when connectivity is restored.

## Resource Constraints

- Description: The project operates with a limited budget and a small team, which can restrict the scope of development and marketing efforts.
- Impact: This can lead to compromises in features, marketing strategies, and overall project quality.

## Feedback Management

- Description: Collecting and implementing user feedback effectively is essential for improving the app.
- Impact: Poorly managed feedback processes can result in misaligned features and user dissatisfaction.

# strategy to overcome difficulties

## Limited Experience

- Training and Workshops: Invest in training sessions and workshops on mobile development best practices and tools.
- Mentorship: Pair less experienced developers with seasoned mentors who can provide guidance and support.
- Incremental Learning: Start with smaller, less complex features to allow the team to build confidence and skills progressively.

## Tight Timeline

- Agile Methodology: Adopt an Agile approach to break the project into manageable sprints, allowing for regular assessments and adjustments.
- Prioritization: Focus on core functionalities first, deferring non-essential features to later phases or updates.
- Regular Milestones: Set clear milestones and deadlines to keep the team accountable and ensure steady progress.

## Scope Creep

- Change Control Process: Establish a formal process for evaluating and approving changes to the project scope, including impact assessments.
- Clear Requirements Documentation: Ensure that project requirements are well-documented and agreed upon at the start to minimize misunderstandings.
- Stakeholder Engagement: Regularly communicate with stakeholders to manage expectations and clarify priorities.

# Strategy to overcome difficulties

## Technical Challenges

- Technology Research: Conduct thorough research on the best tools and frameworks for cross-platform development (e.g., React Native, Flutter).
- Prototyping: Create prototypes to test integrations early in the development process and identify potential issues.
- Continuous Testing: Implement automated testing to catch technical issues early and ensure consistent performance across platforms.

## Data Security and Privacy

- Security Best Practices: Follow industry best practices for data encryption, secure authentication, and data storage.
- Regular Audits: Conduct regular security audits and vulnerability assessments to identify and address potential risks.
- Compliance Training: Provide training on data protection regulations to ensure the team understands compliance requirements.

## Resource Constraints

- Lean Development: Adopt lean principles to maximize efficiency, focusing on delivering value with minimal waste.
- Outsource Where Necessary: Consider outsourcing specific tasks (e.g., UI/UX design) to experts to improve quality without overextending the team.
- Prioritize Features: Identify the most critical features that deliver the highest value to users and prioritize those within the budget.
-

# Strategy to overcome difficulties

## Feedback Management

- Feedback Mechanism: Implement an in-app feedback feature to gather user insights easily.
- Regular Reviews: Schedule regular review sessions to analyze feedback and determine actionable improvements.
- Iterative Development: Use feedback to inform iterative updates, ensuring continuous improvement of the application.

## Offline Functionality

- Local Storage Solutions: Use local databases (e.g., SQLite) to store data on the device, enabling offline access and functionality.
- Data Synchronization: Implement data synchronization processes to update the server when the device is back online.
- User Testing: Conduct thorough testing in offline scenarios to ensure seamless user experience.

# Platform Used

Android: The app will be developed for Android devices to reach a large user base, as many students use Android smartphones.

iOS: Developing for iOS ensures accessibility for users who prefer Apple devices, allowing the app to cater to a wider audience.

# Comparison table of iOS and Android

Feature/Aspect	iOS	Android
Market Share	Approximately 27% of global smartphone market	Approximately 73% of global smartphone market
Development Language	Swift, Objective-C	Kotlin, Java
Development Environment	Xcode	Android Studio
App Store Approval	Strict review process, typically faster	More lenient review process, can take longer
User Demographics	Higher income users, often more loyal	Diverse user base, including budget-conscious consumers
Monetization	Higher average revenue per user (ARPU)	More ad-driven revenue opportunities
Fragmentation	Limited fragmentation (fewer device models)	High fragmentation (many devices and OS versions)

# Comparison table of ios and android

Customization	Limited customization options	Highly customizable user interface
Updates	Regular and timely updates from Apple	Updates depend on device manufacturers
Hardware Integration	Strong integration with Apple hardware	Wide range of hardware options and manufacturers
User Experience	Consistent user experience across devices	Varied <u>user</u> experience due to device differences
Testing	Fewer devices to test on, easier to manage	More devices and configurations increase testing complexity
Development Cost	Generally higher due to stricter guidelines	Typically lower, but varies with device diversity
Community Support	Strong community support and resources	Large community with extensive resources

# Required Features

## 1. User Registration and Login

Enables users to create accounts with usernames and passwords.

Ensures information security and allows logins to access spending data.

## 2. Spending Tracking

Allows users to add, edit, and categorize spending items (e.g., rent, groceries).

Each transaction should have a description, date, amount, and category for easy management.

# Required Features

## 3. Set Up a Budget

Users can set up monthly budgets for each spending category and change them as needed.

## 4. Spending Overview

Displays total monthly spending, budget balance, and breakdown by category.

Enables review of spending trends over time.

# Required Features

## 5. Recurring Spending

Enables setting up recurring expenses (e.g., monthly rent) with start and end dates.

Automatically updates recurring expenses to monthly budgets.

## 6. Spending Reports

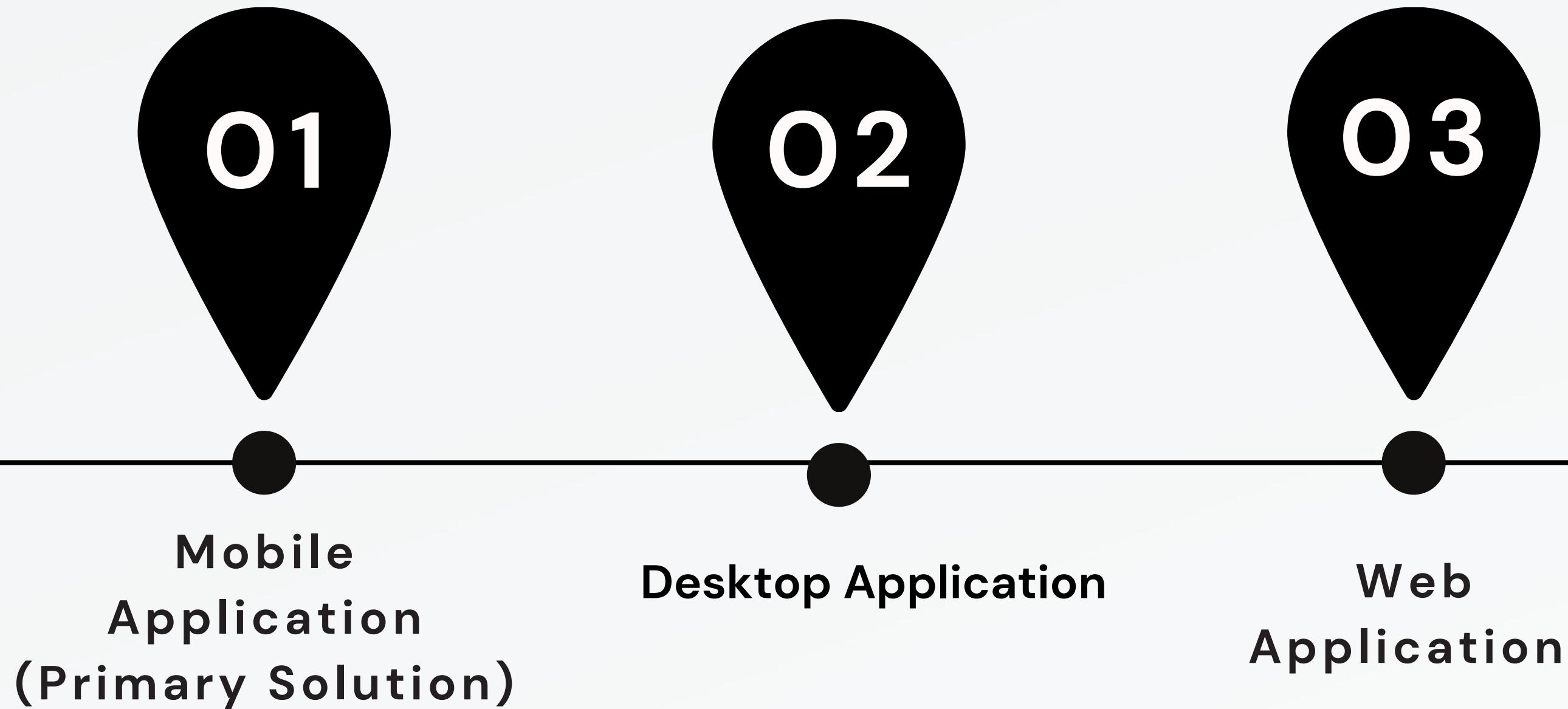
Generate detailed reports over time periods (e.g. monthly, yearly), analyzing spending by category.

# Required Features

## 7. Budget Notifications

Send notifications when users are close to or over their budget for a category, helping to control spending.

# Business application solutions



## Mobile Application (Primary Solution)

Platform: Android

Description: A mobile application is the most suitable solution for students who are constantly on the go. It allows users to track expenses in real-time, set budgets, and receive notifications directly on their smartphones.

Tool : android studio

### Key Features:

- User-friendly interface with intuitive navigation.
- Secure user registration and authentication.
- Expense tracking with categorization.
- Budget setting and adjustment functionalities.
- Recurring expense management.
- Expense reports generation.
- Notifications for budget limits.

### Benefits:

- Accessibility and convenience for users.
- Real-time updates and notifications.
- Offline functionality to cater to students with limited connectivity.

## Desktop Application (Optional Solution)

### Features:

Local Storage: Allow users to store data locally with an option to sync with the cloud.

Comprehensive Reporting: Provide advanced reporting features for detailed financial analysis.

Integration with Other Tools: Allow integration with spreadsheet software for exporting data.

### Advantages:

Performance: Desktop applications can handle larger datasets more efficiently.

Offline Capability: Users can manage expenses without needing an internet connection.

## Web Application (Supplementary Solution)

- Platform: A responsive web application that complements the mobile app.
- Features:

Similar functionality to the mobile app, allowing users to manage expenses from a desktop.

Data synchronization between mobile and web platforms.

- Benefits:
- Broader accessibility for users who prefer desktop usage.  
Enhanced data visualization capabilities on larger screens.

# Comparison between mobile and desktop applications :

Feature/Aspect	Mobile Application	Desktop Application
<b>Accessibility</b>	Accessible anytime, anywhere via smartphones and tablets.	Requires a computer, limiting access to specific locations.
<b>User Interface</b>	Designed for touch interactions; <u>typically</u> simpler and more intuitive.	Can accommodate more complex interfaces with detailed features.
<b>Screen Size</b>	Smaller screens necessitate concise information and larger touch targets.	Larger screens allow for more information and detailed views.
<b>Performance</b>	May be limited by device capabilities, but optimized for speed and responsiveness.	<u>Generally</u> more powerful, capable of handling larger datasets and complex calculations.

Feature/Aspect	Mobile Application	Desktop Application
<b>Offline Capability</b>	Can store data locally for offline use, ideal for students with limited connectivity.	Can also function offline, but typically requires an initial internet connection for setup.
<b>Notifications</b>	Supports push notifications for reminders and alerts, enhancing user engagement.	Limited notification capabilities; relies on system alerts or email notifications.
<b>Development Cost</b>	<u>Usually</u> lower initial development costs due to cross-platform frameworks (e.g., Flutter, React Native).	Higher costs due to the need for separate development for different operating systems (Windows, macOS, etc.).
<b>User Demographics</b>	Appeals more to younger users and students who prefer mobile devices.	Attracts users who prefer working on larger screens or need advanced features.
<b>Integration</b>	Can easily integrate with mobile device features (camera, GPS, etc.).	Can integrate with other desktop applications and tools (spreadsheets, accounting software).

<b>Data Security</b>	Requires robust security measures to protect user data on mobile devices.	Can implement strong security protocols, but may require more extensive setup.
<b>User Engagement</b>	Higher engagement rates due to constant accessibility and mobile usage habits.	Engagement may be lower as users may not use the app as frequently.
<b>Updates</b>	Regular updates can be pushed through app stores, ensuring users have the latest version.	Users may need to manually download and install updates, leading to version fragmentation.
<b>Feedback Mechanism</b>	Easier to implement in-app feedback forms and surveys for real-time user input.	Feedback mechanisms can be included but may not be as immediate or accessible.

# Potential risks in project implementation and how to manage risks

## Platform compatibility risks

Description: The project requires the app to work on both Android and iOS platforms, which requires a lot of testing and optimization time, which can lead to compatibility issues.

How to manage:

- Develop for each platform or use a cross-platform framework: Consider cross-platform frameworks like React Native to reduce development time.
- Cross-device testing: Use real devices or emulated platforms to test compatibility regularly throughout the development process.
- Minimize platform-dependent features: Limit features that may be platform-dependent to simplify testing and development.

## Risks regarding financial resources and additional costs

Description: With a limited budget, any unexpected costs can affect project progress and quality.

How to manage:

- Estimate additional costs: Allocate a reserve budget for unplanned costs.
- Choose cost-effective tools and services: Consider free or low-cost tools and services that fit your project's needs.
- Closely monitor costs: Evaluate actual costs against expectations regularly to make timely adjustments.

## Data security and privacy compliance risks

Description: The application will store users' financial data, so it needs to comply with data security regulations. Security flaws can cause loss of trust from users and affect company reputation.

How to manage:

- Data encryption: Use encryption measures for sensitive data both in transmission and in storage.
- Regular security testing: Perform regular security checks to ensure there are no vulnerabilities.
- Security training: Train the team on security standards and user data protection regulations.

## Risks regarding project completion time

Description: The project had a deadline of only 12 weeks from start to launch, a fairly short period of time for an inexperienced team.

How to manage:

- Detailed planning: Make detailed plans for each stage and closely monitor progress.
- Using Agile development method: Applying Agile development method helps the team work effectively, receive feedback and continuously adjust the product in each stage.
- Prioritize key functions: Focus on completing key functions (like tracking spending and setting budgets) before developing additional features.

# THANK YOU



GROUP 3