Effective expense management app called "Campus Expense Manager "

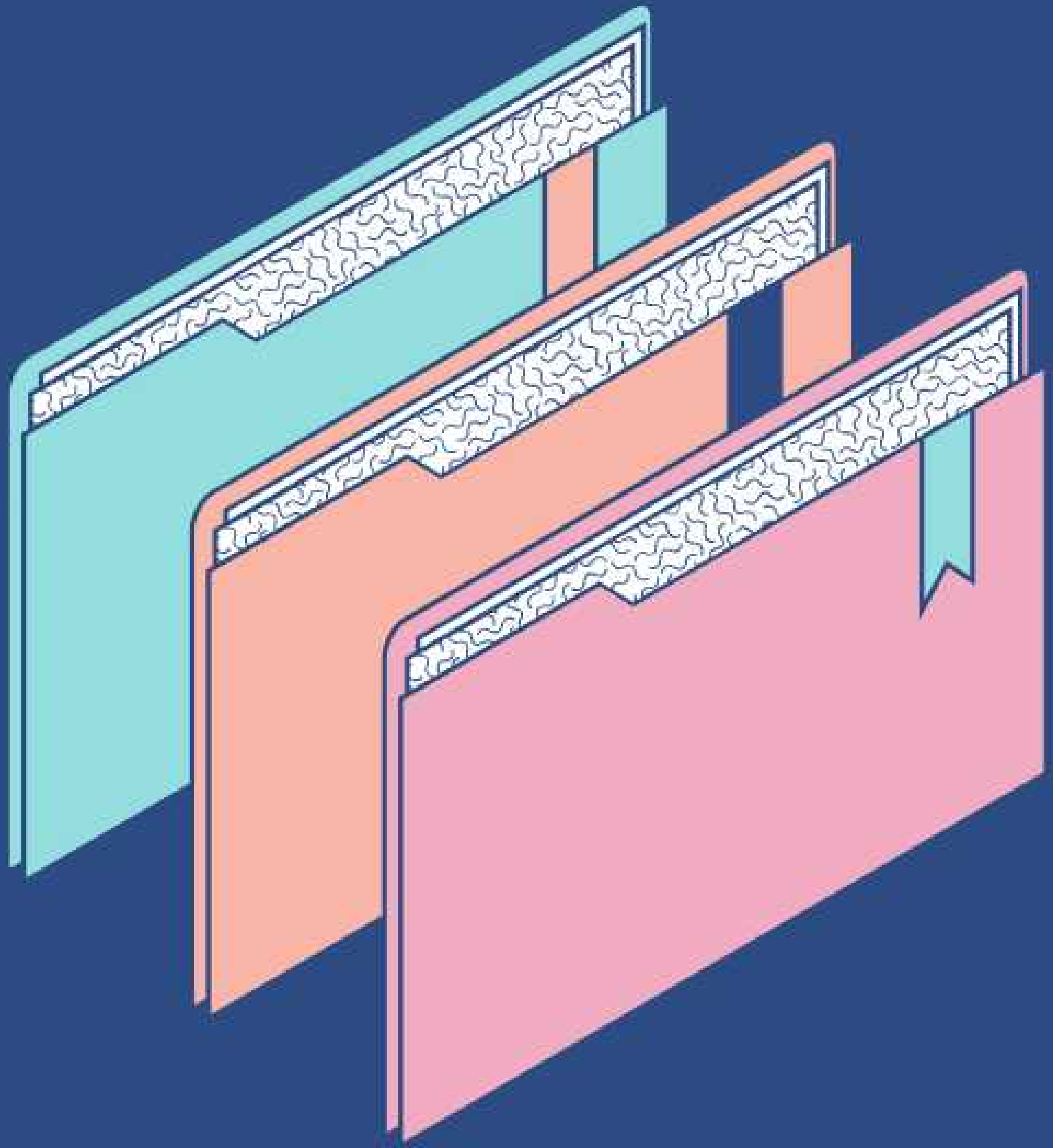
Group 5:

Nguyen Anh Tu
Tran Thanh Tung
Nguyen Tien Dung
Do Viet Anh

Date: 12/02/2024

Project Overview

The CampusExpense Manager is a mobile application designed to help university students manage their finances effectively. It allows users to track expenses, set budgets, manage recurring costs, and generate detailed reports, all through an intuitive and user-friendly interface. The app aims to simplify financial management for students, whether they live on or off-campus, by providing tools to promote budget discipline and deliver valuable financial insights. With features like offline functionality, platform compatibility, and robust data security, the app ensures accessibility and privacy for its primary target audience: university students striving to maintain financial stability during their academic journey.

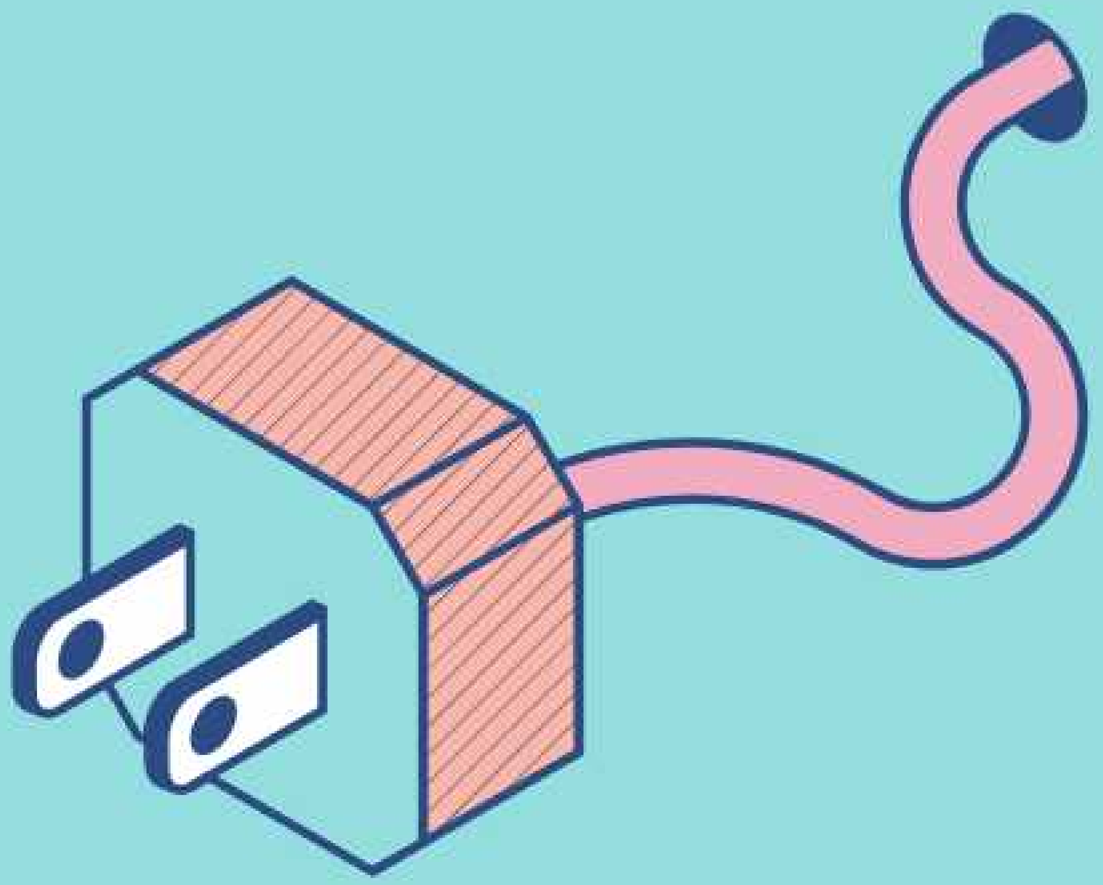


Objectives of the App

- Track Expenses: Log and categorize spending for better financial awareness.
- Budgeting: Set and manage monthly budgets for various categories.
- Ease of Use: Provide an intuitive, user-friendly interface for students to easily manage finances.
- Recurring Expense Management: Automate the addition of regular expenses like rent and subscriptions.
- Financial Insights: Provide detailed reports and trends to help students make informed financial decisions.
- Offline Functionality: Ensure the app works seamlessly without an internet connection for students with limited access.

Key user requirements identified during initial analysis:

- Expense Tracking: Allow users to log, edit, and categorize expenses efficiently.
- Budget Alerts: Notify users when they approach or exceed their budget limits.
- Intuitive UI: Provide a user-friendly interface for easy navigation and functionality.
- Categorization of Expenses: Enable organization of expenses into predefined or custom categories.
- Summary Reports: Generate detailed reports to review spending trends and financial summaries.



Systems Investigation and Research

Research Phase

Methods: Conducted surveys (50+ students), interviews (10 participants), and market research.

Key Findings:

- Difficulty tracking daily expenses.
- Lack of awareness of spending habits.
- Challenges managing group expenses.
- Need for simple, intuitive tools.

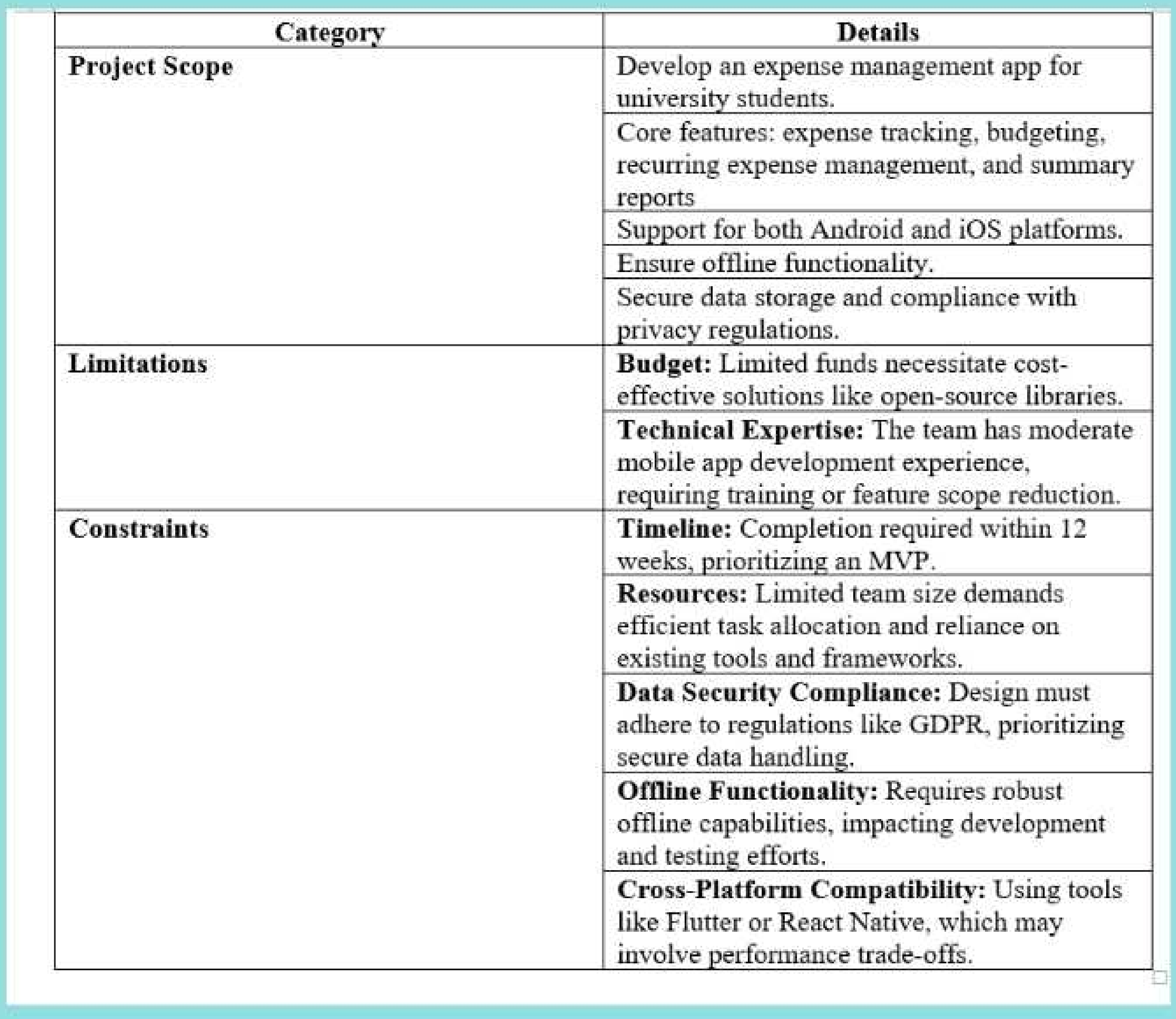
Competitor Analysis

- Mint: Feature-rich but complex; requires bank linkage.
- Splitwise: Great for group expenses but lacks personal expense tracking.
- YNAB: Goal-driven but expensive.
- Expensify: Useful for receipts but not student-focused.
- Conclusion: Competitors lack student-specific features and simplicity.

Student Needs

- Ease of Use: Quick expense logging, automation.
- Budget Tools: Alerts for overspending.
- Visual Summaries: Charts/graphs for spending patterns.
- Group Expense Management: Bill splitting, debt tracking.
- Privacy: Manual logging, no mandatory bank linkage.

Project Scope and Constraints

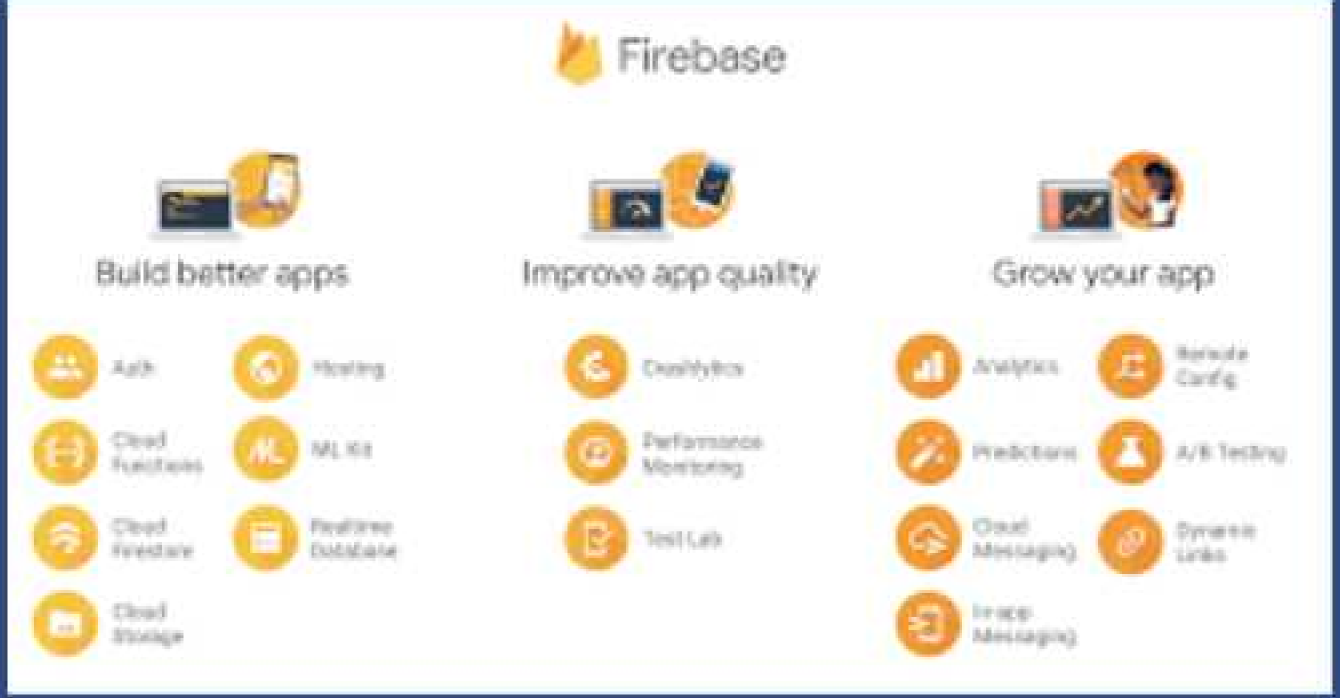


Category	Details
Project Scope	Develop an expense management app for university students.
	Core features: expense tracking, budgeting, recurring expense management, and summary reports
	Support for both Android and iOS platforms.
	Ensure offline functionality.
	Secure data storage and compliance with privacy regulations.
Limitations	**Budget:** Limited funds necessitate cost-effective solutions like open-source libraries.
	Technical Expertise: The team has moderate mobile app development experience, requiring training or feature scope reduction.
Constraints	**Timeline:** Completion required within 12 weeks, prioritizing an MVP.
	Resources: Limited team size demands efficient task allocation and reliance on existing tools and frameworks.
	Data Security Compliance: Design must adhere to regulations like GDPR, prioritizing secure data handling.
	Offline Functionality: Requires robust offline capabilities, impacting development and testing efforts.
	Cross-Platform Compatibility: Using tools like Flutter or React Native, which may involve performance trade-offs.

Tools and Technologies Used

Firebase

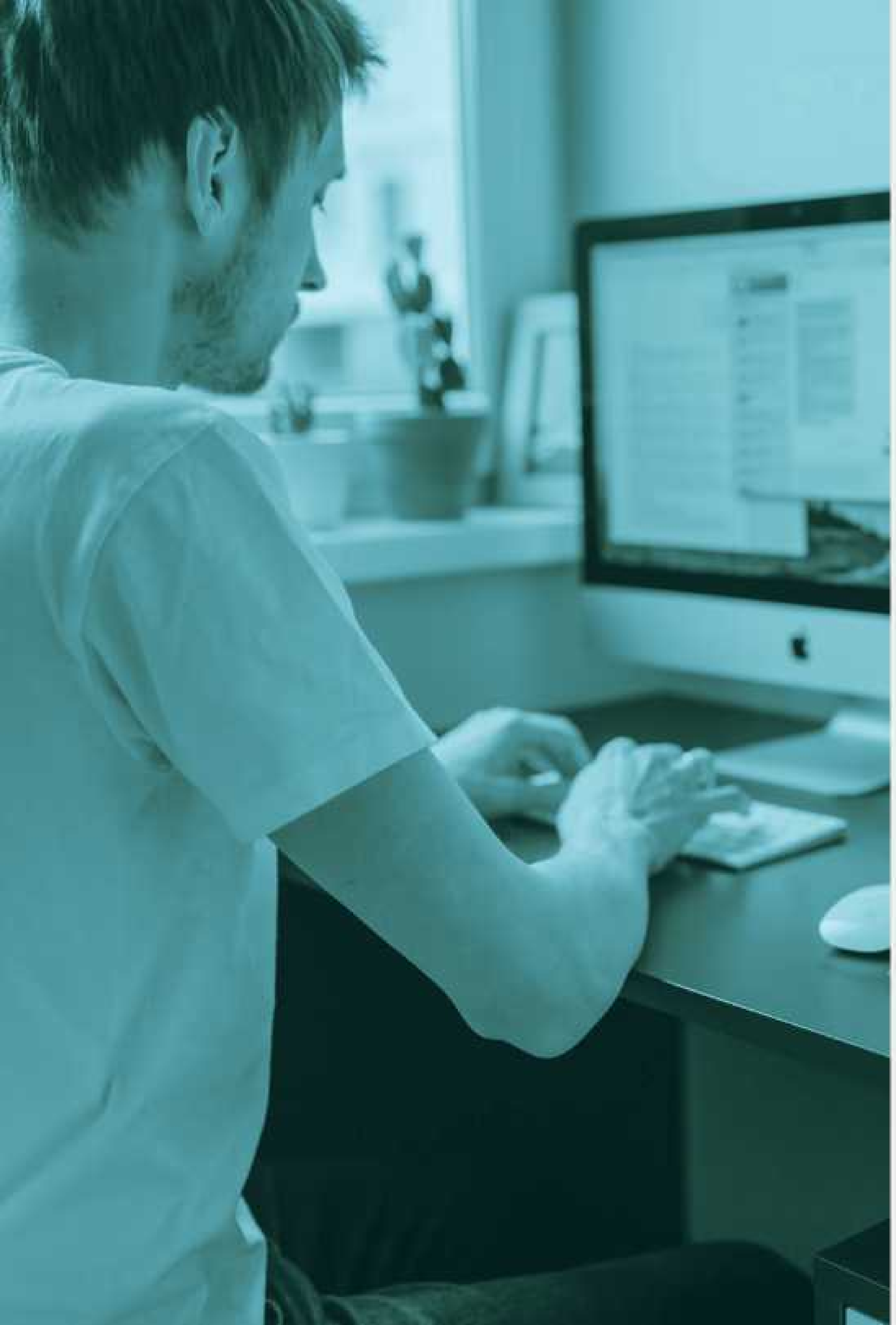
Google Firebase is Google-backed application development software which allows developers to develop Android, IOS, and Web apps. For reporting and fixing app crashes, tracking analytics, creating marketing and product experiments.



Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android App development. It is a powerful tool that allows developers to build high-quality applications for the Android platform. From writing code to testing and deployment, ANdroid studio has all the functionalities for developers to develop an Android App.

- Android Studio IDE
- Android SDK tools
- Android 5.0 (Lollipop) Platform
- Android 5.0 emulator system image with Google APIs



Explanation of the chosen methodology

Agile Methodology Overview

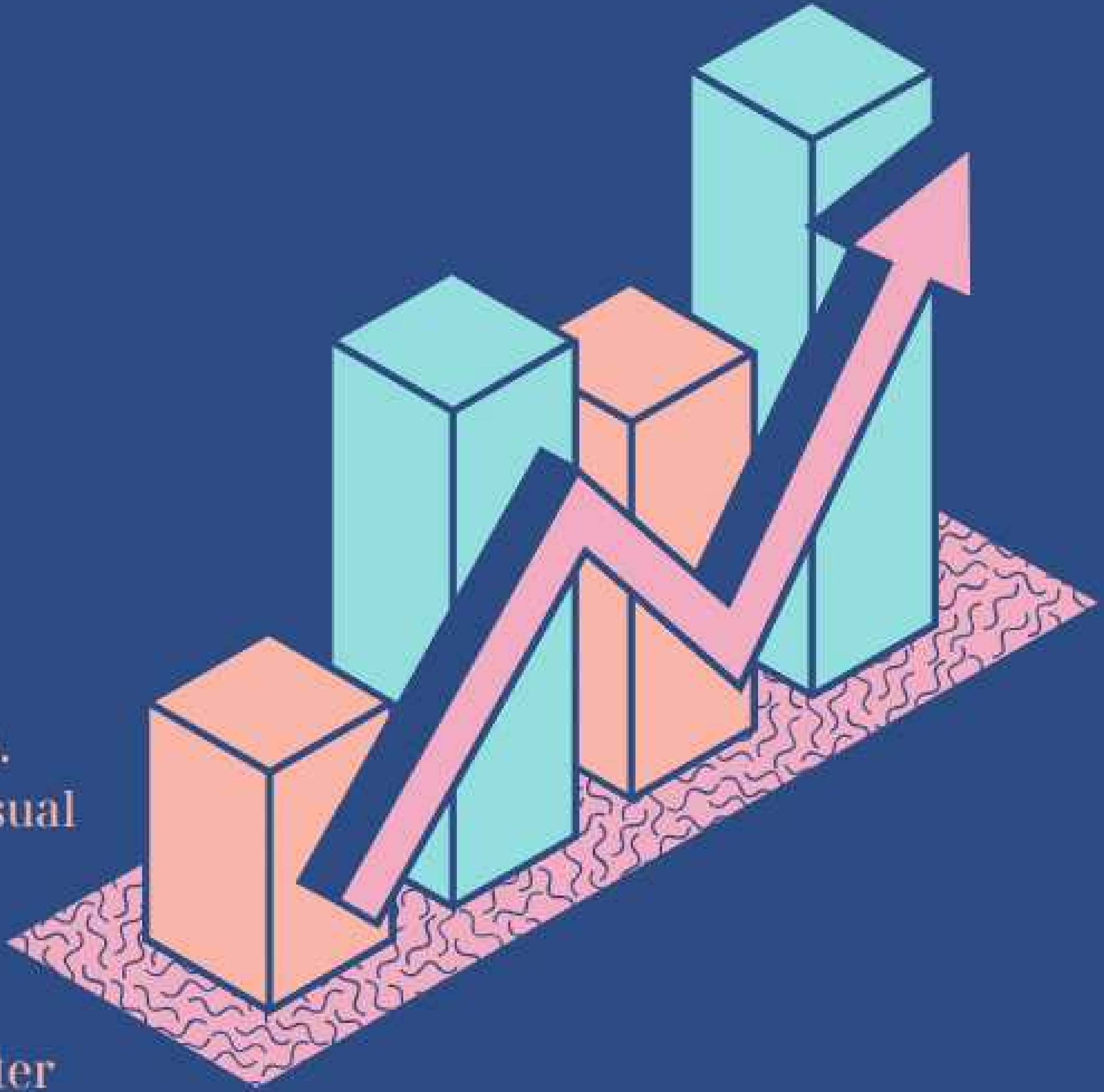
Agile is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and customer-centric solutions. It focuses on breaking down the project into smaller, manageable cycles called sprints (typically 2–4 weeks long) to deliver functional increments of the product. Agile encourages adaptive planning, continuous improvement, and quick response to change.

Why Agile Was Chosen for the Project:

- Dynamic and Flexible Requirements: Projects involving features like student information management systems or data management applications often have evolving requirements.
- Frequent Deliverables: Delivering functional increments of the system allows stakeholders to review progress frequently, ensuring alignment with the goals of the project.
- Collaboration and Communication: Agile emphasizes collaboration within the team and with stakeholders.
- Focus on Quality: By integrating regular testing and feedback loops in each sprint, Agile helps maintain a high-quality codebase.

Initial Design and Prototyping

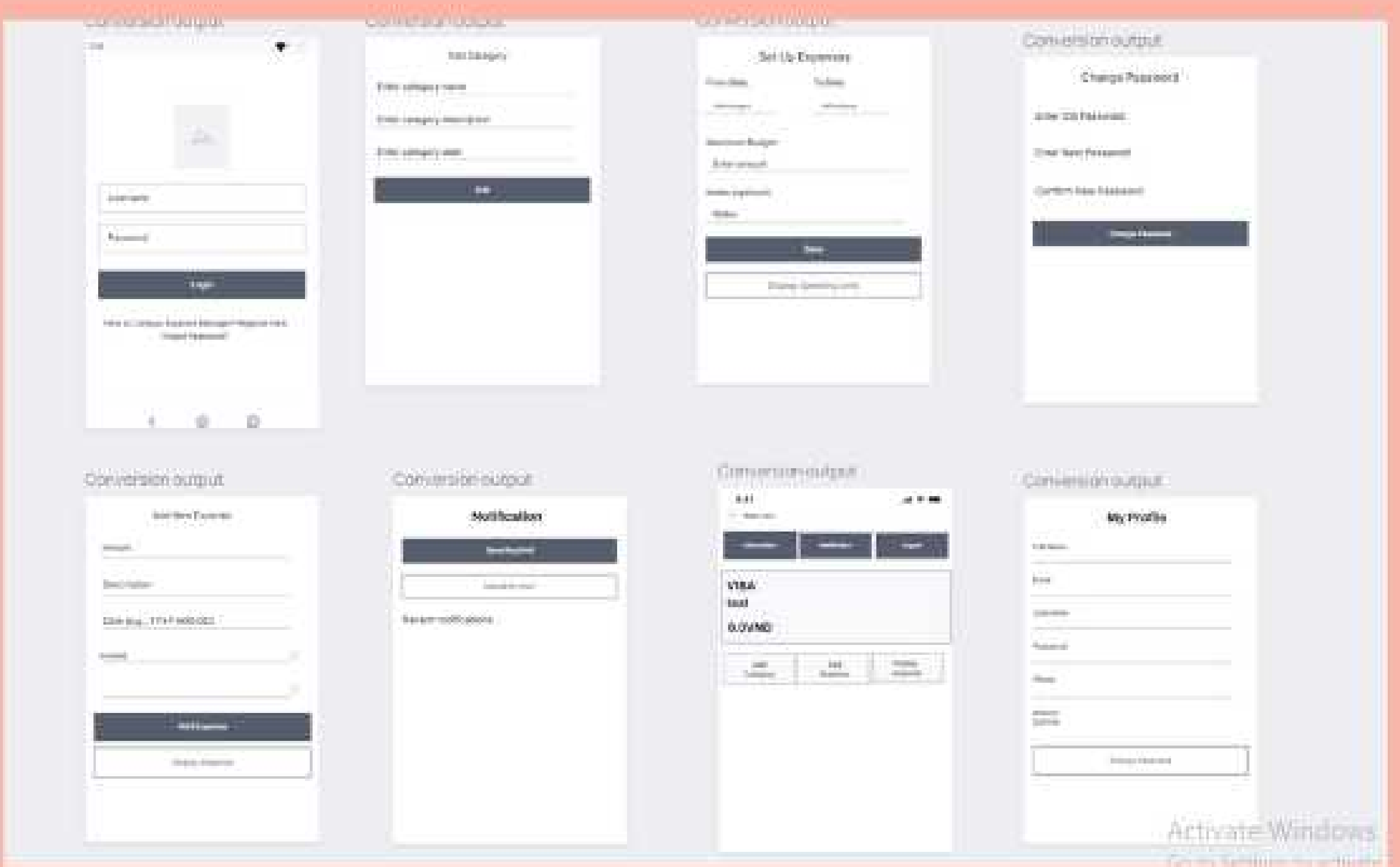
Purpose: Visualize app structure (wireframes) and simulate workflows (prototypes) to validate design early.



Process:

- Requirement Gathering: Identify key features and user personas (e.g., students managing budgets).
- Wireframes: Low-fidelity sketches for layout and navigation flow.
- Feedback: Gather input from students on usability and missing features.
- Prototypes: High-fidelity, interactive designs reflecting feedback and visual themes.
- Testing: Usability tests with real tasks (e.g., adding expenses, setting budgets).
- Iteration: Address feedback (e.g., clearer navigation, category icons, better charts).

User Interface (UI) Design

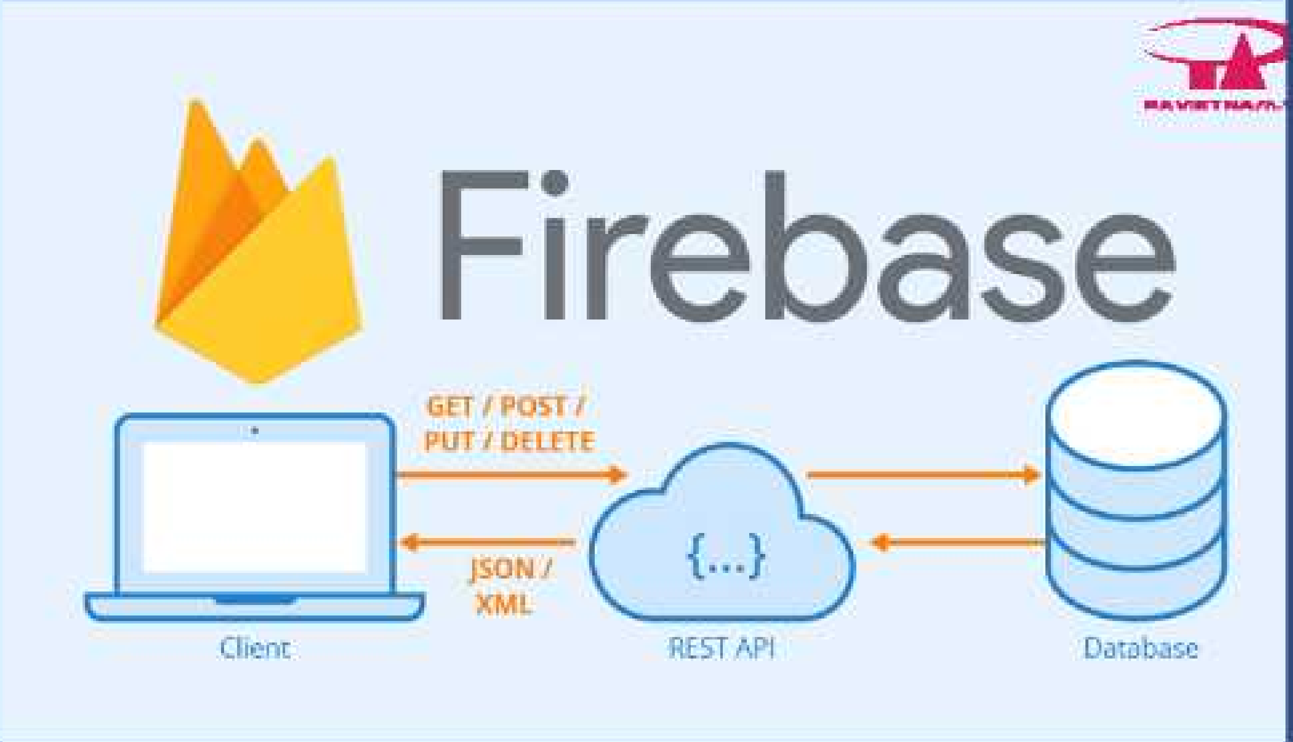


- Login: Simple interface for secure access with options to register or recover the password.
- Add Category: Define expense categories with name, description, and date fields.
- Set Up Expenses: Configure budget limits within a specified date range.
- Change Password: Secure password update with old and new credentials.
- Add New Expense: Log expense details (amount, description, date) with income tracking.
- Notifications: View spending limits, charts, and recent alerts.
- Dashboard: Central hub for navigation, expense summary, and quick actions (add display expenses).
- My Profile: Manage user details and account information, with a password update option.

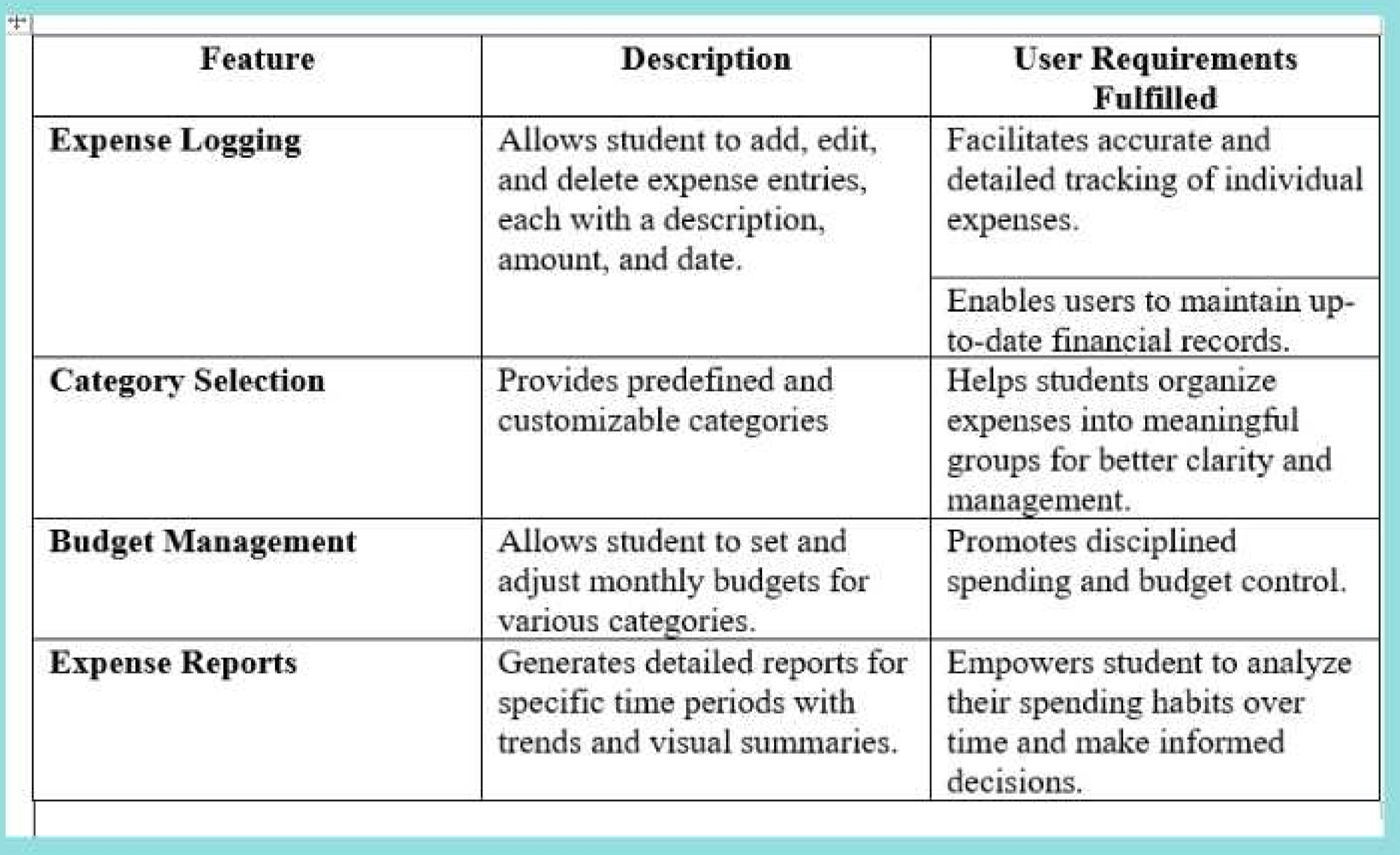
Overview of backend choices

Backend Choice: Firebase

Firebase, a Backend-as-a-Service (BaaS) platform by Google, offers a suite of tools for building scalable, secure, and high-performing applications. For this project, Firebase was chosen to manage user data storage and authentication due to its robust features, scalability, and ease of integration.



Core Features Implemented



Feature	Description	User Requirements Fulfilled
Expense Logging	Allows student to add, edit, and delete expense entries, each with a description, amount, and date.	Facilitates accurate and detailed tracking of individual expenses.
		Enables users to maintain up-to-date financial records.
Category Selection	Provides predefined and customizable categories	Helps students organize expenses into meaningful groups for better clarity and management.
Budget Management	Allows student to set and adjust monthly budgets for various categories.	Promotes disciplined spending and budget control.
Expense Reports	Generates detailed reports for specific time periods with trends and visual summaries.	Empowers student to analyze their spending habits over time and make informed decisions.

Explanation of extra features and the prioritization based on user feedback.

Extra Features

Manual Categorization:

- Lets users create custom categories for personalized expense organization.
- Prioritized for flexibility and addressing diverse spending habits.

Notifications for High Spending:

- Alerts users when they approach or exceed budgets.
- Crucial for promoting financial discipline and avoiding overspending.

Prioritization

- High Priority: Notifications and category customization, as they directly meet user needs.
- Medium Priority: Expense reports for long-term insights.
- Low Priority: Advanced analytics and bank integration for future updates due to complexity and time constraints.

Data Management and Security

Firebase Authentication:

- Provides secure login and user authentication with encrypted credentials.
- Protects against unauthorized access.

Real-Time Database:

- Ensures data is stored securely with end-to-end encryption.
- Allows instant data synchronization across devices.

Data Privacy Compliance:

- Adheres to regulations such as GDPR to safeguard user information.
- Implements strict access controls and data minimization practices.

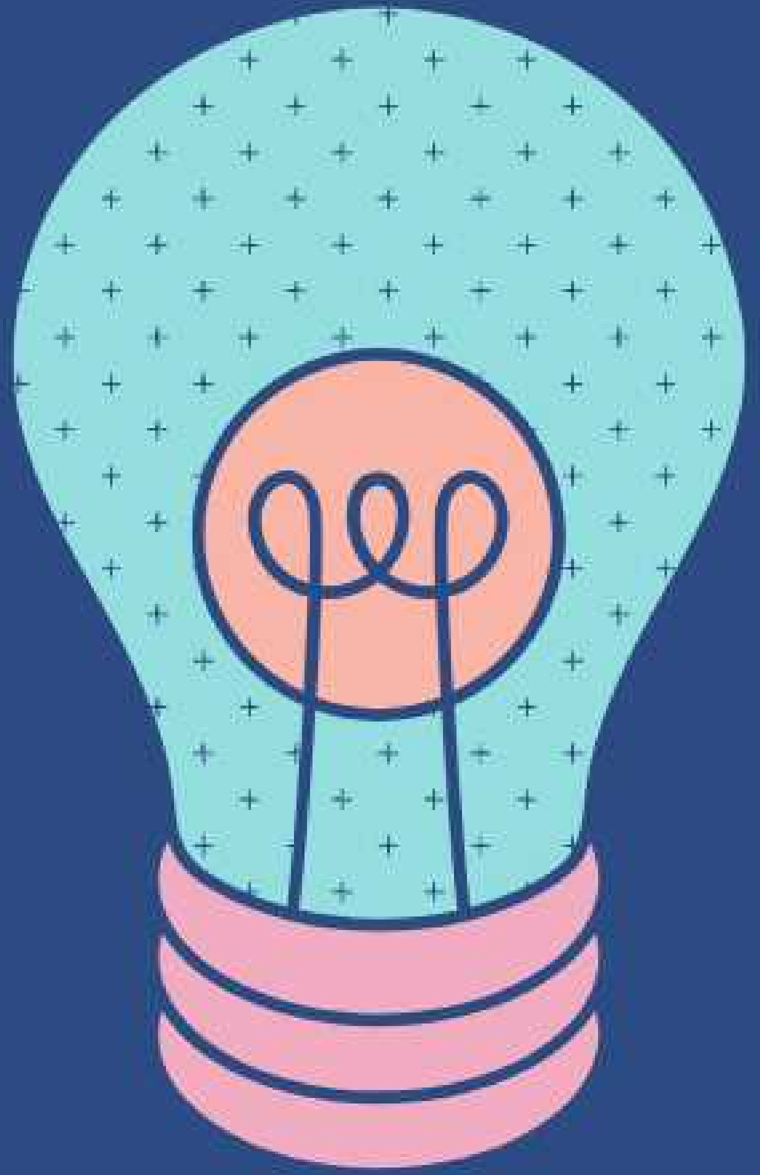
User Testing and Feedback

Testing Process:

- Participants: 10 university students.

Tasks Tested:

- Register and log in.
- Add and categorize expenses.
- Set and adjust budgets.
- View expense summaries and trends.



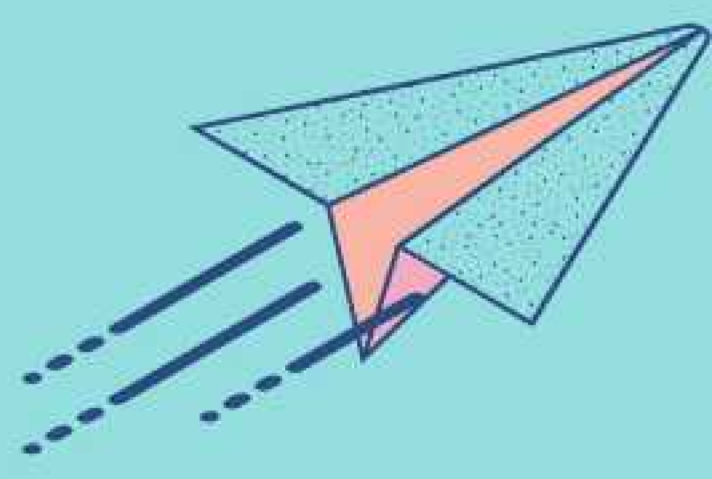
Feedback Highlights:

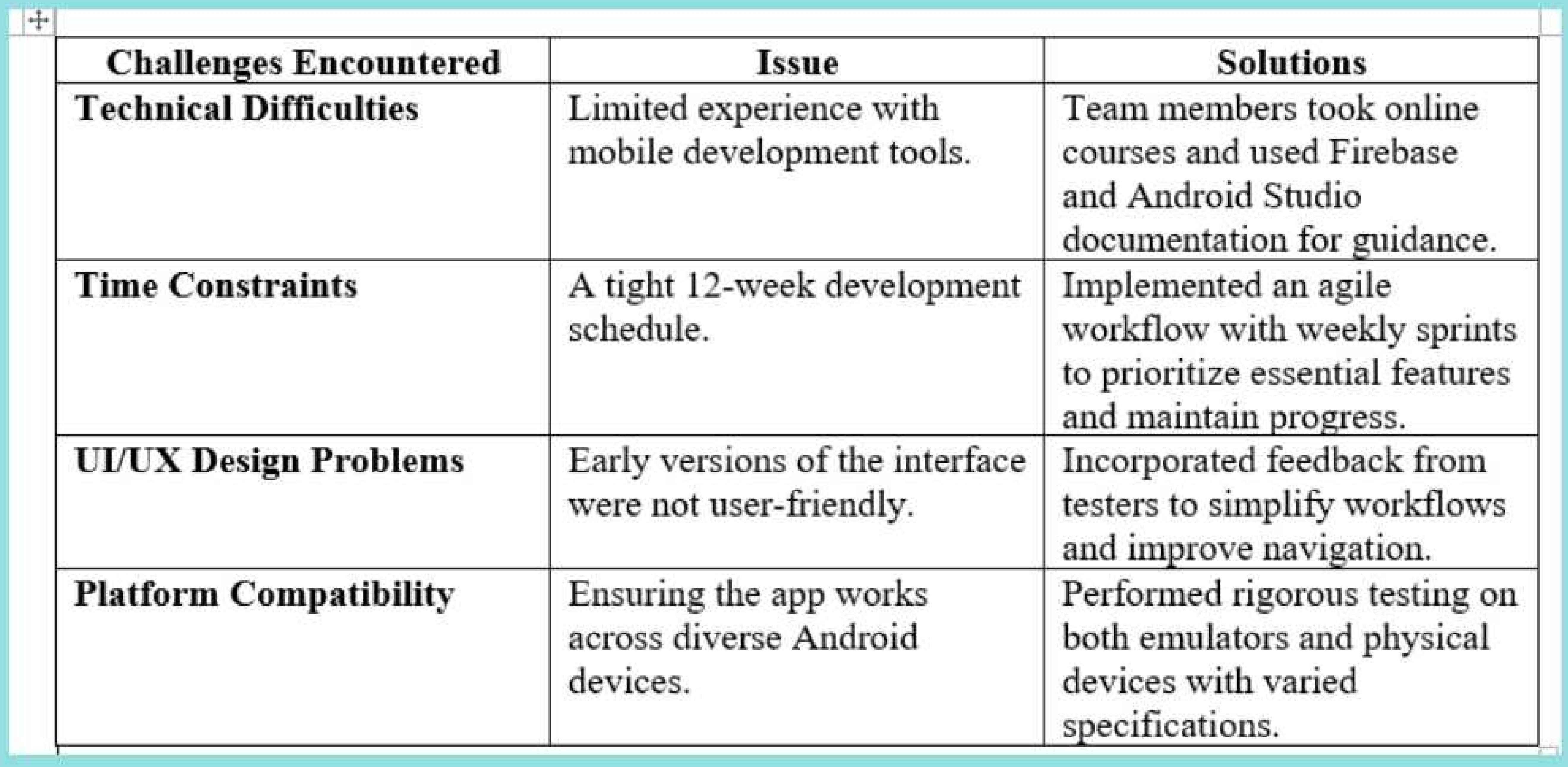
- Navigation: Users found recurring expenses difficult to locate.
- Button Placement: "Add Expense" button was not easily accessible.
- Visuals: Charts were helpful but needed weekly/monthly toggles.
- Complexity: Budget setting workflow felt cumbersome.

Changes Made:

- Navigation: Added a dedicated tab for recurring expenses.
- Button Placement: Introduced a floating "Add Expense" button on the Dashboard.
- Visuals: Added a toggle for switching between weekly and monthly views in charts.
- Workflow Simplification: Streamlined budget setup with pre-filled category suggestions.

Discussion of challenges encountered and how the team addressed them.





Challenges Encountered	Issue	Solutions
Technical Difficulties	Limited experience with mobile development tools.	Team members took online courses and used Firebase and Android Studio documentation for guidance.
Time Constraints	A tight 12-week development schedule.	Implemented an agile workflow with weekly sprints to prioritize essential features and maintain progress.
UI/UX Design Problems	Early versions of the interface were not user-friendly.	Incorporated feedback from testers to simplify workflows and improve navigation.
Platform Compatibility	Ensuring the app works across diverse Android devices.	Performed rigorous testing on both emulators and physical devices with varied specifications.

QUALITY ASSURANCE AND TESTING

Functional Testing

Ensures that all features work as intended and meet requirements.

Methods:

- Unit Testing: Tested individual components like CRUD operations and ranking algorithms.
- Integration Testing: Verified seamless interaction between frontend and backend (e.g., Firebase integration).
- Regression Testing: Re-tested functionality after updates to prevent new issues.

Key Results:

- CRUD operations worked correctly, including accurate sorting and ranking.
- Authentication and role-based access performed securely without errors.
- Error handling displayed clear messages for invalid actions.

Usability Testing

Evaluates user experience, accessibility, and ease of use.

Methods:

- Conducted tests with 10 students representing target users.
- Participants performed tasks like adding expenses, viewing charts, and splitting group bills.
- Collected feedback on interface clarity and navigation.

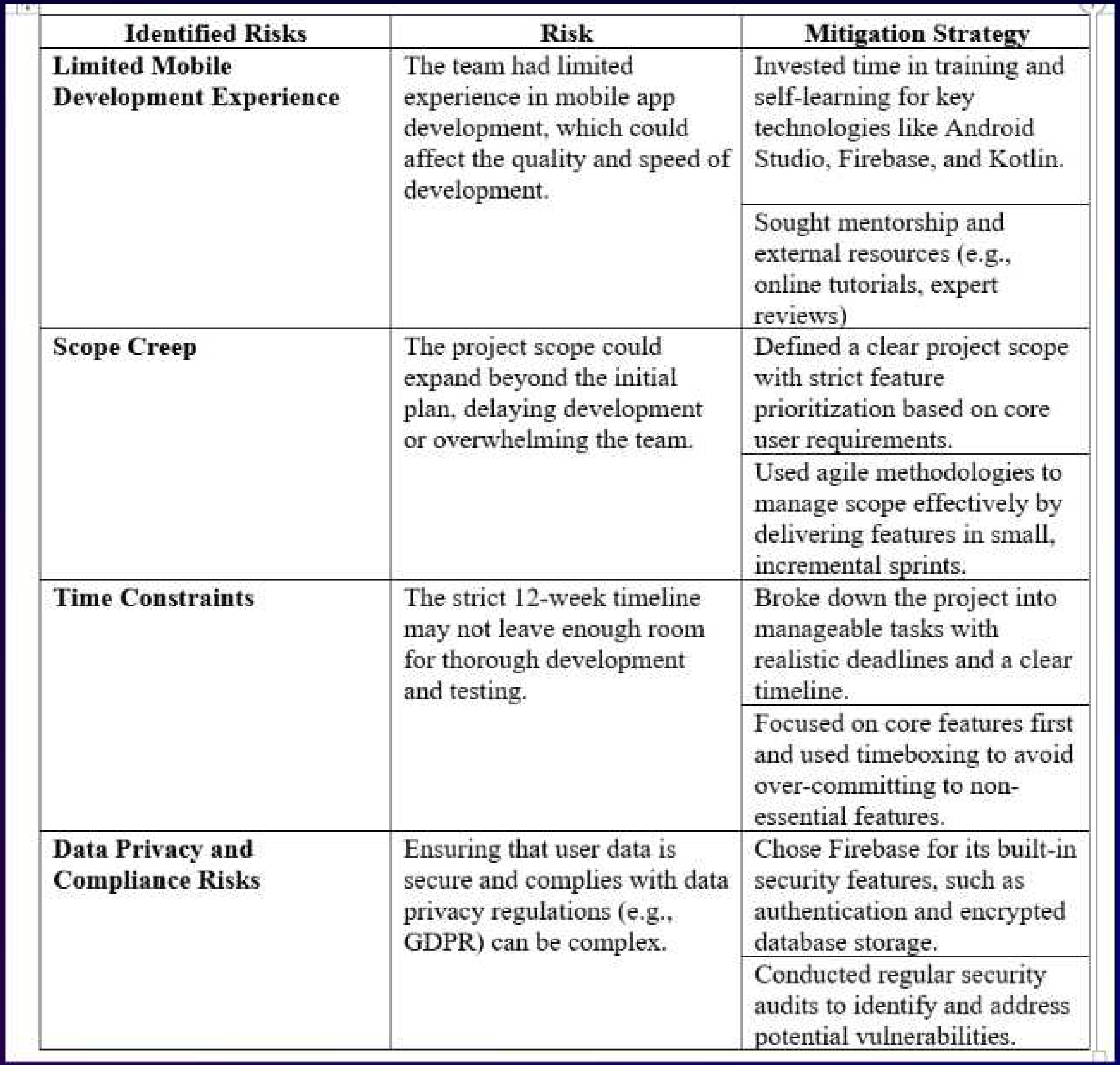
Key Results:

- 95% task completion rate without assistance.
- Positive feedback on intuitive interface and clear design.
- Accessibility improved with screen reader support and keyboard navigation.

Ensuring Reliability and Stability

- Bug-Free Performance: Features like sorting and group expense splitting worked flawlessly.
- Stability: App handled high data volumes and concurrent users effectively.
- Improved User Experience: Users reported the app as easy and enjoyable to use.

Overview of identified risks and the mitigation strategies implemented to address them.



Identified Risks	Risk	Mitigation Strategy
Limited Mobile Development Experience	The team had limited experience in mobile app development, which could affect the quality and speed of development.	Invested time in training and self-learning for key technologies like Android Studio, Firebase, and Kotlin. Sought mentorship and external resources (e.g., online tutorials, expert reviews)
Scope Creep	The project scope could expand beyond the initial plan, delaying development or overwhelming the team.	Defined a clear project scope with strict feature prioritization based on core user requirements. Used agile methodologies to manage scope effectively by delivering features in small, incremental sprints.
Time Constraints	The strict 12-week timeline may not leave enough room for thorough development and testing.	Broke down the project into manageable tasks with realistic deadlines and a clear timeline. Focused on core features first and used timeboxing to avoid over-committing to non-essential features.
Data Privacy and Compliance Risks	Ensuring that user data is secure and complies with data privacy regulations (e.g., GDPR) can be complex.	Chose Firebase for its built-in security features, such as authentication and encrypted database storage. Conducted regular security audits to identify and address potential vulnerabilities.

Tools and strategies used to coordinate tasks, track progress, and collaborate effectively within the team.

Microsoft Project

GitHub

Zoom

Performance Evaluation Criteria

Alignment with User Requirements:

- Feature Coverage: 90% of core features (e.g., expense tracking, budget setting, and notifications) were implemented as per the requirements.
- Task Success Rate: 92% of users successfully completed tasks like adding expenses and setting budgets during testing.

User Satisfaction:

- Survey Scores: Average score of 4.3/5 for overall usability and design.
- Net Promoter Score (NPS): 78, indicating high likelihood of user recommendations.
- Feedback Analysis: 85% of comments were positive, highlighting simplicity and clarity.

App Quality:

- Performance Metrics: Average screen load time of 1.8 seconds, with offline functionality fully operational for 95% of tested cases.
- Error Rates: Only 3% of tasks encountered bugs (e.g., minor UI glitches).
- Crash Rate: Less than 0.5% during testing across both platforms (Android and iOS).

Usability Metrics:

- Time on Task: Average time to add a new expense: 12 seconds; set a budget: 20 seconds.
- Click Efficiency: Average clicks to complete core actions: 3-5 clicks.
- User Complaints: Only 2 recurring issues were reported, primarily related to navigation and were resolved.

Summary of how well the app meets the original user requirements

CRUD Operations on Student Data:

- Requirement: The app should allow users to create, read, update, and delete student records.
- Example: A user can add a student named John Doe with an ID of 101 and a score of 85, update his score later, and delete his record if needed.

Sorting and Searching Students:

- Requirement: Enable sorting and searching of students by various criteria (e.g., name, score, ID).
- Example: An instructor can sort students by their scores to identify top performers or search for a specific student by name.

Ranking Students Based on Scores:

- Requirement: Automatically rank students based on their scores using predefined criteria.
- Example: If two students have the same score, the app ranks them using their IDs, ensuring no conflicts in ranking.

Key takeaways from the project.

Skills Acquired:

- Mobile Development: Gained practical experience in using Android Studio, Firebase, and Kotlin/Java for building a functional and secure mobile app.
- Agile Project Management: Mastered agile techniques, such as sprint planning and task prioritization, which helped meet deadlines and manage the project efficiently.
- UI/UX Design: Enhanced skills in designing user-friendly interfaces, focusing on simplicity and usability through continuous feedback and iteration.

Understanding User-Centered Design:

- Importance of User Testing: Recognized the value of early user involvement in the design process to uncover pain points and improve functionality.
- Iterative Design Process: Realized the need for continuous refinement based on user feedback to ensure the app remains intuitive and accessible.
- Simplicity is Key: Learned that keeping the app design simple and focused improves user engagement and overall satisfaction.

Insights into Mobile Development:

- Real-Time Data Handling: Developed an understanding of how real-time data synchronization works in mobile apps, particularly for tracking dynamic data like expenses.
- Device Compatibility: Learned how to ensure the app works consistently across different devices, handling varying screen sizes and hardware specifications.

Feedback Analysis and Future Improvements

Feedback Analysis

Positive Highlights:

- Intuitive Interface: Users appreciated the clean design and ease of navigation.
- Expense Tracking: Features like quick logging and spending summaries were well-received.
- Group Management: Splitting bills and tracking debts were considered highly useful.

Constructive Criticism:

- Customization Limitations: Users wanted more flexibility to create custom categories.
- Offline Capability: Logging expenses without an internet connection was a frequent request.
- Export Features: A lack of data export options (CSV, PDF) was noted.
- Dark Mode: Many users requested a dark theme for comfort in low-light conditions.

Future Improvements

- Data Export and Import: Allow users to export expenses to CSV or PDF formats and import data from external files.
- Recurring Transactions: Add functionality for setting recurring expenses (e.g., rent, subscriptions).
- Dark Mode: Implement a dark theme to improve user experience in low-light environments.
- AI Insights: Use AI to analyze spending patterns and provide personalized financial advice.
- Multi-Language Support: Expand accessibility by supporting multiple languages.
- Improved Analytics: Add more detailed reports and comparisons to help users track financial goals.

Q&A Preparation

Why did you choose Firebase for authentication and data management?

How did you ensure the app's usability for university students?

What challenges did you face with cross-platform development?

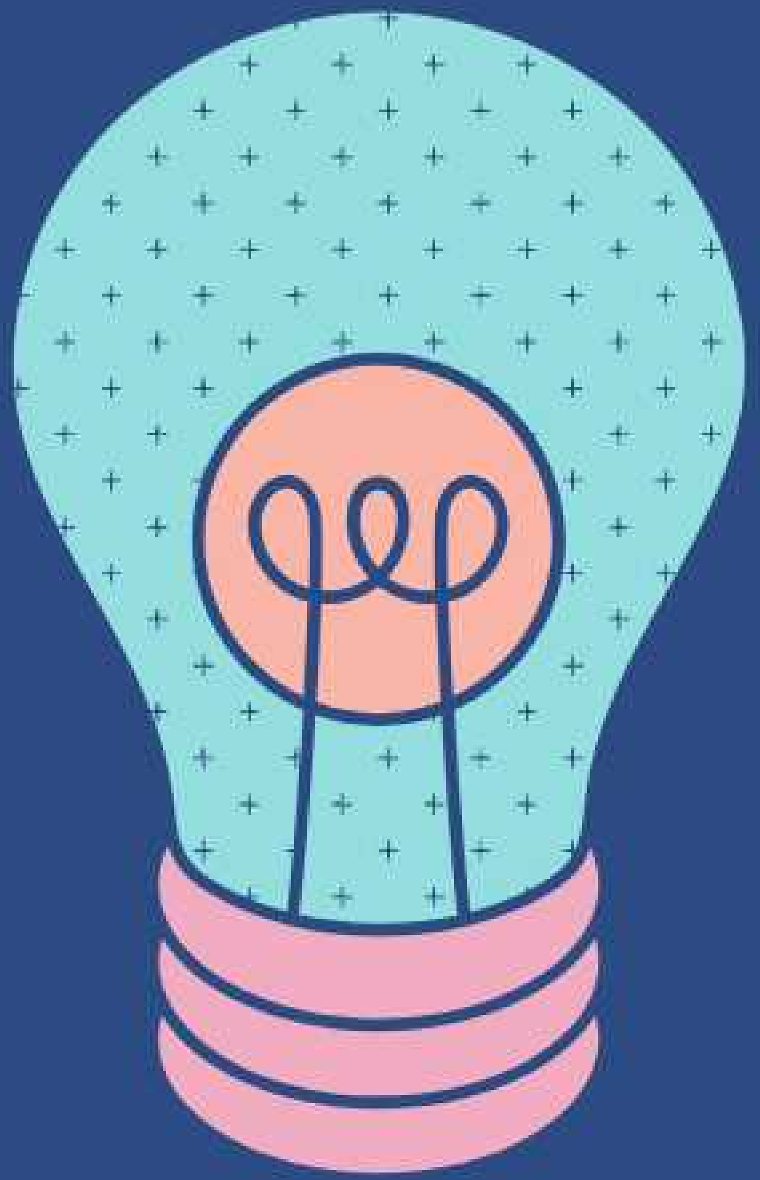
What are the key lessons you learned during this project?

How do you plan to address user feedback and future improvements?

How did you handle the limited development budget and time constraints?

Conclusion

Successfully developed the CampusExpense Manager app within the 12-week timeline.
Implemented key features such as expense tracking, budget alerts, and expense categorization, meeting user requirements.
Addressed user feedback to enhance usability and improve satisfaction.
Overcame technical challenges, gaining valuable skills in mobile development and user-centered design.



Thank you for your time and support throughout this journey. We are grateful for the opportunity to present our work and look forward to your feedback!

Thank
You