

Spendwise

Your Personal Expense Tracker

Haseeb Ahmed - 26077

Fatima Naeem - 26933

Fatima Shahid - 26905



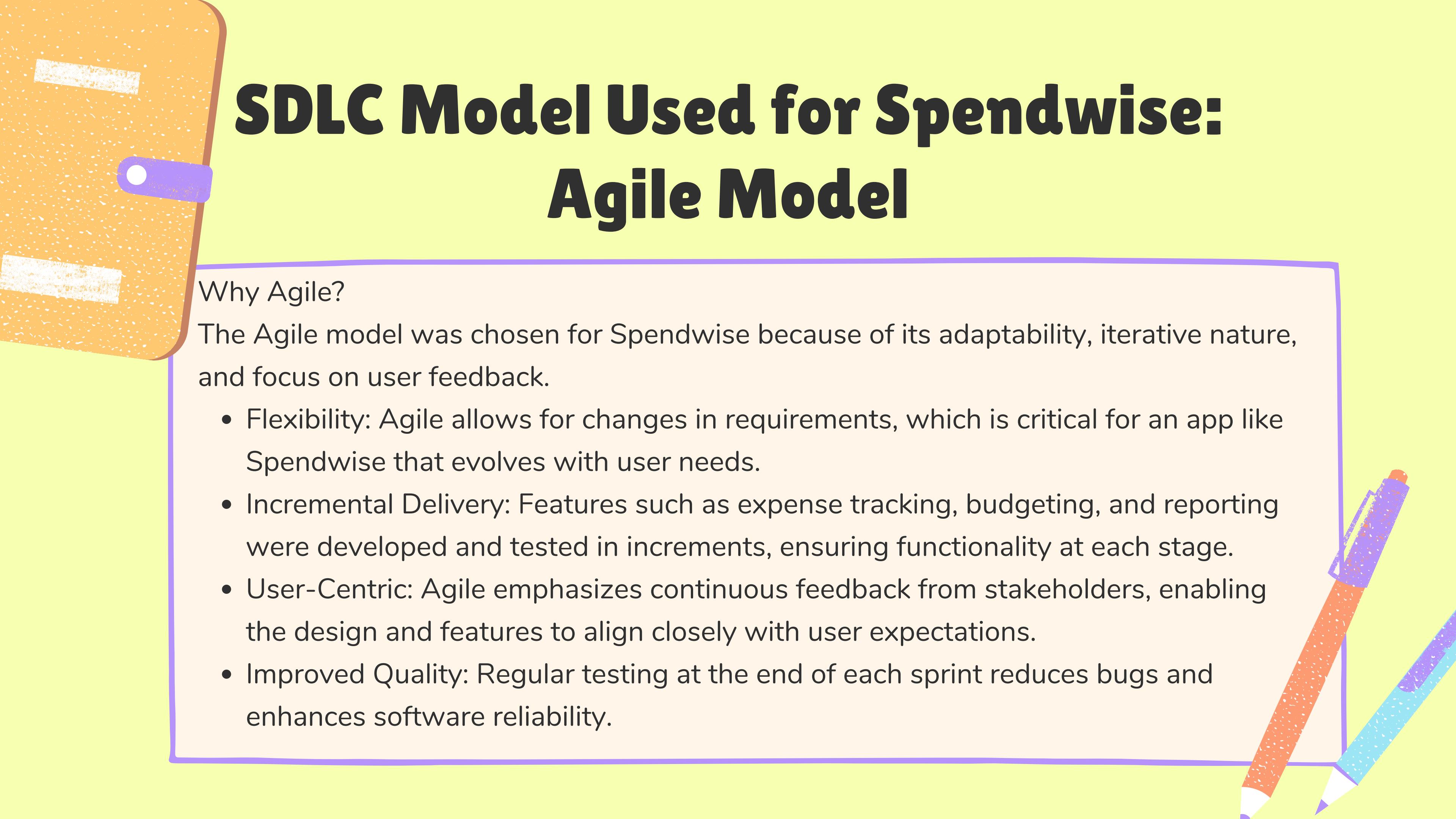
Spendwise is a modern expense tracking application designed to help users take control of their finances. It offers a seamless and user-friendly interface where individuals can:

- Track their daily, weekly, and monthly expenses.
- Categorize transactions for better understanding of spending habits.
- Generate detailed financial reports and visualizations to evaluate and improve budgeting.
- Set spending limits and receive alerts to avoid overspending.



GOALS:

- Better Expense Management
- Budgeting Assistance
- Financial Awareness
- Accessibility and Ease of Use
- Data Security



SDLC Model Used for Spendwise: Agile Model

Why Agile?

The Agile model was chosen for Spendwise because of its adaptability, iterative nature, and focus on user feedback.

- Flexibility: Agile allows for changes in requirements, which is critical for an app like Spendwise that evolves with user needs.
- Incremental Delivery: Features such as expense tracking, budgeting, and reporting were developed and tested in increments, ensuring functionality at each stage.
- User-Centric: Agile emphasizes continuous feedback from stakeholders, enabling the design and features to align closely with user expectations.
- Improved Quality: Regular testing at the end of each sprint reduces bugs and enhances software reliability.

Requirements Analysis

- Functional Requirements
 - Spendwise empowers users to efficiently manage their finances by providing features such as expense tracking, categorization, budgeting with alerts, detailed reporting with visualizations, and secure user profiles with cross-device synchronization.
- Non-Functional Requirements
 - Spendwise ensures a seamless and secure user experience with an intuitive interface, multi-language support, fast performance, robust security measures, and scalable architecture to accommodate diverse user needs.
- Stakeholders
 - Spendwise caters to individuals, small businesses, and freelancers while involving a dedicated development team, product managers, regulatory bodies, and third-party integrators to ensure compliance, functionality, and seamless integration.

User Stories

Expense Tracking

As a user, I want to record my daily expenses so that I can keep track of where my money is going.

Expense Categorization

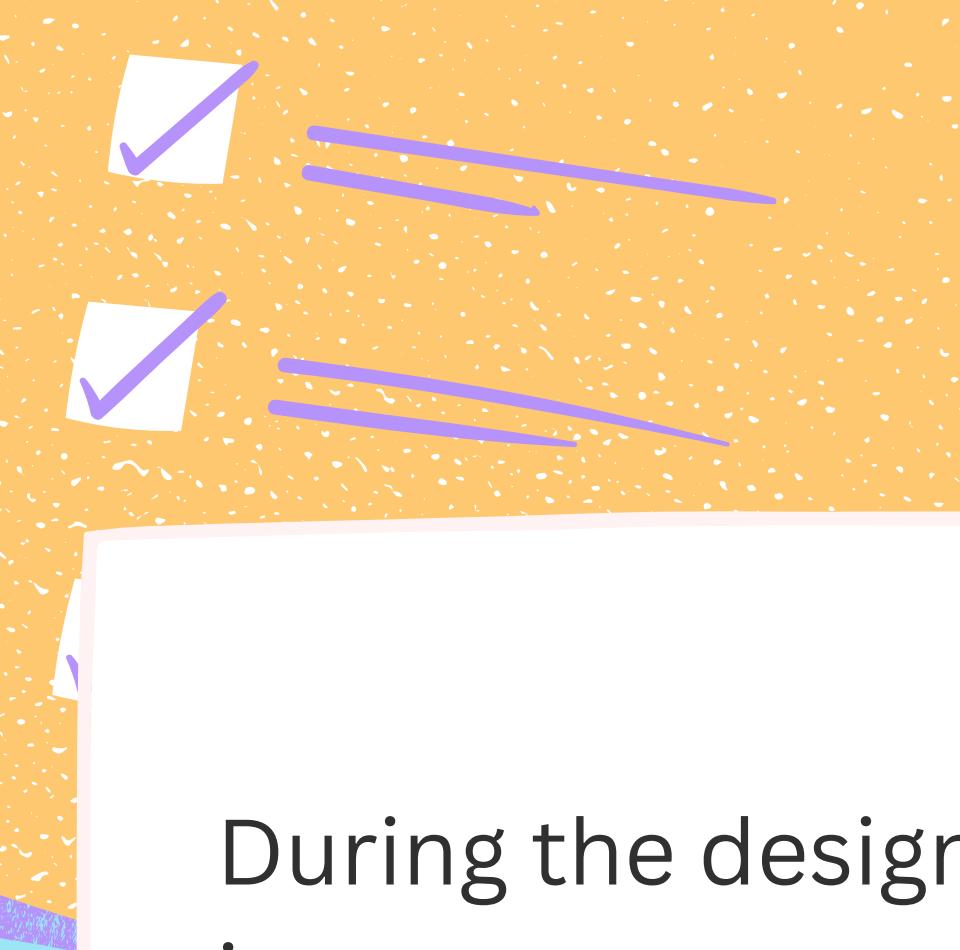
As a small business owner, I want to create custom categories for business expenses so that I can easily separate personal and professional spending.

Budgeting and Alerts

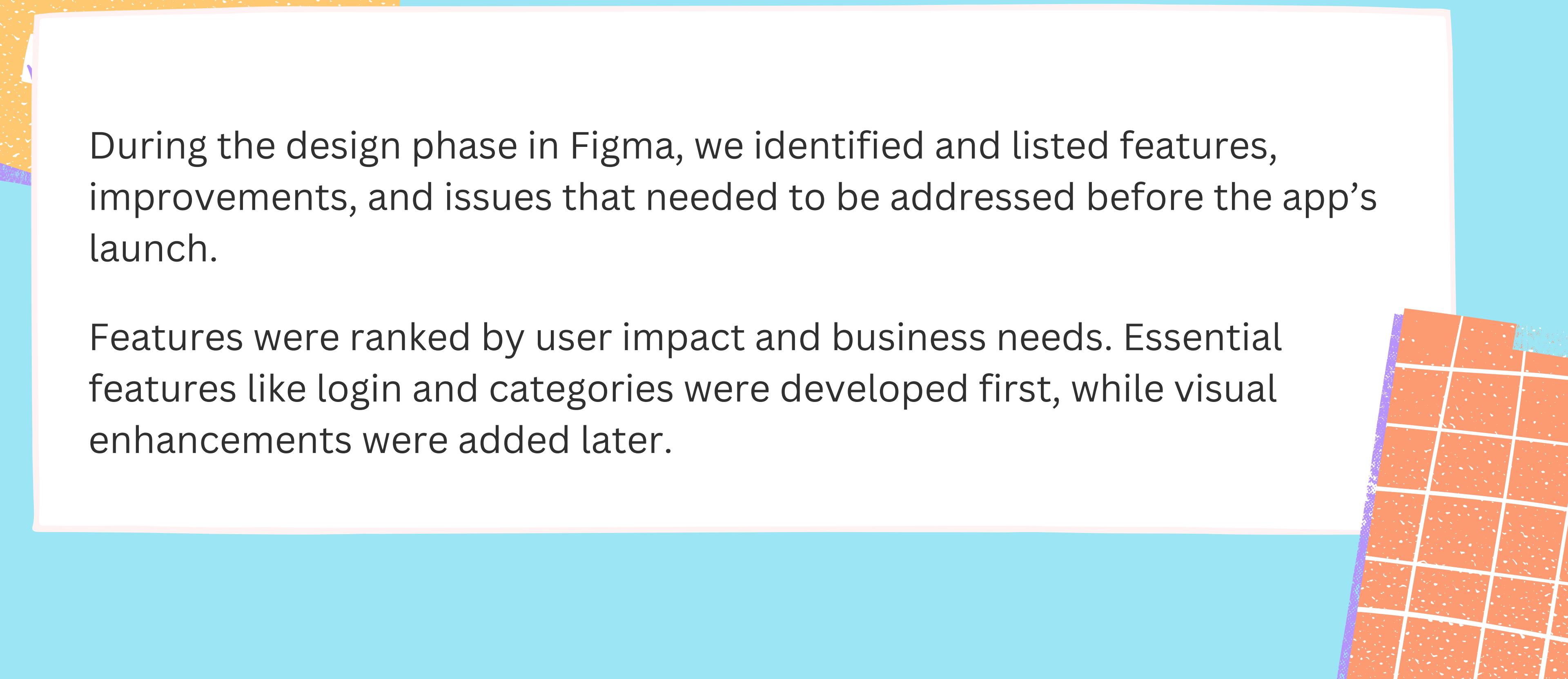
As a user, I want to receive alerts when I am close to exceeding my budget so that I can adjust my spending accordingly.

Report Generation

As a user, I want to view visual reports (e.g., pie charts, bar graphs) of my expenses so that I can quickly analyze my financial health.



Product Backlog



During the design phase in Figma, we identified and listed features, improvements, and issues that needed to be addressed before the app's launch.

Features were ranked by user impact and business needs. Essential features like login and categories were developed first, while visual enhancements were added later.

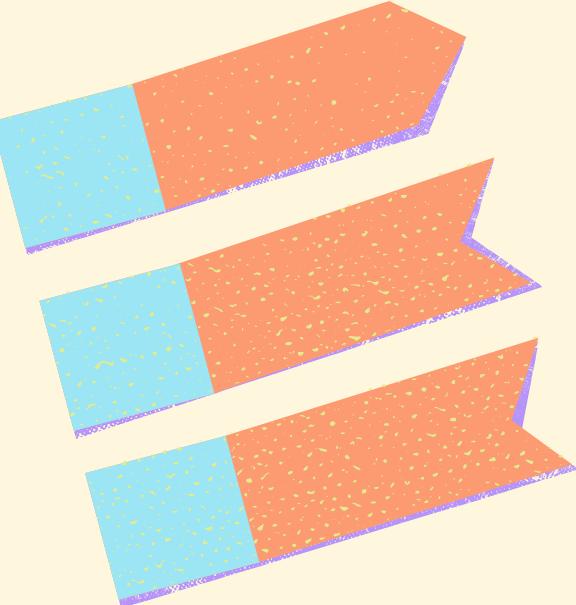
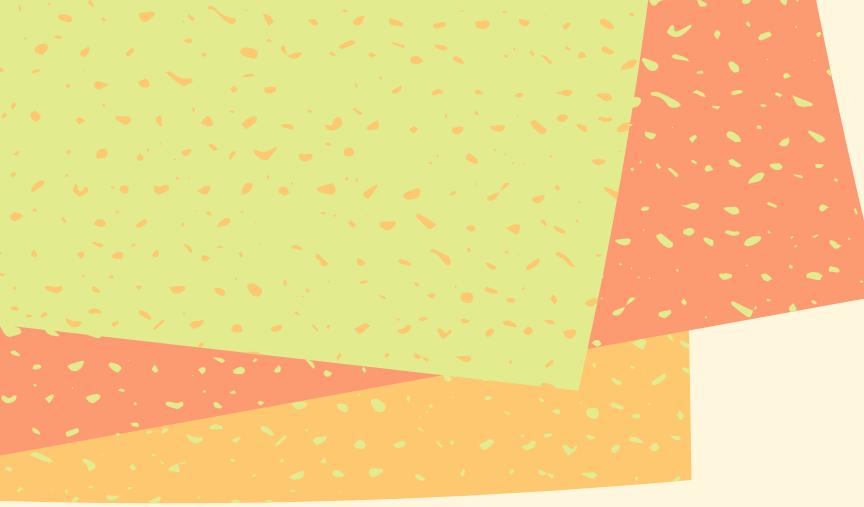
Product Backlog

User Stories	Priority	Improvements
As a user, I want to register an account so that I can securely save and access my expense data.	High	Integrated a user-friendly registration form where user can easily log in with already created social media accounts
As a user, I want to add an expense with a category and notes so that I can track my spending accurately.	High	Designed a clear section where user can easily view existing categories and add it
As a user, I want to view my expenses in a visually appealing list format so that I can manage them easily.	Medium	Utilized listing for the breakdown of expenses
As a user, I want to see a summary of my expenses by category with graphical visualizations so that I can understand my spending patterns.	Medium	Implemented graphs that reflect data dynamically.
As a user, I want to generate a monthly report of my expenses so that I can evaluate my financial health.	Medium	Designed a report layout

Use Cases

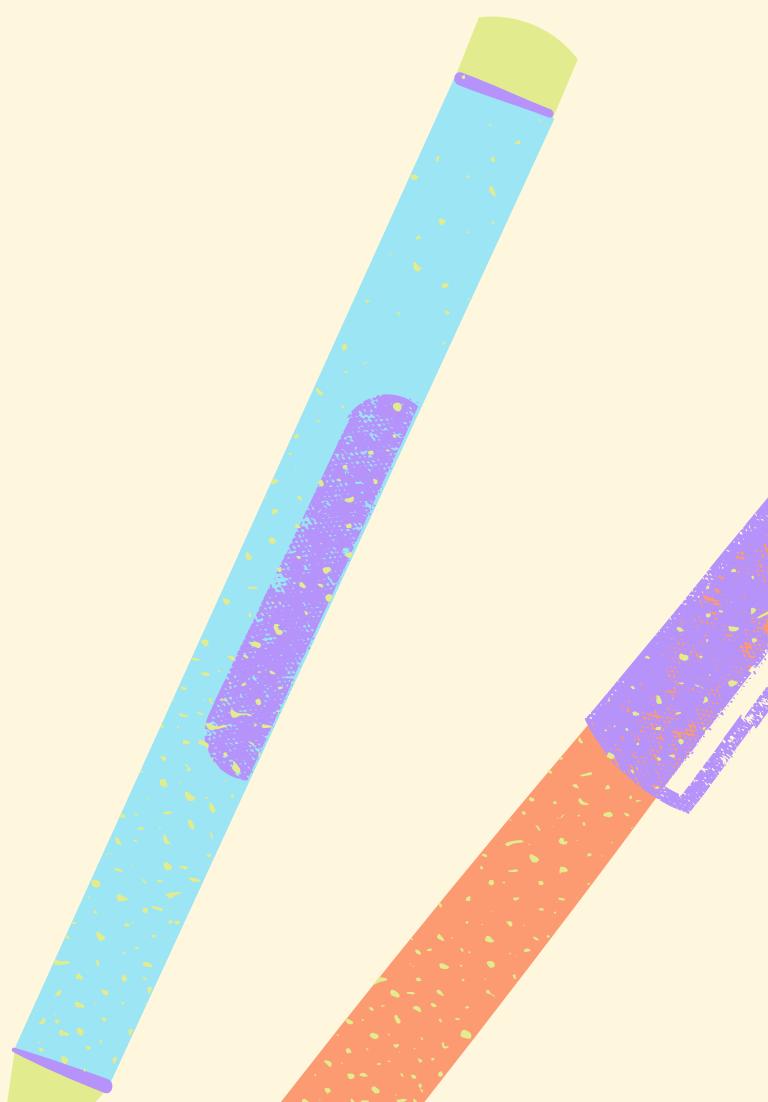
Key Use Cases:

- Add Transaction: Users can enter a new expense into the system by specifying the amount, category, and description.
- Generate Reports: Users can generate detailed reports based on their spending, categories, and time periods.
- View Transaction History: Users can view a list of past transactions and their details
- Set Budget Limits: Users can set spending limits for each category to help manage their finances.
- Category Management: Users can add or remove categories for transactions (e.g., "Food", "Entertainment").
- User Registration/Login: New users can sign up and existing users can log in to the system.



UI/UX Design

Design Principles Followed

- 
- Simplicity: The app focuses on clean, uncluttered interfaces, presenting key information clearly. Example: The dashboard displays essential metrics like budget status and recent transactions.
 - Consistency: Consistent design elements (buttons, fonts, colors) ensure a cohesive user experience. Example: Uniform button styles across all screens.
 - Accessibility: High-contrast text and support for screen readers ensure the app is usable for all, including those with visual impairments.

Architectural Design

Layered Architecture:

The app is structured into three layers:

Frontend (UI): Manages user interactions and displays data.

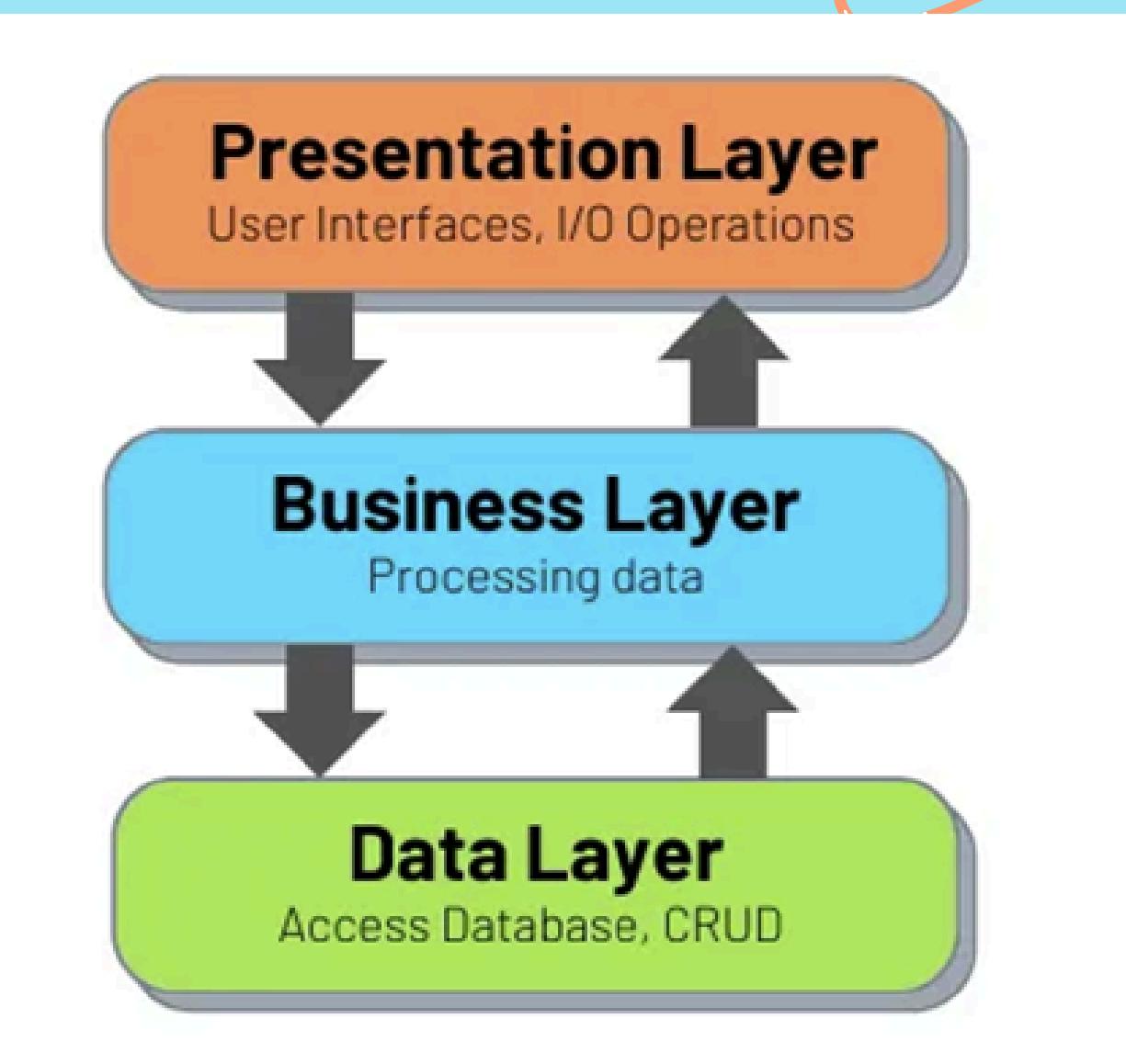
Backend (Logic): Handles business logic and processes requests.

Database: Stores user data and transactions.

2. Component Interaction

Frontend sends requests to Backend.

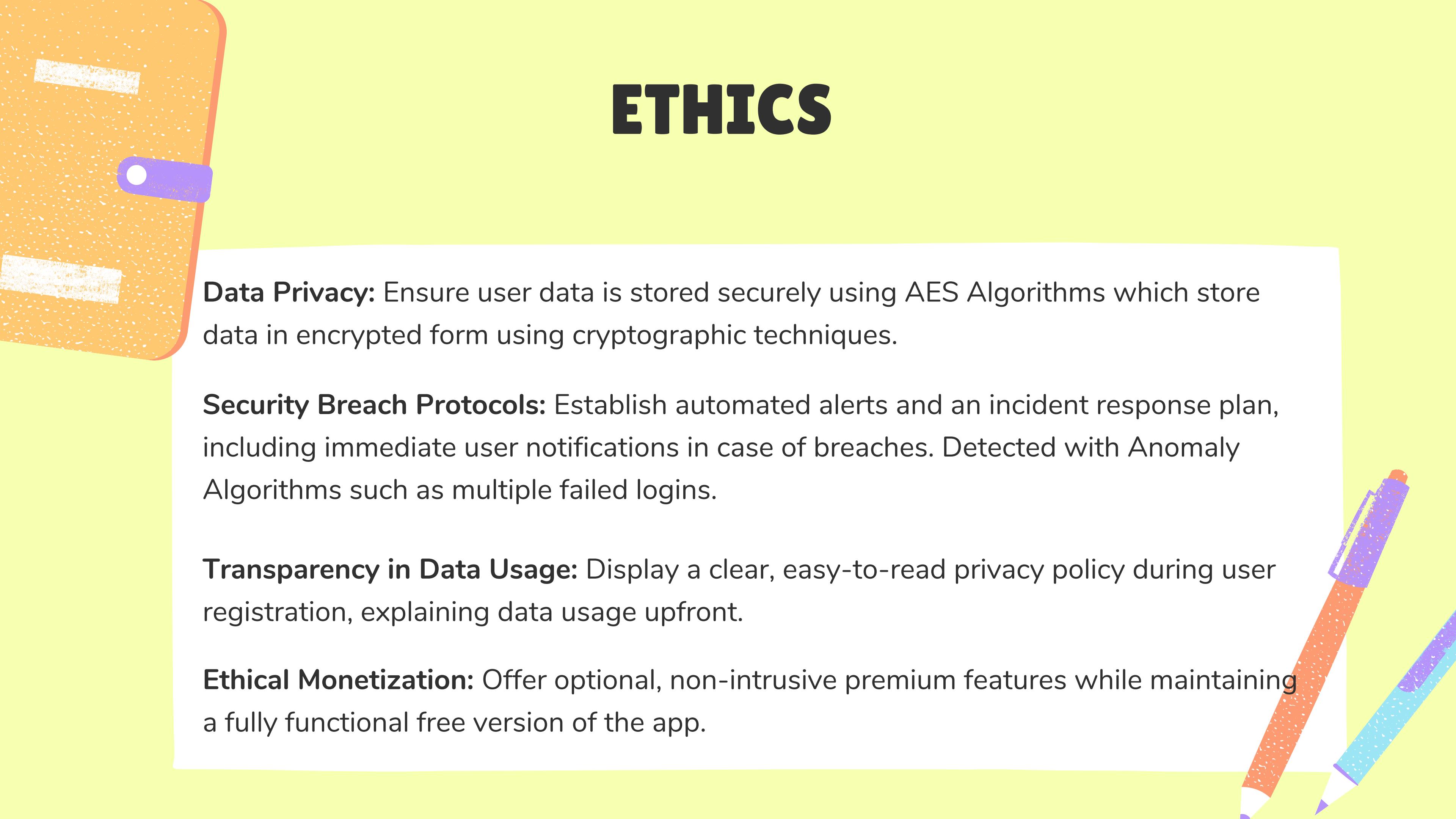
Backend processes the data, communicates with the Database, and returns the result to the Frontend for display.



Software Quality

SpendWise is committed to software quality as a core principle of the Spendwise development process.

- Balance development investments to achieve high quality while controlling resource use and avoiding unnecessary overhead in terms of Quality Costs
- Establish coding practices, enforce regular reviews, and rigorous testing, ensure Spendwise is reliable and user-friendly.
- Employ verification and validation to meet functional and non-functional requirements of Spendwise.
- Adhering to ISO 9126 Standard and making sure they are followed so that Software Quality is not compromised.



ETHICS

Data Privacy: Ensure user data is stored securely using AES Algorithms which store data in encrypted form using cryptographic techniques.

Security Breach Protocols: Establish automated alerts and an incident response plan, including immediate user notifications in case of breaches. Detected with Anomaly Algorithms such as multiple failed logins.

Transparency in Data Usage: Display a clear, easy-to-read privacy policy during user registration, explaining data usage upfront.

Ethical Monetization: Offer optional, non-intrusive premium features while maintaining a fully functional free version of the app.

Pre-Deployment Testing

Pilot Deployment

SpendWise is deployed to a small audience to gather real-world data, usage and feedback through post-testing surveys for final adjustments.

Investor Review

The beta version is shared with investors to ensure alignment with their expectations and business goals.

Compatibility Testing

SpendWise is tested across devices, operating systems, and browsers to ensure consistent performance.

Security Testing

Security checks are performed to identify vulnerabilities and ensure sensitive financial data is protected.

Post-Deployment Testing

Performance Monitoring

Continuous monitoring is conducted to track response times, server stability, and user load after deployment.

User Behavior Analysis

User interactions are analyzed to identify patterns and uncover areas for improvement.

Security

Live security assessments are performed to detect vulnerabilities and ensure continued data protection.

Post-Deployment Surveys

Post-deployment surveys are sent to users to assess satisfaction, usability, and identify potential issues.

Testing Techniques used

Unit Testing	Integration Testing	System Testing	Acceptance Testing
<ul style="list-style-type: none">Purpose: Validate individual components of SpendWise (functions/classes) for correct behavior.Scenario: Test logic for transactions, adding expenses, or performing calculations	<ul style="list-style-type: none">Purpose: Test the interaction between modules or components after integration.Scenario: Validate data between user login and expense tracking modules.	<ul style="list-style-type: none">Purpose: Verify the end-to-end behavior of SpendWise as a whole system in its deployment environment.Scenario: Validate how SpendWise performs when multiple users interact with accounts simultaneously.	<ul style="list-style-type: none">Purpose: Validate if SpendWise meets user requirements and is ready for production deployment.Execute testing with sample users performing real-world usage scenarios like adding expenses or generating reports.

White Box vs Black Box Testing: For SpendWise, White-Box Testing examines internal logic, such as user authentication and expense entry, by analyzing code pathways. Black-Box Testing validates user-facing features like login, expense entry, and report generation without examining the code. White-Box is used after feature implementation, while Black-Box ensures user experience before deployment.

Challenges Faced & Solutions

Requirement Gathering and Analysis

- Challenge: Identifying exact user and business requirements.
- Solution: Conducted stakeholder interviews, user surveys, and requirement workshops to ensure clarity and alignment.

Team Coordination and Communication

- Challenge: Managing cross-functional team members effectively.
- Solution: Adopted Agile methodologies with daily stand-ups and clear task assignments to improve communication.

Cross-Platform Compatibility

- Challenge: Ensuring app compatibility on various devices and OS platforms.

- Solution: Conducted thorough testing across different browsers, devices, and screen sizes.

Scalability & Performance Optimization

- Challenge: Ensuring the application could handle large amounts of concurrent users efficiently.
- Solution: Performed load testing and optimized database queries and server responses.

Future Scope

- Revenue Generation Strategies
 - Premium Subscriptions: Advanced features & AI insights for ad-free experiences.
 - Partnerships with Financial Institutions: Banking integrations for financial services.
 - Micro-Donations: Allow users to donate spare change to partnered charities.
- AI Integration for Enhanced User Experience
 - AI Insights: Predict spending patterns & give budgeting advice.
 - AI Expense Categorization: Automate smart categorization with predictive algorithms.
 - AI Chatbot: Users can interact with a virtual financial assistant.
- Technological Upgrades
 - Cross-Platform Integration: Seamless experiences on wearables & smart home devices.
 - Blockchain Security: Introduce secure, transparent financial tracking solutions.
- Expansion to Other App Stores: Expand SpendWise to Huawei AppGallery, Microsoft Store, Amazon Appstore, and other platforms.

Conclusion

In conclusion, Spendwise empowers users to take control of their finances by offering a simple, intuitive, and efficient way to track expenses and visualize spending habits. With user-friendly features and insightful analytics, Spendwise makes budgeting easy and smart. "Track. Analyze. Save with Spendwise."

