

Application Development

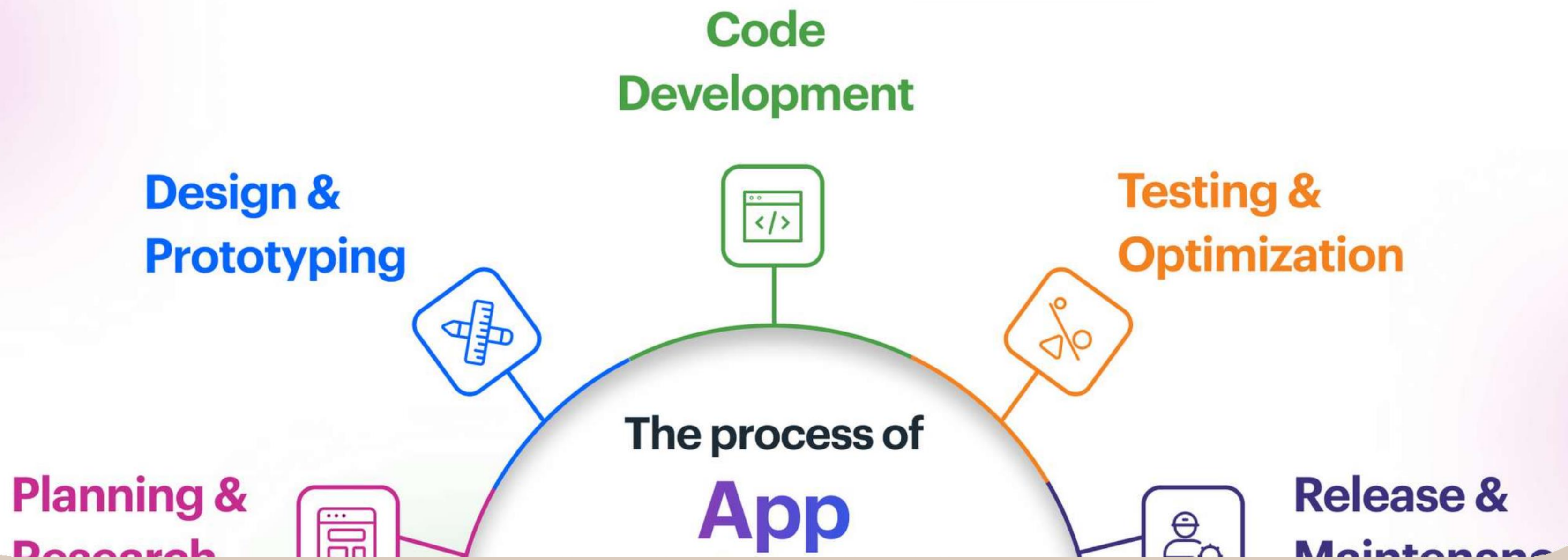





Table of Content * * *

- Overview
 - Stakeholders
 - The difficulties that BudgetWise Solutions might encounter
 - The issues that need to be addressed
 - Business application solutions
 - Areas of risk pertaining to this software project and explore strategies for managing these risks
- 

Project Overview

Develop an intuitive mobile application, CampusExpense Manager, to help university students manage their expenses effectively and stay within their budgets.

Functional Requirement

- User Registration and Authentication
- Expense Tracking
- Budget Setting.
- Expense Overview.
- Recurring Expenses.
- Expense Reports
- Expense Notifications

Non-Functional Requirement

- Performance.
- User-Friendly Interface.
- List an accomplished objective.
- Platform Compatibility.
- Data Security
- Monetization (Optional)

Stakeholders

- Student
- BudgetWise solution team
- University management

Requirement

- User registration and authentication
- The account creation and login process is secure
- Expense TrackingAdd, edit and categorize expenses
- Set a budgetSet monthly budgets for various categories
- Cost overview
- Summary of monthly expenses and budget situation
- Recurring expenses
- Manage recurring expenses with start and end dates
- Expense report
- Generate detailed reports over time
- Cost notifications: Reminders when budget limits are about to be reached or exceeded

Stakeholder requirements

- Student:
 - Track expenses visually
 - Set and adjust budgets easily
 - Detailed expense reports
 - Reliable notifications
- BudgetWise solution group:
 - Successful application development within time and budget
 - Full resource management and skills development
- University management board:
 - Promote financial management among students
 - Support responsible financial behavior



Potential Difficulties Overview



Challenges

- Limited development experience
- Strict timeline and budget
- Ensuring data privacy

Limited Development Experience * * *

Introduction	Impact
Junior Team: The development team mainly consists of junior developers with limited expertise.	Poor Performance: The application may face performance issues and software bugs due to a lack of experience in optimizing the code.
	Longer Development Time: Inexperience can lead to prolonged development as the team encounters and solves unforeseen problems.



Development Timeline Constraints

Timeline:

- **Strict Timeline:** The project has a rigid 12-week timeline from concept to launch. Adhering to this tight schedule can be challenging, especially given the team's limited experience.
- **Training Needs:** Junior developers will need training and skill development throughout the project. This additional time requirement could strain the already tight schedule.

.

Budget Constraints

Budget:

- Limited Funds: The project has a constrained budget for both development and marketing. Managing resources effectively is crucial to avoid budget overruns.
- Additional Costs: Ensuring data privacy and platform compatibility may incur extra expenses that need to be accounted for in the budget.

Data Privacy Compliance

Regulations:

- Compliance Requirements: The app must adhere to stringent data privacy regulations to ensure the security and confidentiality of user data.
- Complexity: Implementing robust data protection measures can be technically complex and time-consuming.

Others



Offline Capability

- **Functionality:** The app must operate effectively without an internet connection to cater to students in areas with limited connectivity.
- **Data Syncing:** Efficiently syncing data when connectivity is restored, without causing data loss or duplication.

Platform Compatibility

- **Platforms:**
 - **Multiple Platforms:** The app needs to be developed for Android platforms to reach a broader user base.
 - **Development Effort:** Ensuring compatibility across different platforms requires additional development and testing effort.

Highlighting and describing the issues that need to be addressed in the given scenario:

User Registration and Authentication

Issues:

- User Account Creation: Ensuring users can easily create accounts with a username and password.
- Secure Authentication: Implementing secure authentication methods to protect user data and ensure only authorized access.

Solutions:

- Account Creation: Implement a simple and intuitive registration form with validation.
- Secure Authentication: Use industry-standard authentication mechanisms, such as OAuth or JWT, and incorporate two-factor authentication for added security.

Expense Tracking

Issues:

- Adding and Editing Expenses: Allowing users to efficiently add, edit, and categorize their expenses.
- Detailed Expense Entries: Ensuring each expense entry includes a description, date, amount, and category.

Solutions:

- User Interface: Design a user-friendly interface for adding and editing expenses with dropdown menus for categories.
- Data Entry Validation: Implement validation to ensure all required fields are filled out correctly.

Highlighting and describing the issues that need to be addressed in the given scenario:

Budget Setting

Issues:

- Setting Monthly Budgets: Enabling users to set and adjust monthly budgets for different expense categories.
- Flexibility: Allowing users to adjust budget amounts as needed.

Solutions:

- Budget Interface: Provide a simple interface for setting and adjusting budgets.
- Notifications: Include notifications to alert users when they adjust their budgets.

Expense Overview

Issues:

- Summary of Expenses: Providing a comprehensive summary of monthly expenses.
- Expense Trends: Allowing users to view expense trends over time.

Solutions:

- Dashboard: Create a dashboard that summarizes total spending, remaining budget, and a breakdown by category.
- Graphs and Charts: Use visual aids like graphs and charts to display expense trends.

Highlighting and describing the issues that need to be addressed in the given scenario:

Recurring Expenses

Issues:

- Adding Recurring Expenses: Allowing users to add recurring expenses with start and end dates.
- Automatic Inclusion: Ensuring recurring expenses are automatically added to the user's monthly budget.

Solutions:

- Recurring Expense Feature: Implement a feature to add recurring expenses with flexible start and end dates.
- Automatic Updates: Ensure the app automatically includes these expenses in the monthly budget.

Expense Reports

Issues:

- Generating Reports: Allowing users to generate detailed expense reports for specific time periods.
- Category Breakdown: Including a breakdown of expenses by category in the reports.

Solutions:

- Report Generation: Implement a feature to generate detailed reports for monthly, quarterly, or annual periods.
- Detailed Breakdown: Ensure reports include a clear breakdown by category.

Highlighting and describing the issues that need to be addressed in the given scenario:

Expense Notifications

Issues:

- Budget Limit Alerts: Sending reminders or notifications when users approach or exceed their budget limits for specific categories.

Solutions:

- Notification System: Develop a notification system that sends alerts when users approach or exceed their budget limits.
- Customizable Alerts: Allow users to customize their notification preferences.

Consider possible business application solutions which address problem.

Mobile Application

Advantages:

- **Accessibility:** Students can track expenses on the go, which is crucial for managing daily expenses.
- **User Engagement:** Mobile apps generally have higher engagement rates due to push notifications and ease of access.
- **Offline Capability:** Easier to implement offline functionalities.
- **Platform Reach:** Availability on both Android and iOS increases user base.

Disadvantages:

- **Development Complexity:** Requires development expertise for both iOS and Android platforms.
- **Performance Variability:** Needs optimization for different devices and operating systems.
- **Higher Maintenance:** Continuous updates and bug fixes for multiple platforms.

Desktop Application

Advantages:

- **Performance:** Typically offers better performance and handling of large datasets.
- **User Interface:** More space for detailed visualizations and reports.
- **Development Focus:** Can focus on a single platform initially, reducing development complexity.

Disadvantages:

- **Accessibility:** Limited to when users are on their computers, which may reduce engagement for expense tracking.
- **Portability:** Lacks the convenience of mobile apps, especially for real-time expense tracking.

Web Application

Advantages:

- **Cross-Platform: Accessible**from any device with a web browser, reducing the need for separate development for Android and iOS.
- **Ease of Updates: Easier** to roll out updates and fixes compared to mobile apps.
- **Cost-Effective: Can be more budget-friendly** as it requires a single development process.

Disadvantages:

- **Offline Functionality: Harder** to implement offline capabilities compared to native mobile apps.
- **Performance: May not be as responsive or smooth** as native apps, especially with large data sets.
- **Security: Web applications can be more vulnerable** to security threats if not properly secured.

Hybrid Mobile Application

Advantages:

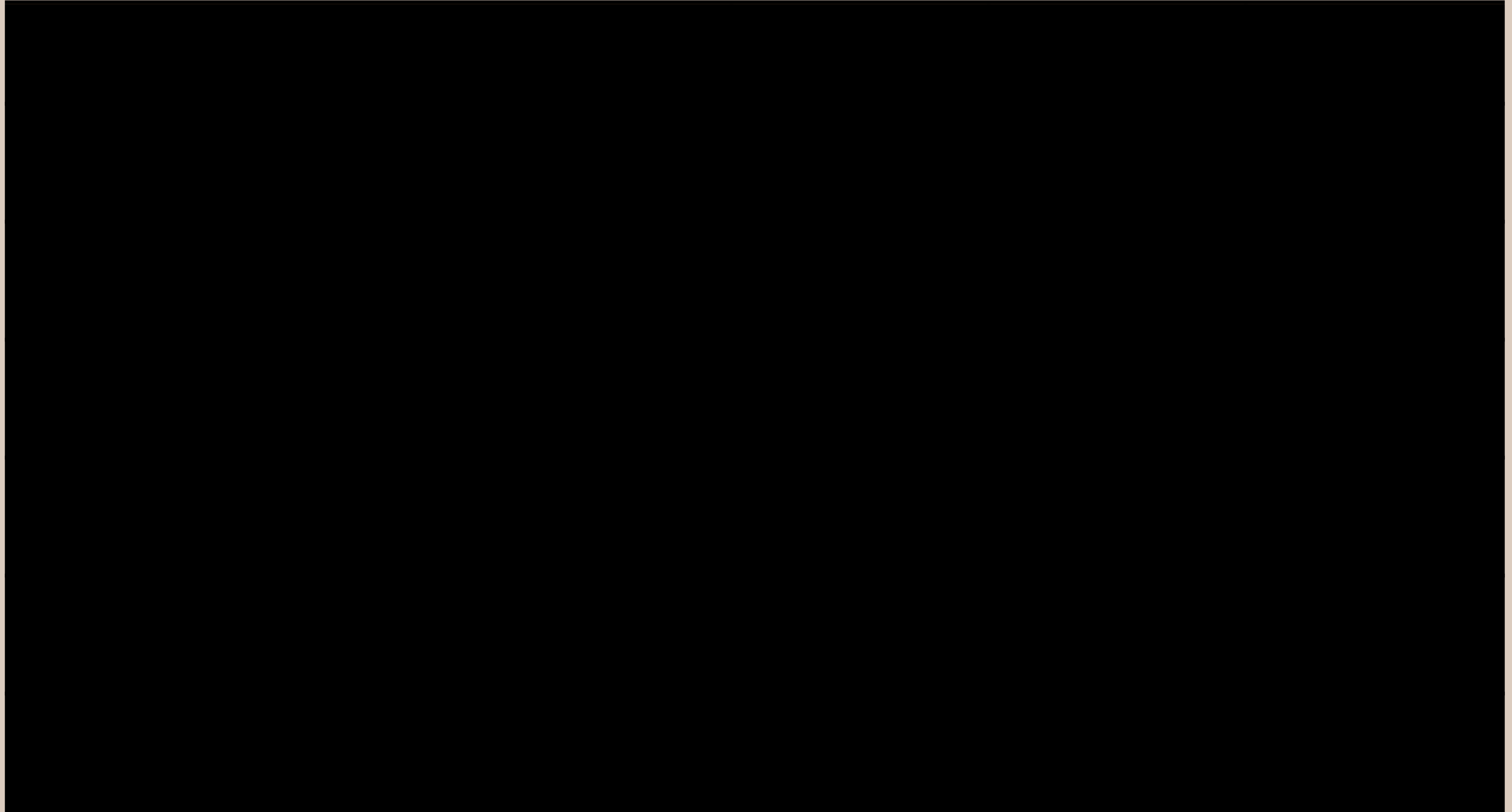
- **Cross-Platform:** Single codebase for both iOS and Android, reducing development time and cost.
- **Offline Capability:** Easier to implement offline features compared to web applications.
- **Faster Development:** Quicker to develop and deploy than native apps for both platforms.

Disadvantages:

- **Performance:** May not be as optimized as native apps.
- **User Experience:** Might not provide the same level of user experience as native apps due to limitations in accessing device-specific features.

Areas of risk pertaining to this software project

Risk



Technical Complexity

- Developing features like recurring expenses, real-time expense tracking, and data synchronization across platforms may prove technically challenging for a team with limited experience.

Timeline Constraints:

- The strict 12-week timeline may lead to rushed development, compromising on quality or completeness of features.

Resource Constraints

- Limited budget and a team of junior developers may impact the quality and speed of development.

Data Security and Privacy:

- Mishandling of user data could lead to breaches, compromising user trust and legal compliance.

Platform Compatibility:

- Inconsistencies in app performance between Android and iOS platforms may lead to user dissatisfaction.

User Acceptance and Feedback

- Failure to meet user expectations could result in low adoption rates or negative reviews.

Strategies for managing these risks

Identifying Risk

- Risks can be identified passively or through active tools and processes.
- Proactive risk management is essential for effective risk reduction.
- Regular internal and external risk assessments uncover hidden risks.
- Compliance frameworks often require annual risk assessments.
- Document all risks, assessments, response plans, and resolutions in a formal register.
- Regularly update and review the risk register to ensure it remains current.

Strategies for managing these risks

Assessing Risk

- Assess each identified risk to determine its likelihood and potential impact.
- This assessment helps prioritize risks for effective management.
- Systematic and documented risk assessments are essential.
- Depending on your business, review or redo assessments at least annually.
- Frequency of risk assessments varies based on business size and complexity.

Strategies for managing these risks

Responding to Risks

- Develop and implement treatments and controls after assessing risks.
- Four common ways to treat risks: risk avoidance, mitigation, acceptance, and transference.
- Responding to risks may involve ongoing projects or immediate action.
- Some risks require detailed action plans for effective management.
- Involve affected stakeholders in decision-making about key risks.

Strategies for managing these risks

Monitoring Risks

- Risk monitoring involves ongoing management by tracking risk management execution.
- It includes identifying and managing new risks as they arise.
- The purpose is to take prompt action if a risk's likelihood, severity, or impact exceeds acceptable levels.
- Continuous monitoring and execution of risk plans ensure readiness for various risk types.
- Risks monitored encompass enterprise, financial, strategic, and external categories.
- Vigilant monitoring is crucial to minimize disruptions and losses.

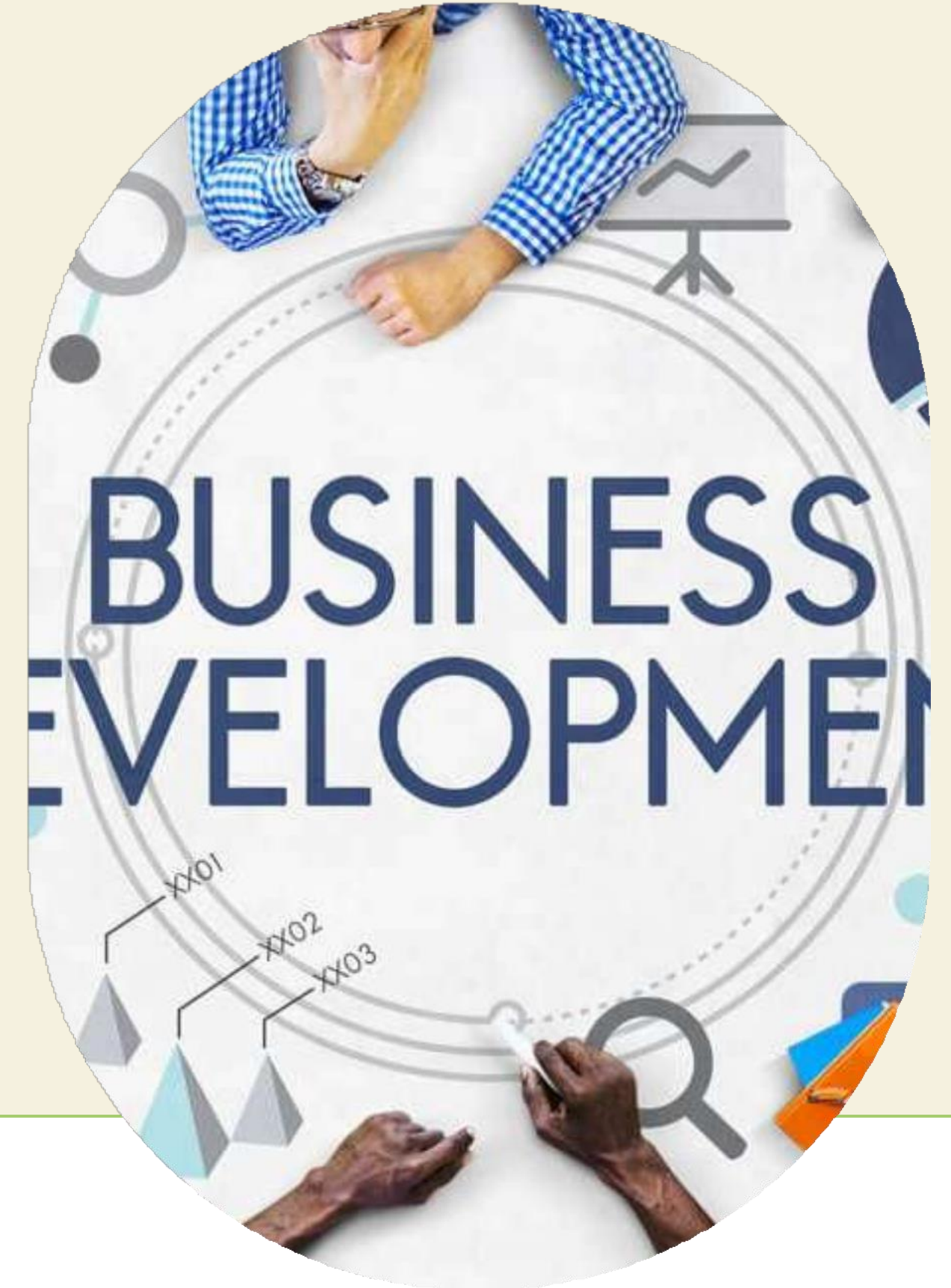
Presentation on Developing a Business Application with Android Studio and SQLite

Group 6

Nguyen Huy Hoang

Nguyen Tuan Anh

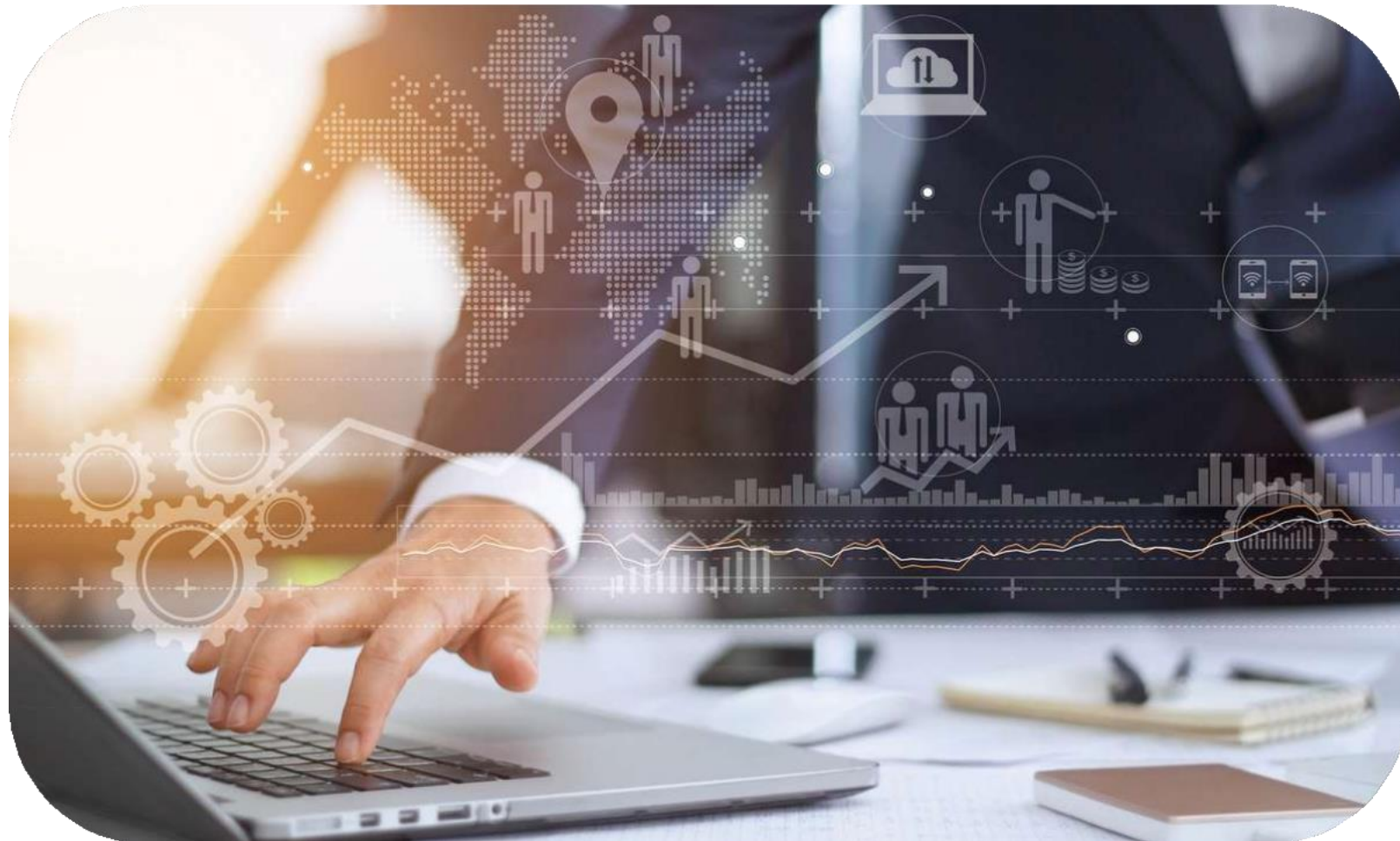
Tran Van Duy





Contents:

Developing a Business Application with
Android Studio and SQLite



[Quay lại Trang Chương trình](#)

Developing a Business Application with Android Studio and SQLite

BudgetWise Solutions

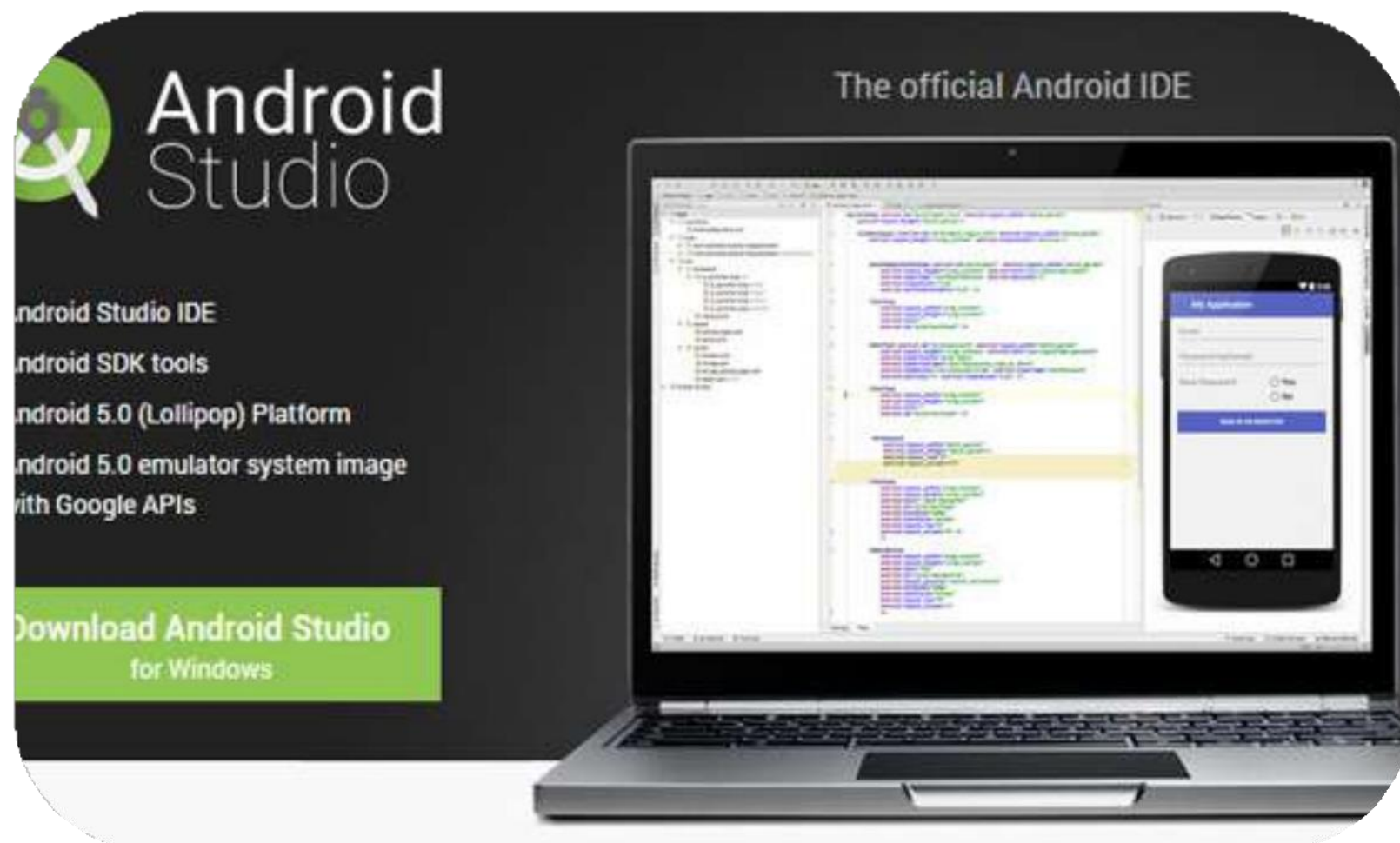
- BudgetWise Solutions is dedicated to creating innovative solutions to simplify financial management for individuals and businesses.

Objective of the Application

- Our goal is to develop a functional business application that helps users manage their expenses and budgets effectively.



Android Studio

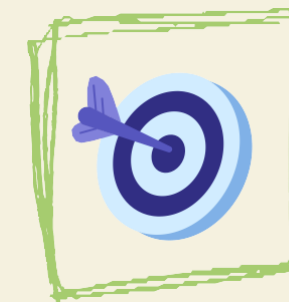


Android Studio: The Platform for Android Application Development



Overview

Android Studio is the official integrated development environment (IDE) for Android app development



Key Features

Supports Java and Kotlin, the two main programming languages for Android

Provides comprehensive tools for designing, coding, testing, and deploying Android applications.



Significance

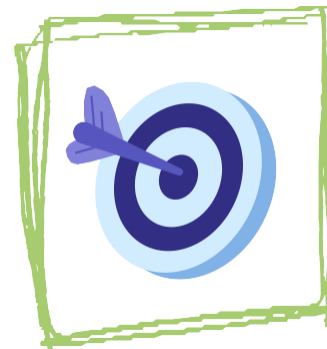
Android Studio streamlines the development process and enhances productivity, making it an essential tool for Android developers

Benefits of Android Studio



Intelligent Code Editor

Features like auto-completion, error detection, and code refactoring help improve coding efficiency and reduce errors.



Visual Interface Designer

Drag-and-drop design tools allow for easy creation and modification of user interfaces

The visual layout editor supports both XML and design view, facilitating a seamless design process



Integrated Emulator

The built-in emulator enables testing on a wide range of virtual devices, ensuring compatibility across different screen sizes and configurations.



SQLite

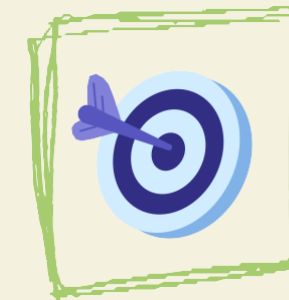


SQLite: A Lightweight Database Engine



Overview

SQLite is a self-contained, serverless, and zero-configuration database engine widely used for local data storage in mobile applications.



Key Features:

- Supports ACID transactions, ensuring reliable data operations
- Stores data in a single cross-platform database file, making it easy to manage and distribute
- Offers a simple SQL-based interface for data manipulation.



Purpose

SQLite provides a reliable and efficient data storage solution for applications, allowing developers to focus more on the app's functionality and user experience.

Key Features of SQLite

Data Storage and Management



- Stores all data in a single file, making it easy to transfer and manage.
- Supports a wide range of data types and complex queries, enabling versatile data operations

Lightweight and Fast



Designed to be lightweight, with a minimal footprint on the device's storage and memory

Efficient performance, even on devices with limited resources

No Configuration Required



- Does not require a separate server or installation, simplifying deployment and maintenance
- Zero-configuration approach allows for seamless integration into the app.

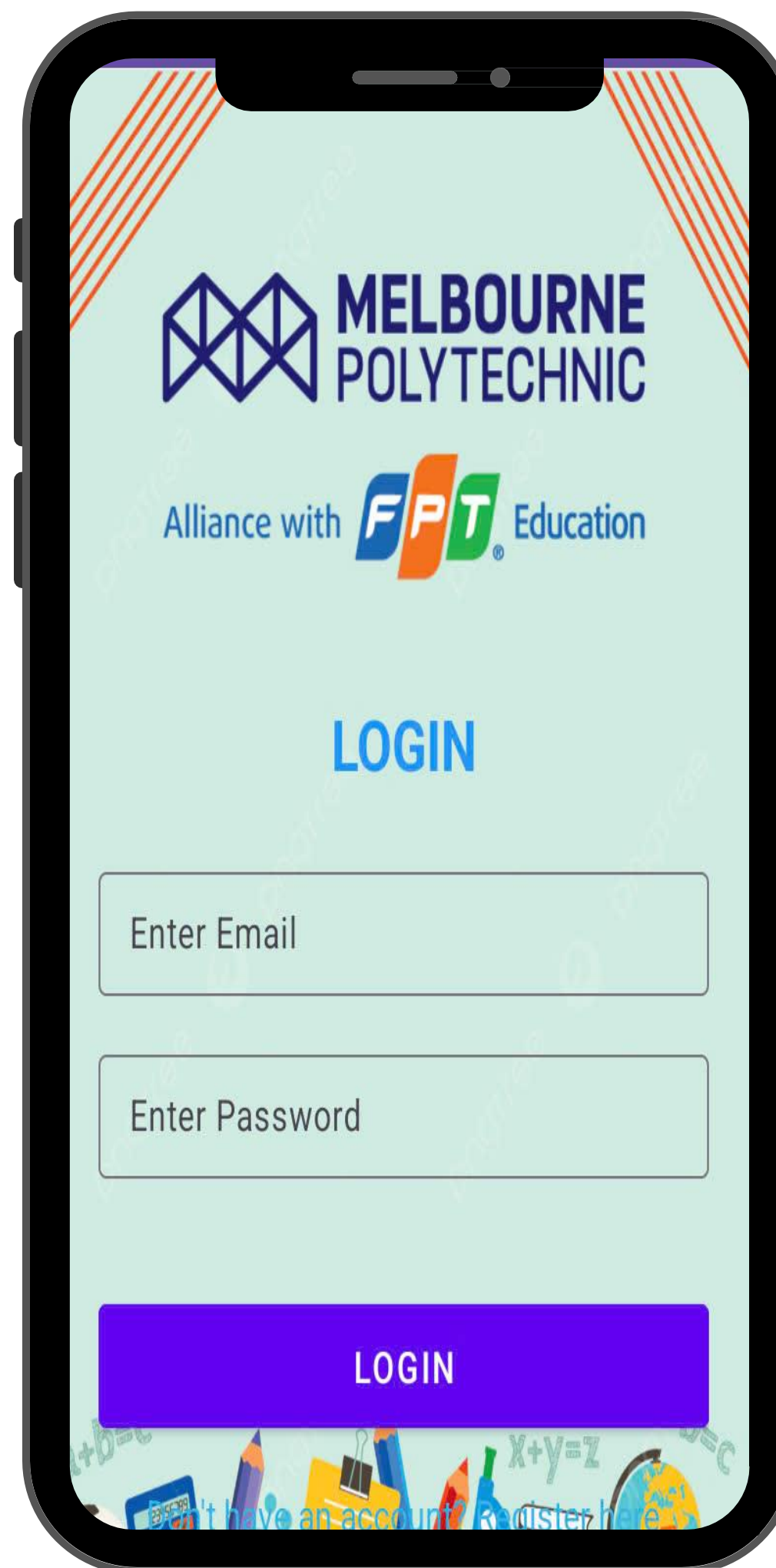
Cross-Platform Compatibility



- Compatible with various operating systems, ensuring broad applicability.
- Supports data synchronization with other systems and databases through custom implementations

Application Development Process

Login and SignUp Form



The login form is displayed on a smartphone screen. At the top, it features the Melbourne Polytechnic logo and the text "Alliance with FPT Education". Below this, the word "LOGIN" is centered in blue. There are two input fields: "Enter Email" and "Enter Password". A large purple button labeled "LOGIN" is at the bottom. The background has a light blue gradient with orange diagonal stripes in the corners and a decorative footer with educational icons and formulas.

MELBOURNE
POLYTECHNIC

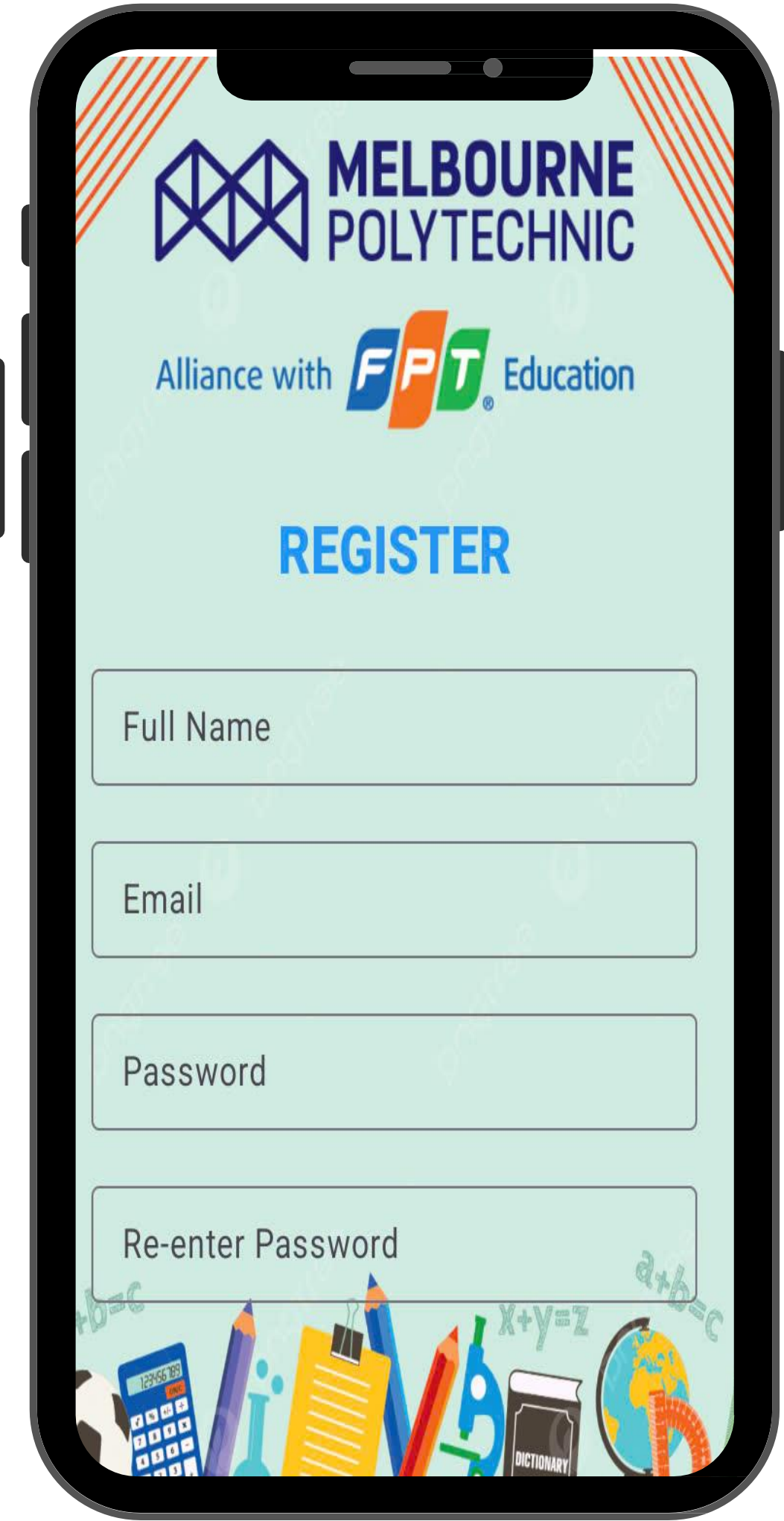
Alliance with **FPT** Education

LOGIN

Enter Email

Enter Password

LOGIN



The register form is displayed on a smartphone screen. At the top, it features the Melbourne Polytechnic logo and the text "Alliance with FPT Education". Below this, the word "REGISTER" is centered in blue. There are four input fields: "Full Name", "Email", "Password", and "Re-enter Password". A large purple button labeled "REGISTER" is at the bottom. The background has a light blue gradient with orange diagonal stripes in the corners and a decorative footer with educational icons and formulas.

MELBOURNE
POLYTECHNIC

Alliance with **FPT** Education

REGISTER

Full Name

Email

Password

Re-enter Password

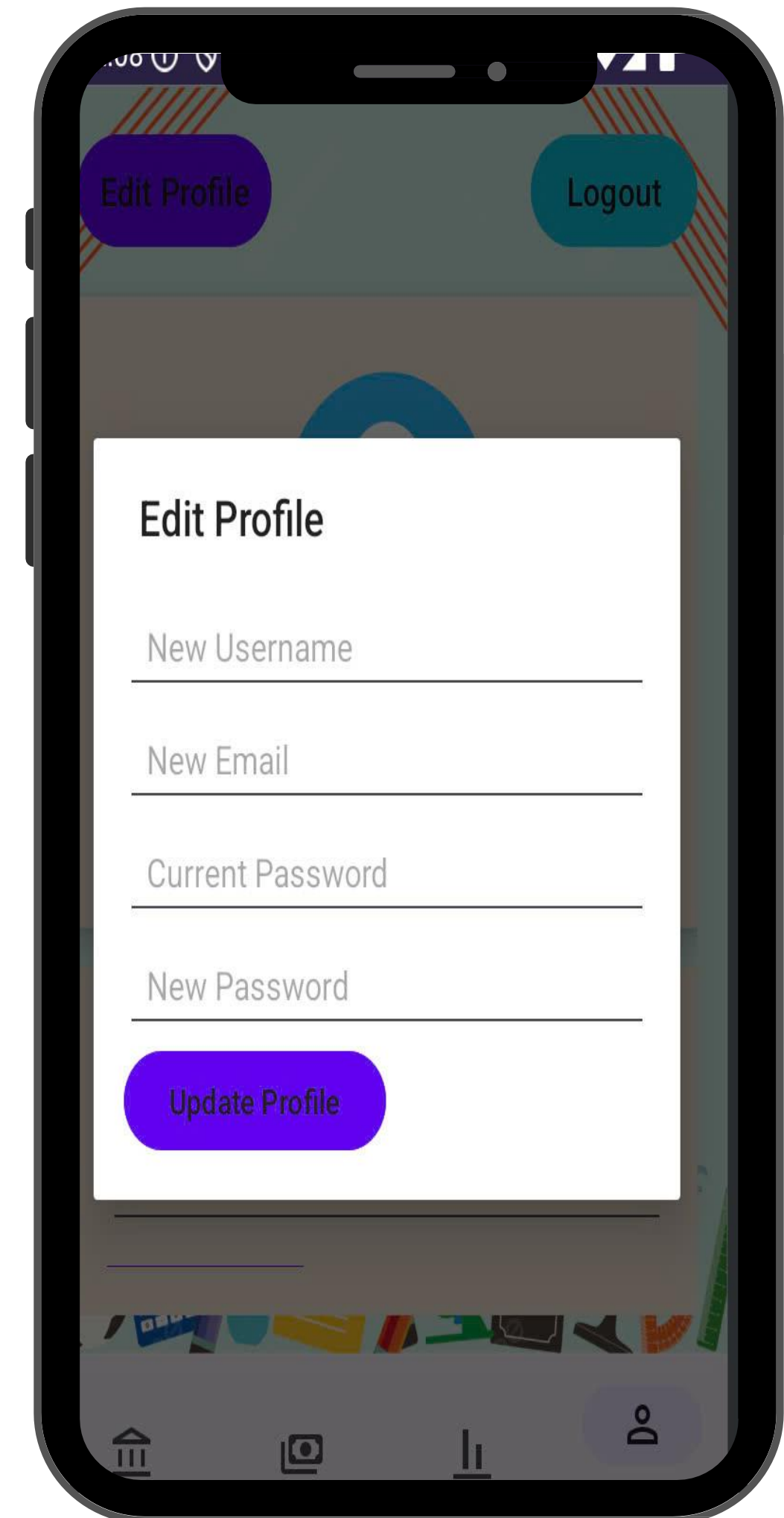
REGISTER

Application Development Process



Login

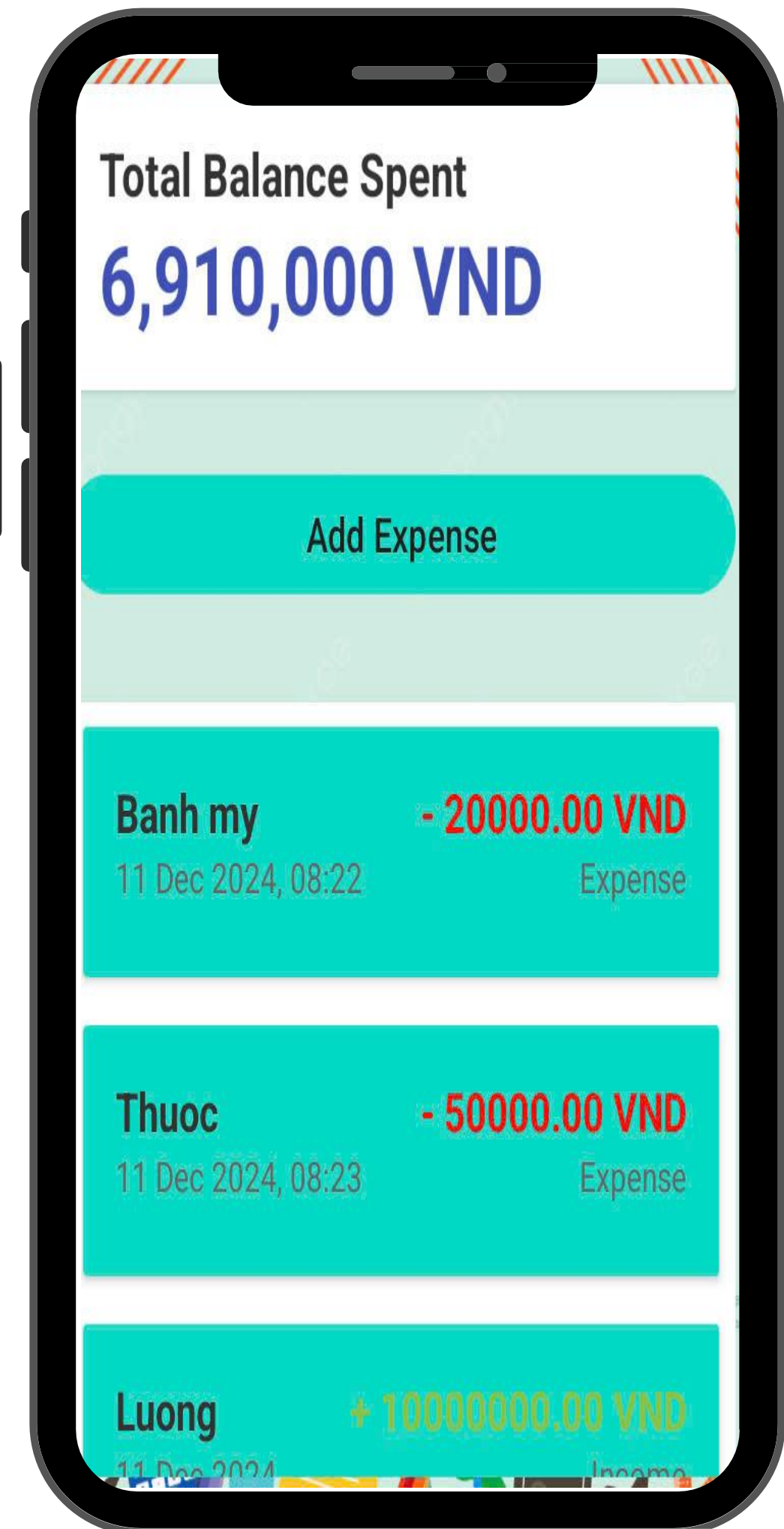
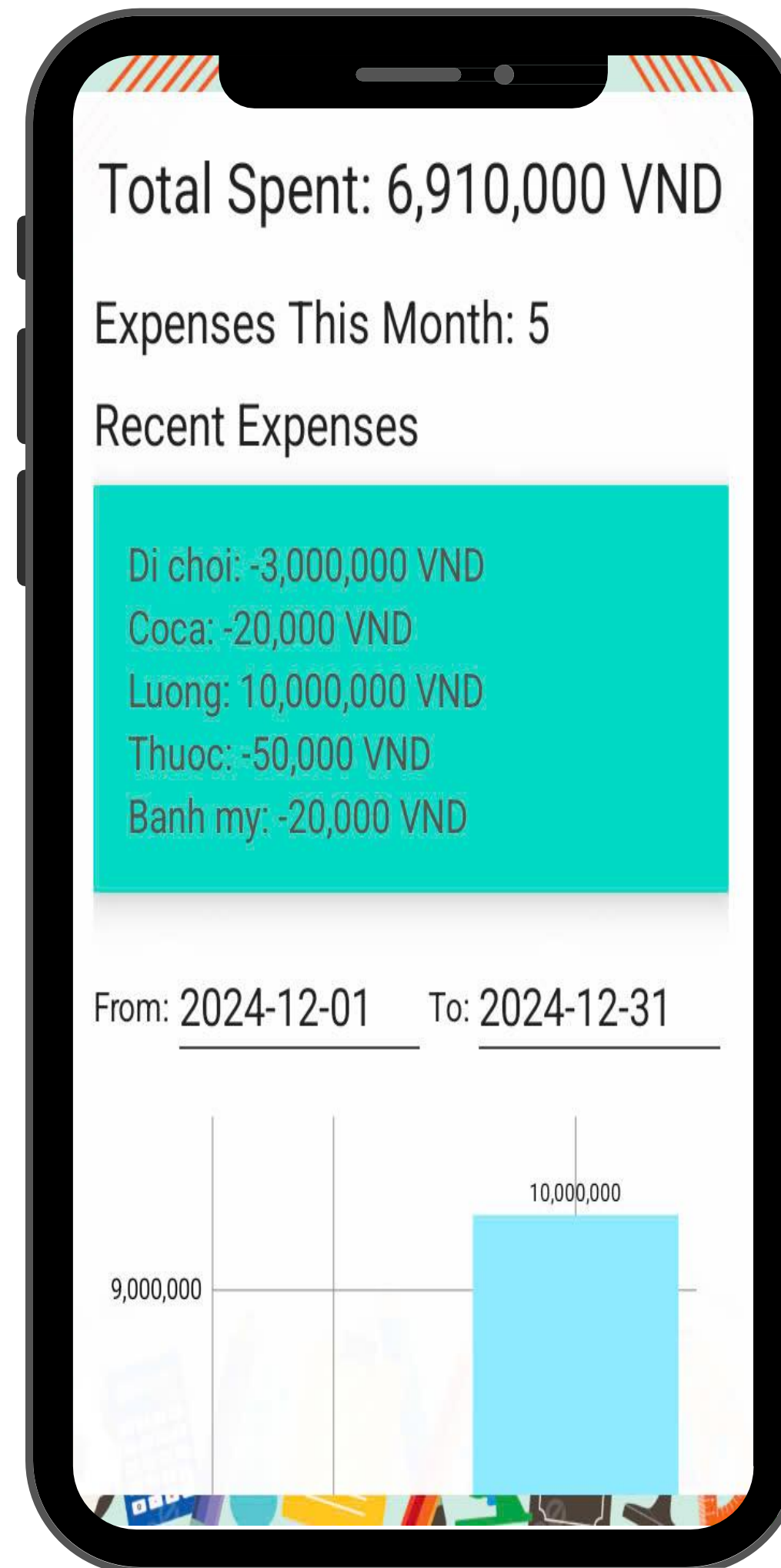
When you forgot password



Application Development Process

Categories

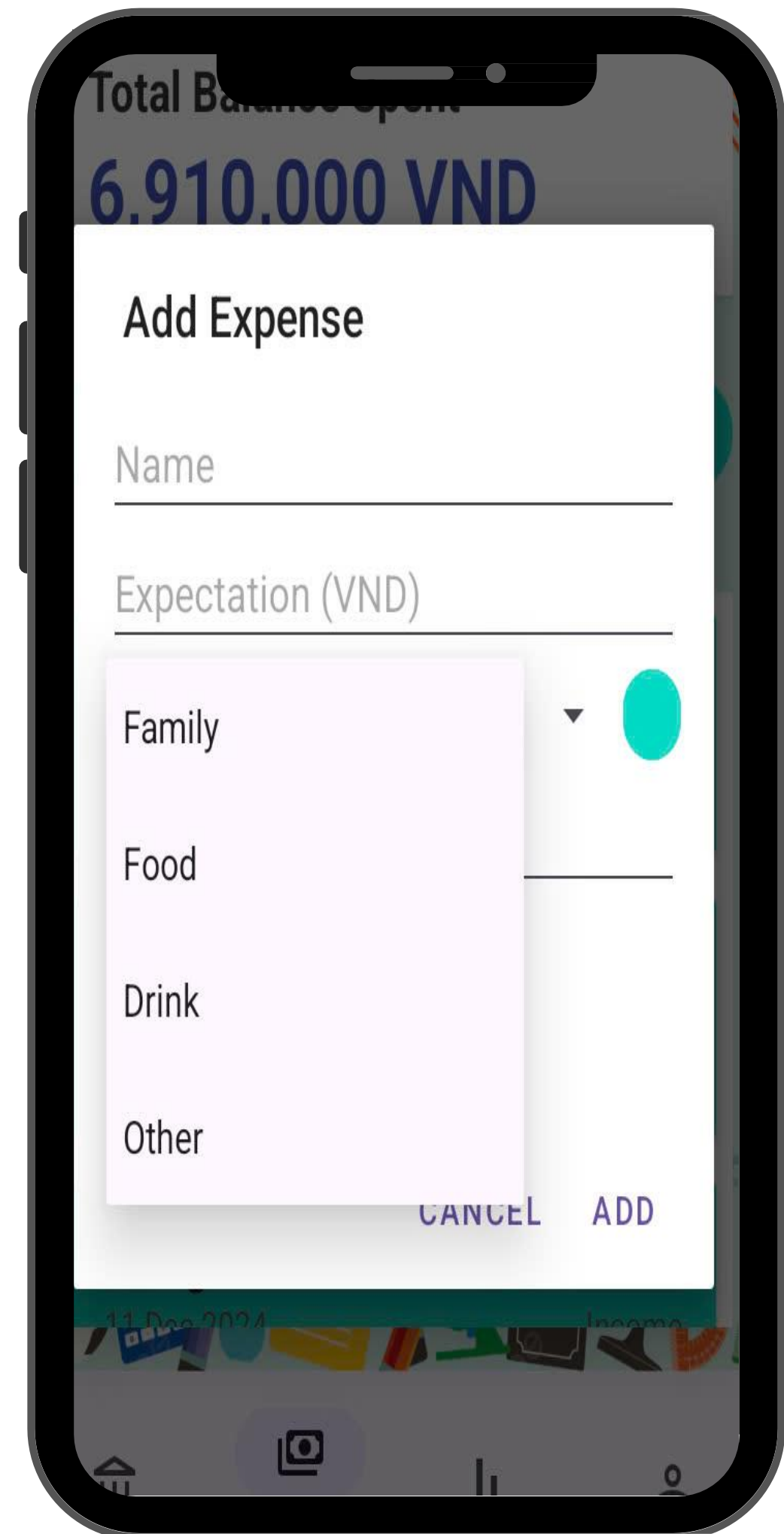
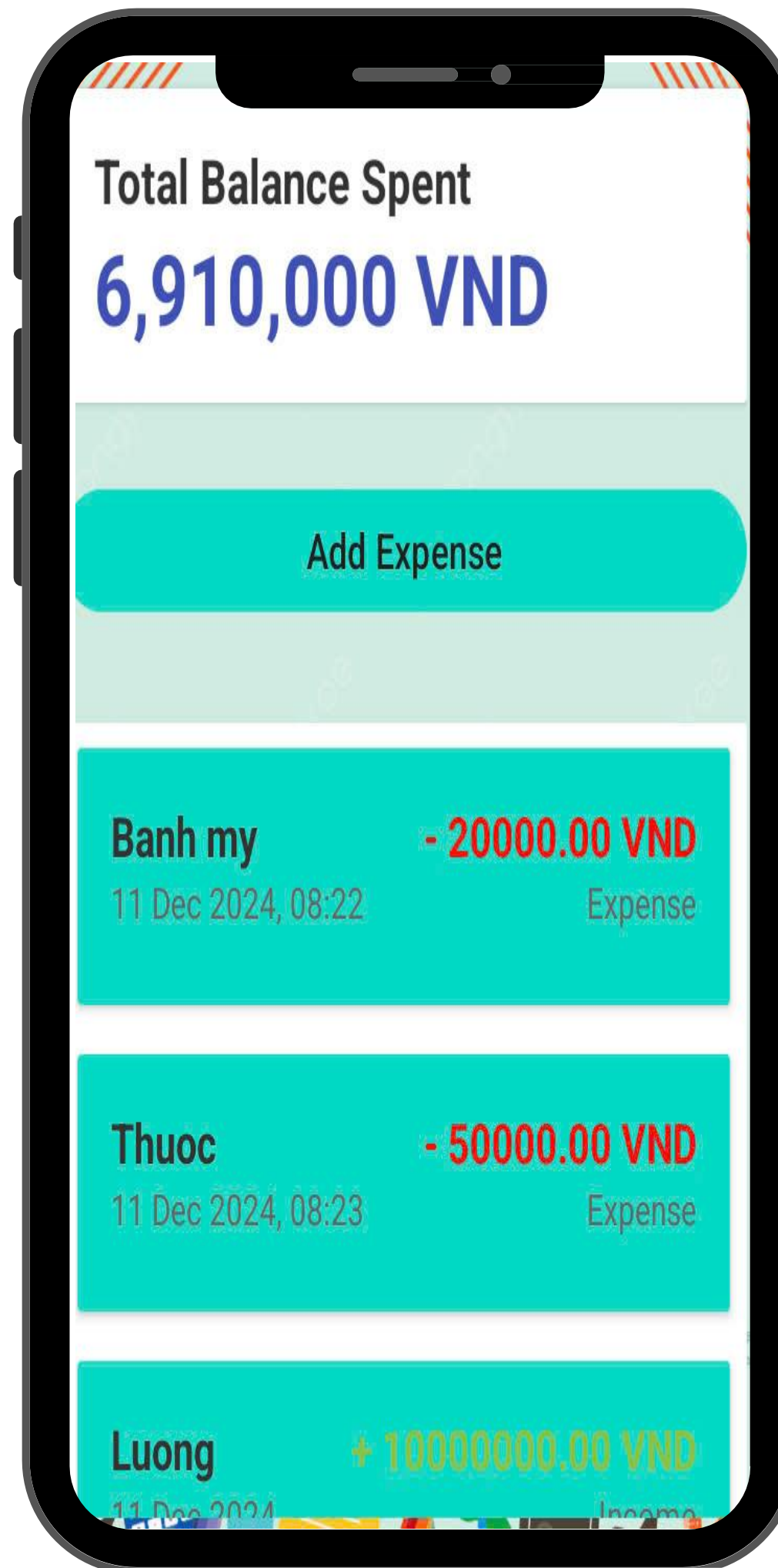
Track monthly expenses



Application Development Process

Add expense

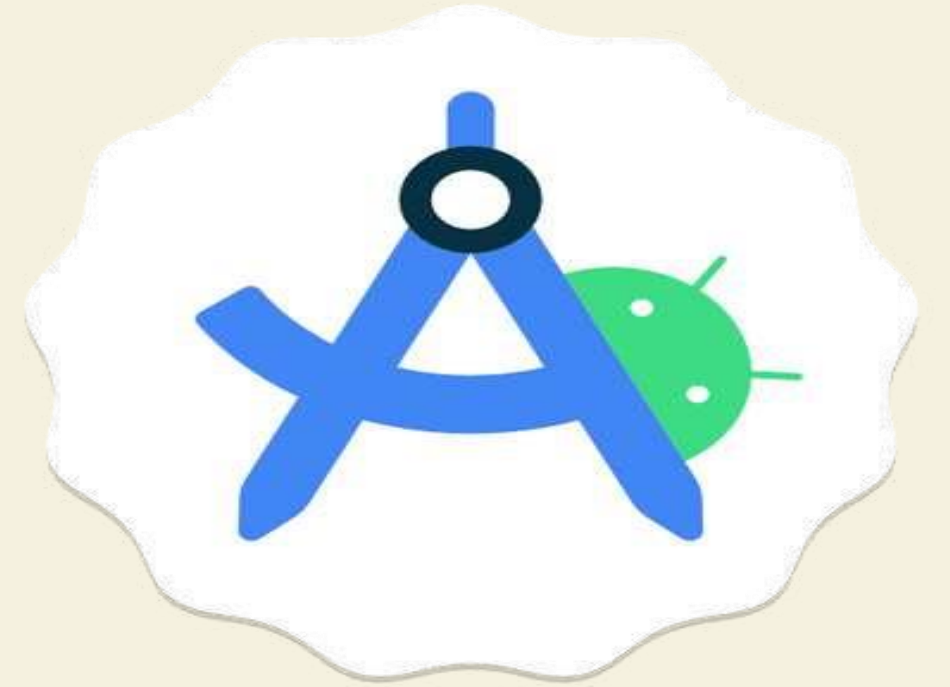
Add expense by category



Project development process.

What tools did the development team use during the development process?

- Android Studio
- SQLite
- Microsoft Project for project management



Is this app easy to use for beginners?

- With information logging operations as well as the interface, we have designed it appropriately to help users easily operate and use it, even if they are new.



Who is this application for?

- This application will be suitable for students in school who want to manage their personal finances effectively and completely automatically.



What goals has your team achieved during development?

- This application will be suitable for students in school who want to manage their personal finances effectively and completely automatically.



During the development process, what difficulties did the development team encounter?

- Resource Constraints: With a limited budget and a team of junior developers, resource management was crucial.
- Data Privacy and Security Compliance: Adhering to data privacy regulations and ensuring the security of user data added complexity to the development process.
- Mobile App Development Experience: As a team with limited experience in mobile app development, the developers faced a steep learning curve in understanding best practices, tools, and frameworks for both Android and iOS platforms.

How did the development team solve those difficulties?

- The team invested in training sessions and online courses to build the necessary skills. They also sought advice from more experienced mobile app developers through forums and professional networks.
- The team prioritized core functionalities for the initial release, deferring less critical features to future updates. They also leveraged open-source libraries and tools to minimize costs.
- The team implemented robust encryption methods for data storage and transmission. They also conducted thorough reviews to ensure compliance with regulations such as GDPR and CCPA.

Application Overview

- ✓ User Registration and Authentication Requirement
- ✓ Expense Tracking Requirement:
- ✓ Budget Setting Requirement
- ✓ Expense Overview Requirement
- ✓ Recurring Expenses Requirement
- ✓ Expense Reports Requirement
- ✓ Expense Notifications Requirement

User Registration and Authentication Requirement

- Users can create accounts with a username and password. Authentication should be secure and allow users to log in and access their expense data.

Expense Tracking Requirement:

- Users can add, edit, and categorize expenses (e.g., rent, groceries, transportation). Each expense entry includes a description, date, amount, and category.

Budget Setting Requirement

- Users can set monthly budgets for various expense categories (e.g., food, entertainment, education). The app should allow users to adjust budget amounts as needed.

Expense Overview Requirement

- The app provides a summary of monthly expenses, including total spending, remaining budget, and a breakdown by category. Users can view expense trends over time.

Recurring Expenses Requirement

- Users can add recurring expenses (e.g., monthly rent) with start and end dates. The app should automatically add these expenses to the user's monthly budget.

Expense Reports Requirement

- Users can generate detailed expense reports for specific time periods (e.g., monthly, annually). Reports include a breakdown of expenses by category.

Expense Notifications Requirement

- The app should send reminders or notifications when users approach or exceed their budget limits for specific categories.

Quality assessment

- Performance: The app provides a smooth and responsive user experience, even with large amounts of data spend.
- Friendly Interface: Intuitive user interface, with clear labels and easy navigation to track spending with ease.
- Data Security: User data, including spending and budget information, is stored securely and protected with encryption. Security and privacy practices are followed to ensure user information is kept confidential.
- Feedback and Support: There is an in-app feedback form for users to report issues or make suggestions. The BudgetWise Solutions team actively monitors user feedback and resolves issues promptly.

Thanks
for
Watching

The text is written in a black, elegant cursive script. It is surrounded by several decorative green elements: a large swirl on the left, a smaller swirl above the 'k' in 'Thanks', a star above the 's' in 'Thanks', a heart above the 'i' in 'Watching', and a clock icon to the right of the 'g' in 'Watching'. There are also some smaller, fainter green swirls.