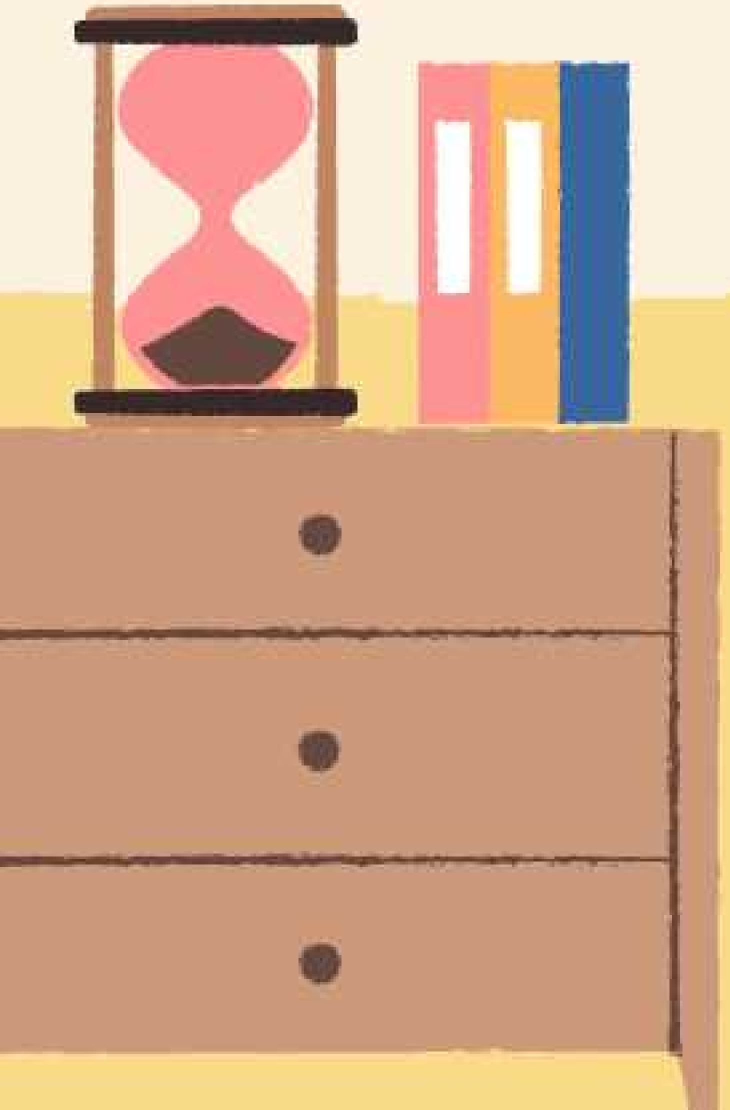
My Project

CampusExpense Manager Project

Presenter's by Group 5

Table Content



Project Introduction

The CampusExpense Manager project is a mobile application developed by BudgetWise Solutions to help university students manage their personal finances efficiently and easily. The app allows students to track expenses, set budgets, receive notifications when approaching budget limits, and view monthly or annual expense reports. With a user-friendly interface, high security standards, and compatibility on both Android and iOS, CampusExpense Manager is designed to meet the specific financial needs of students living on or off-campus, empowering them to build strong financial management skills during their academic journey.

Project Objectives

Main goal: Create an easy-to-use app that helps students manage their spending and maintain their budget.

Functional Requirement

- User Registration and Authentication: Users can create accounts using a username and password. Authentication must be secure and allow users to log in and access their expense data.

- Expense Tracking: Users can add, edit, and categorize expenses (e.g. rent, groceries, transportation) where each expense item must include a description, date, amount, and category.

- Set Budgets: Users can set monthly budgets for various expense categories (e.g. food, entertainment, education), and the app must allow users to adjust the budget amount as needed.

- Expense Overview: The app must provide a summary of monthly expenses, including total expenses, remaining budget, and a breakdown by category.

Functional Requirement

- Recurring expenses: Users can add recurring expenses (e.g. monthly rent) with start and end dates then the app should automatically add these expenses to the user's monthly budget.
- Expense Reports: Users can generate detailed expense reports for specific time periods (e.g. monthly, yearly) and the reports must include a breakdown of expenses by category.
- Expense Notifications: The app must send reminders or notifications when users reach or exceed their budget limits for specific categories.

Non-Functional Requirements:

- Performance: The application must provide a smooth and responsive user experience, even with large amounts of data overhead.
- User-friendly interface: The user interface should be intuitive, clearly labeled, and easy to navigate for easy expense tracking.
- Platform compatibility: The app should be developed for both Android and iOS platforms to reach a wider audience.
- Data security: User data, including expense and budget information, should be stored securely and protected with encryption. Data security measures should be followed to ensure user information is kept secure.
- Feedback and Support: Include an in-app feedback form for users to report issues or make suggestions. BudgetWise Solutions should proactively monitor user feedback and resolve issues promptly.
- Monetization (Optional): Monetization features, such as in-app advertising or premium features, may be explored in future updates. Initial development should prioritize core cost management functionality.

Non-Functional Requirements:

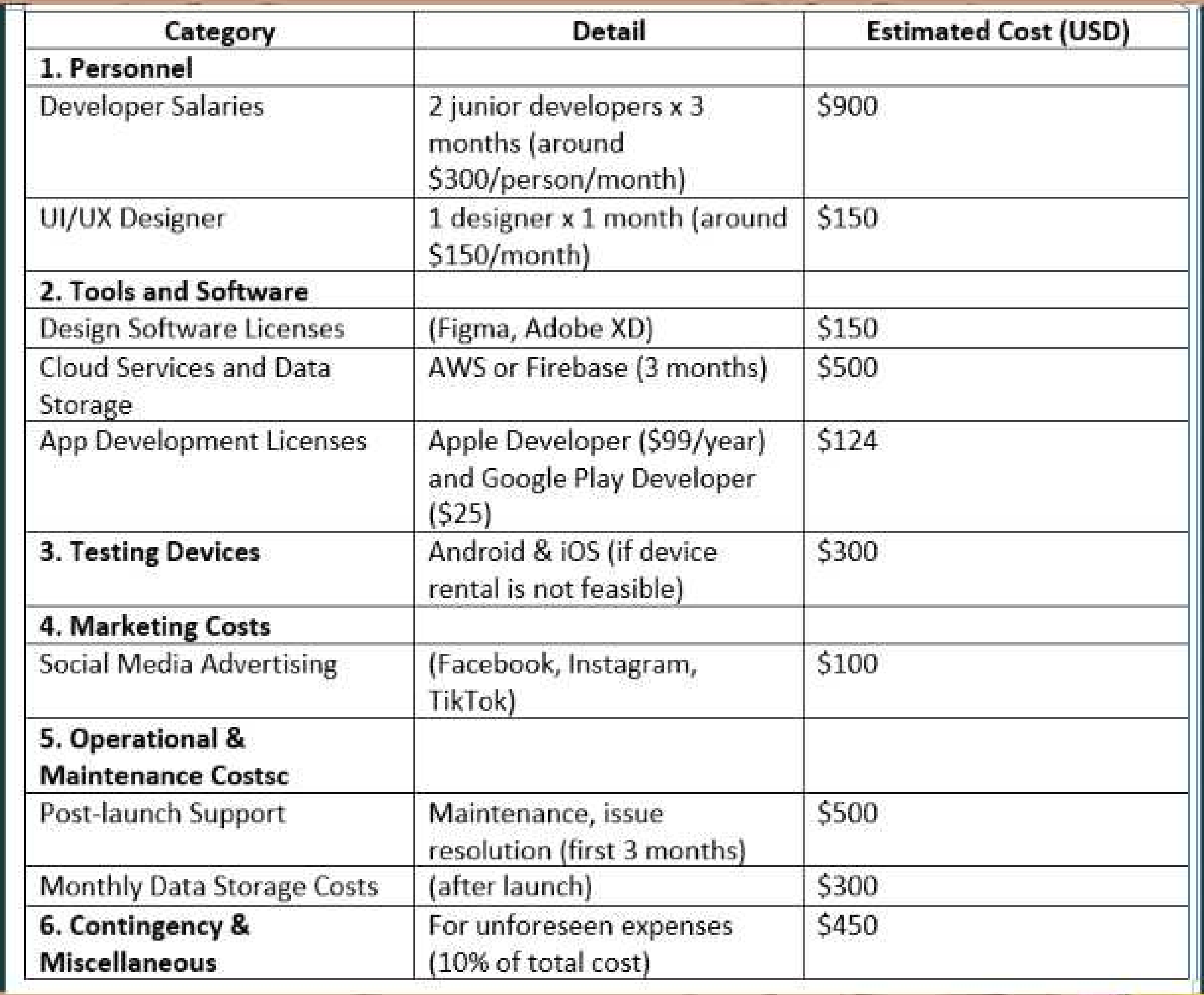
- Performance: The application must provide a smooth and responsive user experience, even with large amounts of data overhead.
- User-friendly interface: The user interface should be intuitive, clearly labeled, and easy to navigate for easy expense tracking.
- Platform compatibility: The app should be developed for both Android and iOS platforms to reach a wider audience.
- Data security: User data, including expense and budget information, should be stored securely and protected with encryption. Data security measures should be followed to ensure user information is kept secure.
- Feedback and Support: Include an in-app feedback form for users to report issues or make suggestions. BudgetWise Solutions should proactively monitor user feedback and resolve issues promptly.
- Monetization (Optional): Monetization features, such as in-app advertising or premium features, may be explored in future updates. Initial development should prioritize core cost management functionality.

Stakeholders and Their Expectations

In our Student Expense Management application development project: Campus Expense Manager, the stakeholders and their expectations are:

- Investor Hoang Nam Tien: Interested in the budget, profit of the application and growth potential.
- Users Nguyen Anh Tu, Tran Thanh Tung...(students): Want a reliable and easy-to-use cost management tool.
- Project manager Nguyen Van Dai & Btec Dev development team: Complete the project on time, on budget and according to standards.
- FPT University administrator: Encourage students to use the application to practice financial management.

Estimated Budget



Category	Detail	Estimated Cost (USD)
1. Personnel		
Developer Salaries	2 junior developers x 3 months (around $300/person/month)	$900
UI/UX Designer	1 designer x 1 month (around $150/month)	$150
2. Tools and Software		
Design Software Licenses	(Figma, Adobe XD)	$150
Cloud Services and Data Storage	AWS or Firebase (3 months)	$500
App Development Licenses	Apple Developer ($99/year) and Google Play Developer ($25)	$124
3. Testing Devices	Android & iOS (if device rental is not feasible)	$300
4. Marketing Costs		
Social Media Advertising	(Facebook, Instagram, TikTok)	$100
5. Operational & Maintenance Costsc		
Post-launch Support	Maintenance, issue resolution (first 3 months)	$500
Monthly Data Storage Costs	(after launch)	$300
6. Contingency & Miscellaneous	For unforeseen expenses (10% of total cost)	$450

Estimated Budget

Estimated Total Budget: 3474$

Notes:

- Surcharges and Contingencies: Contingency funds are in place to cover unforeseen costs that may arise during development.
- People and Tools: The team will primarily consist of entry-level developers, so additional training may be required.
- Testing and Release: The software will be tested on real devices to ensure cross-platform compatibility.
- Marketing and Early Adopters: Focus on social media, promoting directly to college students.

Challenges for BudgetWise Solutions

- Limited skills of the development team: being a small development team with limited experience in application development, lack of skills can be a challenge for BudgetWise Solutions.
- Ensuring cross-platform compatibility (Android and iOS): Since the team has no experience in mobile application development, compatibility with other platforms is also considered a challenge for BudgetWise Solutions.
- Limited budget and team, pressure on completion time with allowed finances: being a small development team with limited team and experience, pressure on completion time with limited budget can be considered a challenge for BudgetWise Solutions

Challenges for BudgetWise Solutions

- Data Privacy Compliance: Security and compliance to ensure privacy of user data for a team with no experience in application development programming can be a challenge for BudgetWise Solutions.

- Offline Capability: To serve some students in areas with limited connectivity, it is necessary for the application to be able to function effectively without an Internet connection and this can be a challenge for BudgetWise Solutions.

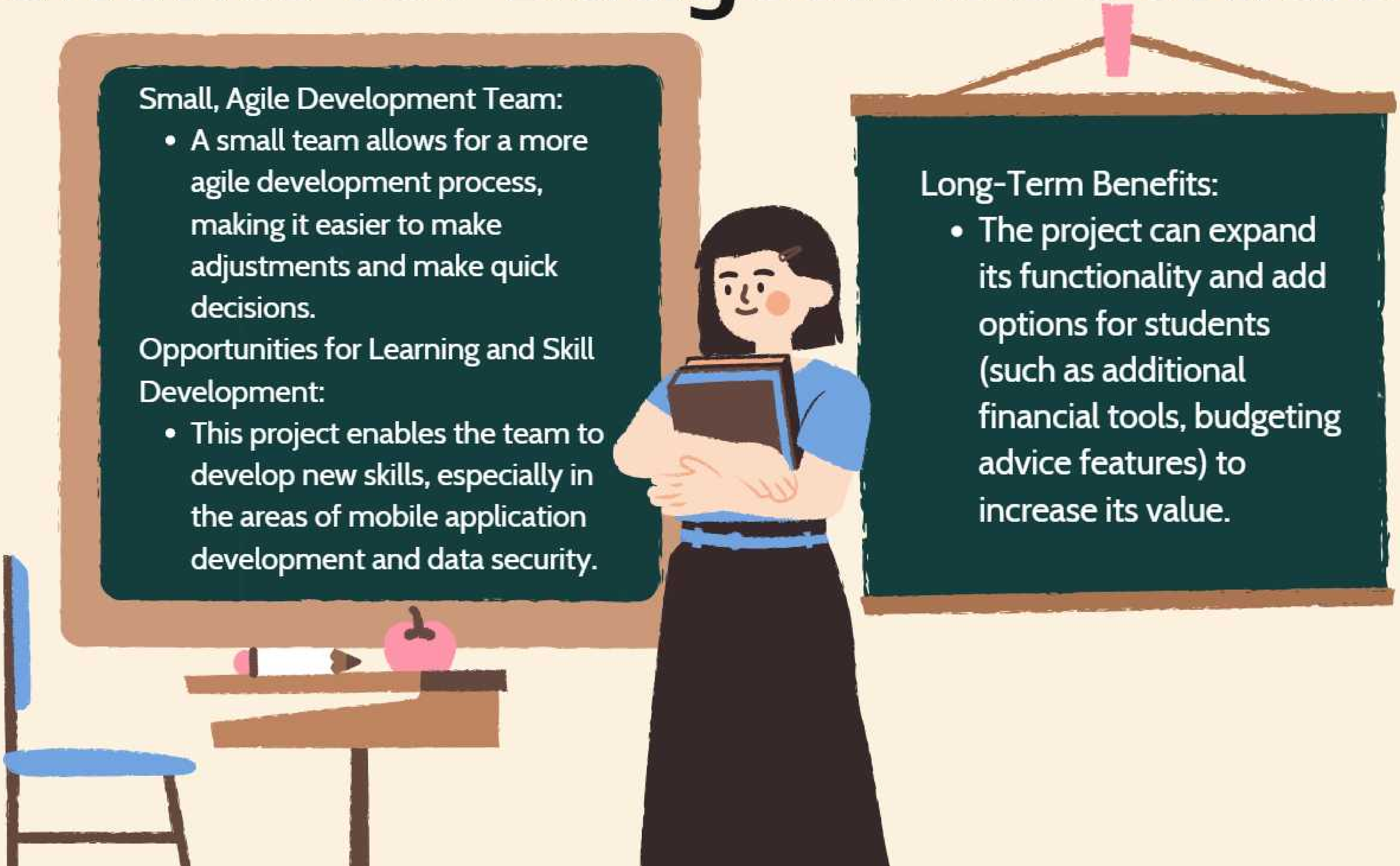
Benefits for BudgetWise Solutions

Potential Market:

- Targeting university students - a group of users with high demand for personal finance management and spending control.
- The project is assessed to be able to develop into a tool to support students in managing costs effectively.

Investor Support (School Management Board): The school management board (or school association) is the project investor, providing financial support and potential for user expansion through the school system.

Benefits for BudgetWise Solutions



Small, Agile Development Team:

- A small team allows for a more agile development process, making it easier to make adjustments and make quick decisions.

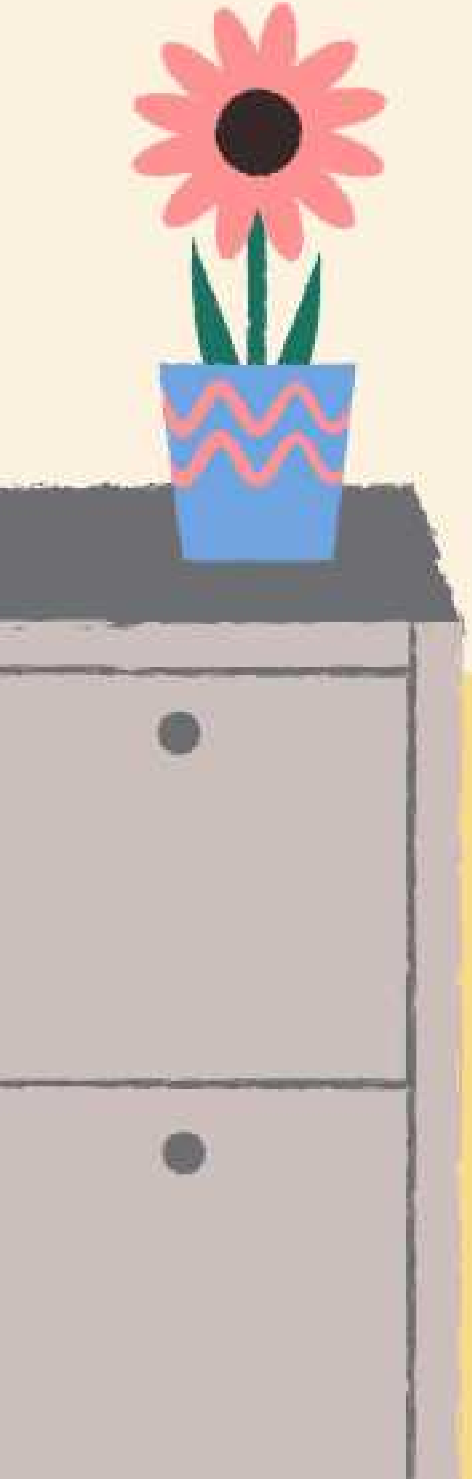
Opportunities for Learning and Skill Development:

- This project enables the team to develop new skills, especially in the areas of mobile application development and data security.

Long-Term Benefits:

- The project can expand its functionality and add options for students (such as additional financial tools, budgeting advice features) to increase its value.

Development Platform & Tools Used



Benefits of Mobile Apps

- Convenient & Flexible: Users can access it anytime, anywhere, enabling students to track expenses and manage budgets in real time.
- Integration with Mobile Features: It can leverage phone features like push notifications, GPS, camera, etc.
- Offline Capability: The app can function offline (as required in the project), allowing students to manage expenses even without an internet connection.
- High Accessibility: Students are more likely to have mobile phones than computers, making the app easily accessible to the target audience.

Development Platform & Tools Used

Limitations of Mobile Apps

- Limited Screen Space: The small screen may restrict the ability to display detailed financial information or reports.
- Optimization for Cross-Platform Use: To run smoothly on both Android and iOS, the app needs optimization, which may increase development time and costs.
- Storage and Performance Constraints: Mobile apps generally have limited system resources compared to desktop applications.

Development Platform & Tools Used

Benefits of Desktop Apps

- Larger and More Powerful Interface: The larger screen space allows for displaying detailed information, such as expense reports or financial charts.
- Stronger Storage and Processing Capabilities: Desktops usually have more processing power and memory, allowing for faster data processing.
- Integration with Other Systems: Desktop apps can be more easily integrated with the school's financial management systems or larger databases.

Development Platform & Tools Used

Limitations of Desktop Apps

- Limited Screen Space: The small screen may restrict the ability to display detailed financial information or reports.
- Optimization for Cross-Platform Use: To run smoothly on both Android and iOS, the app needs optimization, which may increase development time and costs.
- Storage and Performance Constraints: Mobile apps generally have limited system resources compared to desktop applications.

Development Platform & Tools Used

The Right Choice for CampusExpense Manager Projects

A Mobile App is the ideal choice for the CampusExpense Manager project as it aligns well with students' needs for convenience and accessibility, allowing them to track and manage expenses anytime, anywhere. The mobile platform will enable students to manage their finances on the go, even offline, which is especially beneficial for those with limited internet access. Additionally, integrating mobile features such as push notifications will enhance the user experience by providing timely budget alerts and reminders. Overall, a Mobile App offers the best fit for achieving the project's goals of effective financial management for students.

Development Platform & Tools Used

Tools I Used

With the choice of Mobile App programming, I will use Android Studio and Java language to develop the application. Using Android Studio and Java will provide a powerful environment to build the CampusExpense Manager application, with good support for Android specific features and many rich libraries to handle requirements such as user authentication, data storage and offline capabilities.

Risk Analysis

Risk of Development Timeline

With only 12 weeks to finish the app and the development team's limited experience, there could be delays or features that aren't fully developed.

Budget Risk

The project's small budget might not be enough to finish it, particularly if unforeseen problems crop up or if the staff needs more training.

App Performance Risk

When using low-end devices or managing high volumes of data, the application may experience performance problems.

Risk Analysis

Security and Data Privacy Compliance Risk

Data loss or legal infractions may result from the app's failure to secure user information, which is required by data privacy legislation.

Compatibility and Offline Functionality Risk

It can be difficult to guarantee compatibility between the iOS and Android operating systems, and there may be technological limitations when putting offline functions into practice.

User Experience (UX) Risk

Poor user adoption could result from an interface that is difficult to use or non-intuitive.

Risk Management

Risk of Development Timeline

- Planning in Detail: Establish clear benchmarks for every stage (analysis, design, development, testing, and deployment).
- Utilize the Agile Methodology: Divide the project into sprints so that you can evaluate its progress and make necessary modifications.
- Prioritize Core Features: If time allows, prepare for extra features after concentrating on the most important ones.

Budget Risk

- Effective Resource Management: Keep an eye on spending and make the most of resources by utilizing open-source tools and libraries.
- Budget Contingency: Reserve a tiny percentage of the spending plan for unforeseen expenses.
- Targeted Training: To enhance the team's abilities, conduct brief but efficient training.

App Performance Risk

- Performance Testing: To make sure the app functions properly, do performance tests on a regular basis.
- Code Optimization: Make use of optimization strategies such as effective resource management, asynchronous data loading, and memory reduction.
- Data Loading Limits: Limit how much data can be loaded or processed at once to keep the program from becoming sluggish.

Risk Management

Security and Data Privacy Compliance Risk

- Data encryption: To safeguard private information, use robust encryption techniques.
- Security Testing: To find and address vulnerabilities, do routine security checks.
- Privacy Policy: Create a transparent privacy policy and make sure it complies with the most recent data protection laws.

Compatibility and Offline Functionality Risk

- Cross-Platform Development: Make use of resources that facilitate cross-platform development, such as frameworks designed for both iOS and Android or Flutter.
- Cross-Device Testing: Verify the app's reliability across a range of devices and operating systems.
- Local Data Storage: To enable offline functionality, use local data storage programs (such as SQLite).

User Experience (UX) Risk

- User research: To learn about students' wants and preferences, get their feedback.
- Use contemporary, approachable UX/UI design ideas to create intuitive user interfaces.
- User Testing: To get input and make changes, run user tests with a small number of students.