The CampusExpense Manager Mobile Application

Agenda

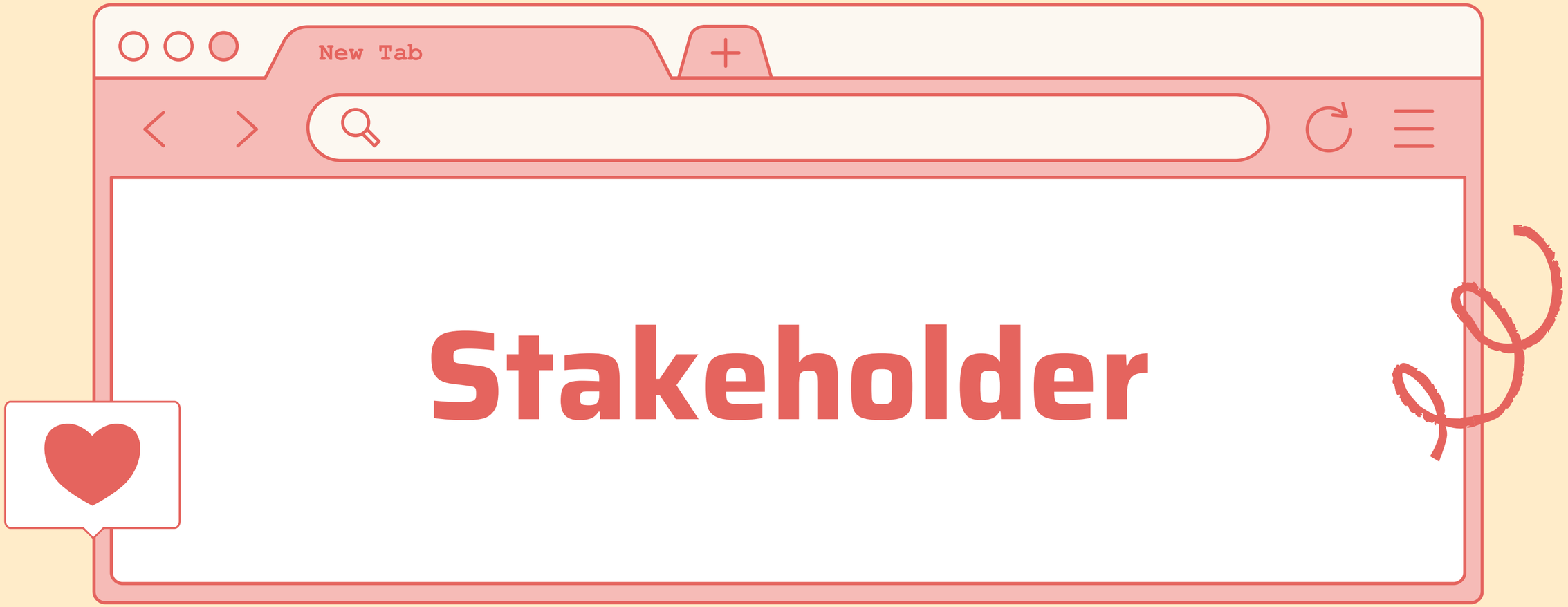
Stakeholders

Functional and Non-functional Requirements

Favourable and Challenges in Project Implementation

Platforms and Essential Features of the Software

Analyze potential risks and risks management



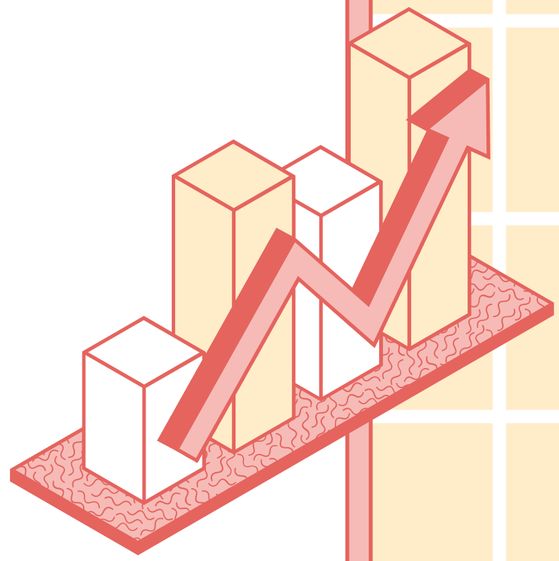
Stakeholder

Quay lại Trang Mục lục

1.Student

2.BudgetWise Solutions Team

3.University Administrators



Students

Role: Primary users of the CampusExpense Manager app.

Wants:

- Easily track spending and manage budgets.
- Get insights and analysis on their financial habits.
- User-friendly, intuitive, and easy to use.
- Get reminders when they are close to their budget.

BudgetWise Solutions Team

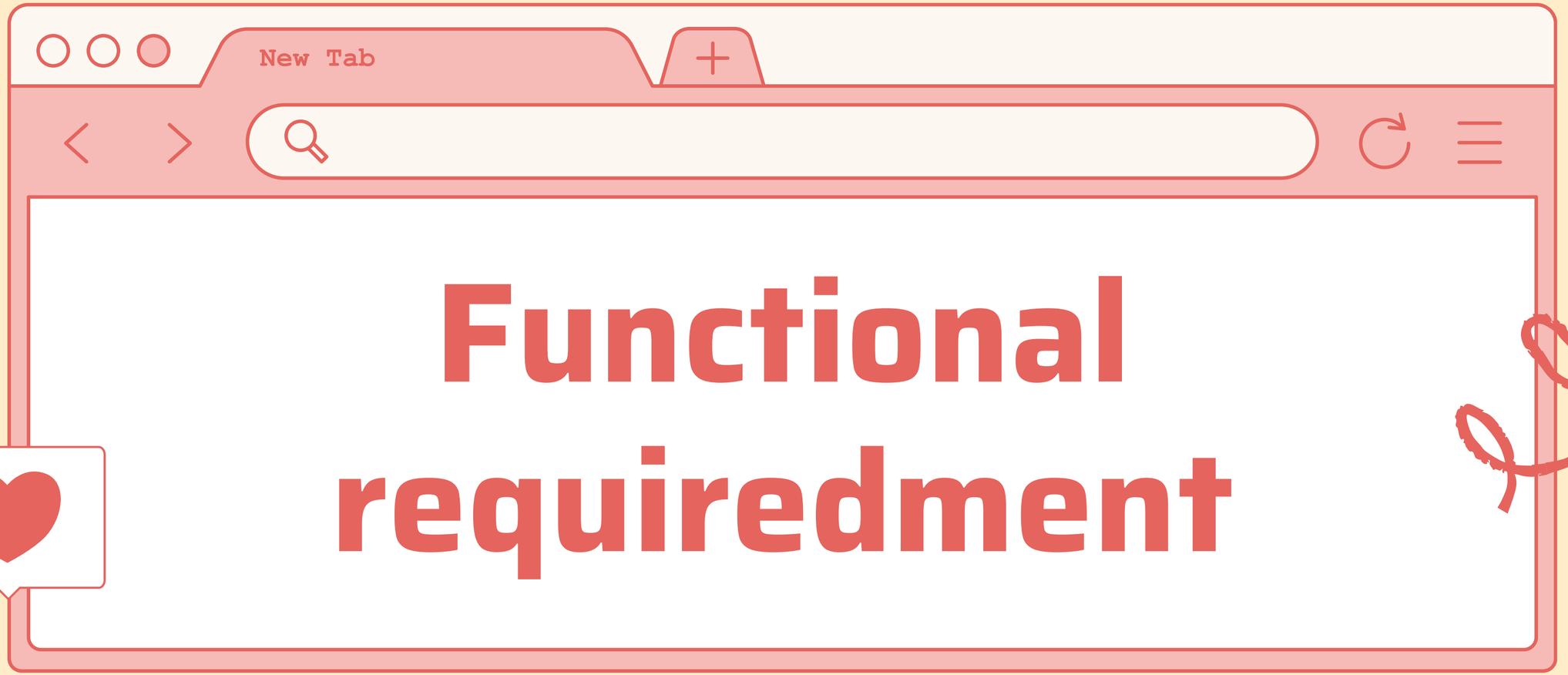
Role: Development team, including designers, programmers and testers.
Desired:

- Clear functional and non-functional requirements of the application.
- Have enough time and resources to develop and maintain the application.
- Get feedback from users to improve the product.
- Full technical support and documentation during development.

University Administrators

Role: University administrator.
Desire:

- Encourage students to use financial management tools to develop responsible financial habits.
- Provide information about the app to support students in managing their spending.
- Organize workshops or events to introduce the app to students.

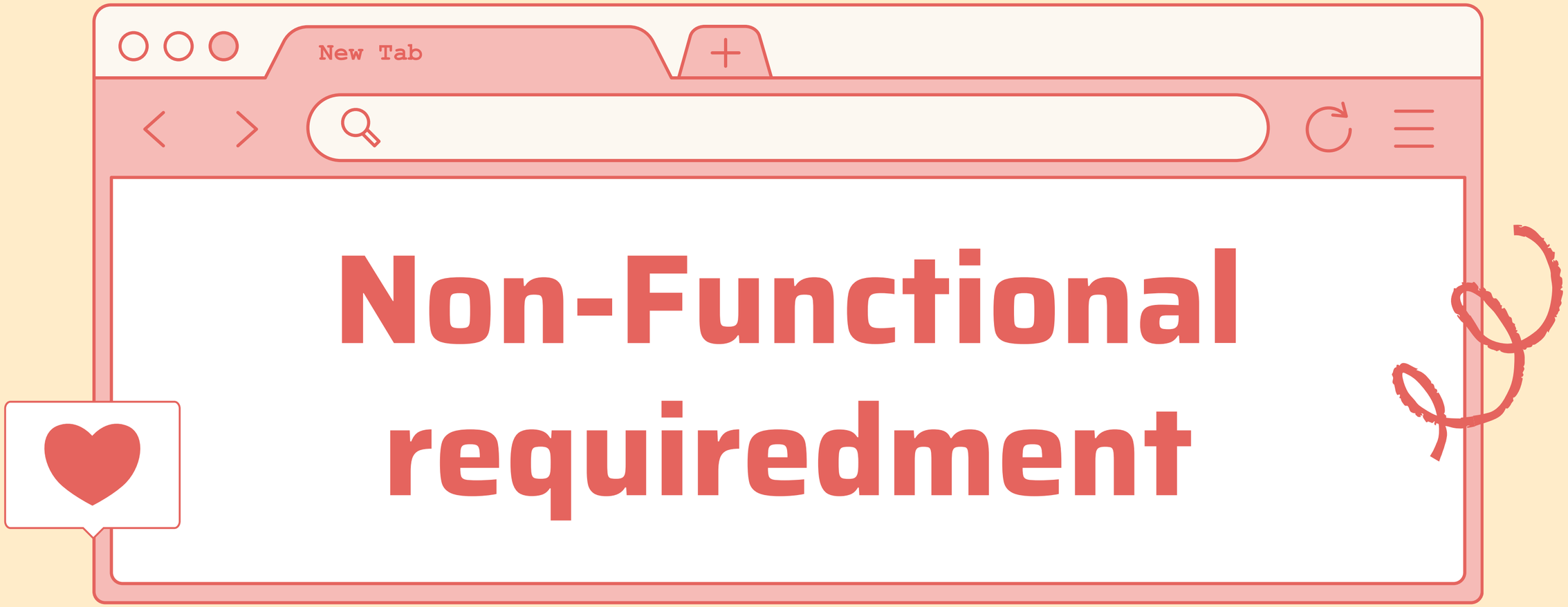


Functional requiredment

Quay lại Trang Mục lục

Functional requiredment

1. User Registration and Authentication
Users can create accounts with a secure username and password.
Users can log in to access and manage their expense data.
Password recovery options should be available.
2. Expense Tracking
Users can add, edit, and delete expense entries.
Each expense entry must include:
Description
Date
Amount
Category (e.g., rent, groceries, transportation)
Users can categorize expenses for better organization.
3. Budget Setting
Users can set monthly budgets for various categories (e.g., food, entertainment, education).
Users can adjust budget amounts and receive notifications for changes.
4. Expense Overview
Provide a summary of monthly expenses, including:
Total spending
Remaining budget
Breakdown by category (visual representation such as charts)
Users can view and analyze expense trends over time.
5. Recurring Expenses
Users can set up recurring expenses with specified start and end dates.
Automatically add recurring expenses to the user's monthly budget.
6. Expense Reports
Users can generate detailed reports for specific time periods (e.g., monthly, annually).
Reports should include a breakdown of expenses by category and visual analytics.
7. Expense Notifications
Users receive notifications when they approach or exceed budget limits for specific categories.
Reminder notifications for upcoming recurring expenses.
8. Feedback and Support
Include a feedback form for users to report issues or provide suggestions.
Provide an FAQ section for common questions.

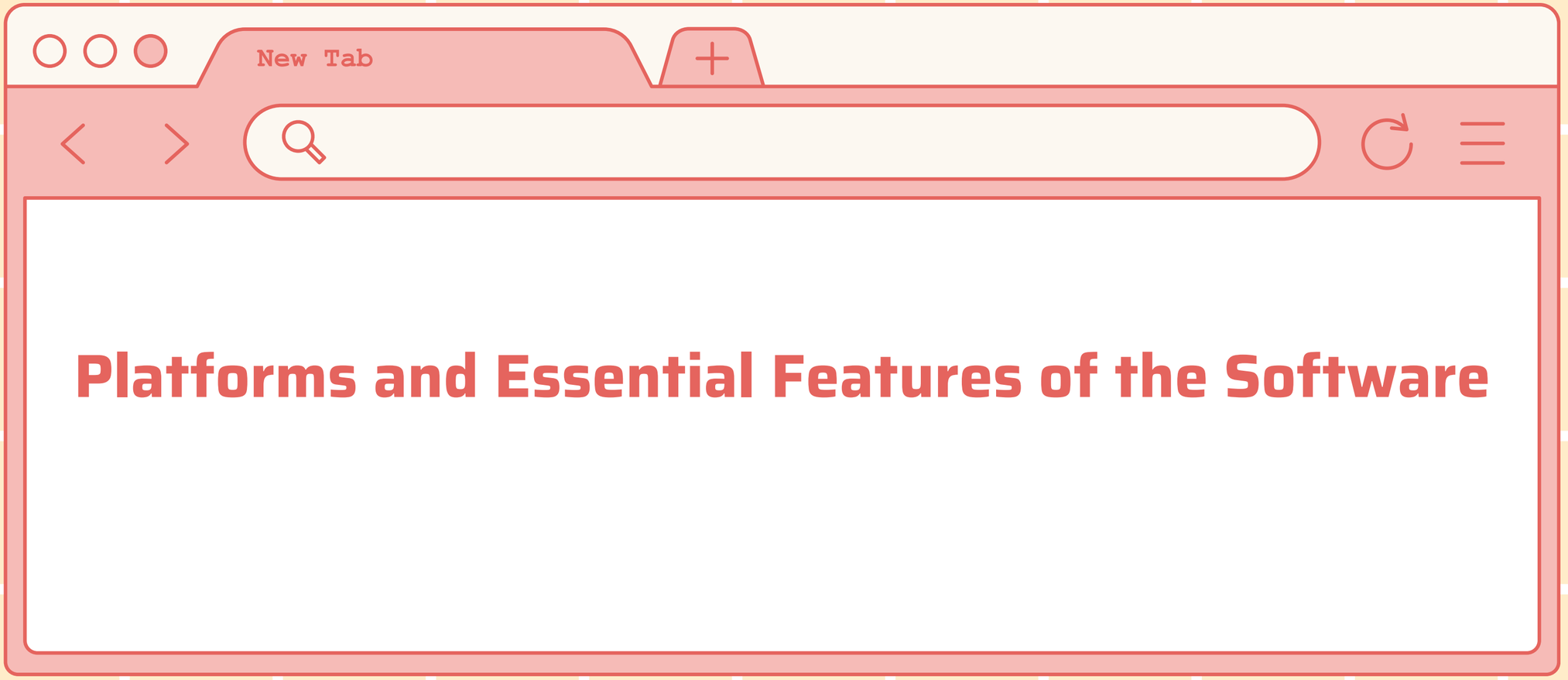


Non-Functional requiredment

Quay lại Trang Mục lục

Non-Functional requiredment

1. Performance
   - The app should load quickly and provide a smooth user experience, even with a large database of expenses.
   - Response time for user actions (e.g., adding an expense) should be minimal.
2. User-Friendly Interface
   - The user interface should be intuitive and visually appealing.
   - Clear labels and navigation to enhance user experience.
   - Support for accessibility features (e.g., screen readers, high-contrast modes).
3. Platform Compatibility
   - The app must be compatible with both Android and iOS platforms.
   - Ensure consistent functionality and design across different devices and screen sizes.
4. Data Security
   - User data must be securely stored with encryption.
   - Implement secure authentication protocols to protect user accounts.
   - Follow best practices for data privacy and compliance with regulations (e.g., GDPR).
5. Scalability
   - The application should be designed to handle a growing user base and increased data volume without performance degradation.
6. Maintainability
   - Code should be modular and well-documented to facilitate future updates and maintenance.
   - Regular updates should be planned for bug fixes and new features based on user feedback.
7. Reliability
   - The app should have high availability and minimal downtime.
   - Implement backup and recovery processes to prevent data loss.
8. Feedback and Monitoring
   - Include analytics to monitor user engagement and app performance.
   - A system for tracking and addressing user feedback promptly.



Platforms and Essential Features of the Software

Platforms of the Software

The app should be developed for both Android and iOS platforms, which may require additional development effort and testing.

Android

Development should target popular Android versions

IOS

The application must be developed for iOS devices (iPhone, iPad, etc.)

User Interface

Responsive Design: Ensure the application works well on various screen sizes.

Database

Firebase/SQLite: Used for storing user data and expenses.

Language

- Java/Kotlin: For Android development.
- Swift: For iOS development.
- React Native/Flutter: Considered for cross-platform development.

Essential Features of the Software

USER REGISTRATION AND AUTHENTICATION

• Account Creation: Allow users to create an account using a username and password.
• Secure Authentication: Ensure user information is protected during login.

Expense Management

- Add, Edit, and Categorize
- Expenses: Enable users to add and edit expenses with categories such as rent, groceries, and transportation.
- Expense Details: Each entry should include a description, date, amount, and category

Budget Setting

- Monthly Budgets: Allow users to set budgets for different expense categories.
- Adjustable Budgets: Users should be able to modify budget amounts as needed.

Expense Overview

- Monthly Summary: Provide information on total spending, remaining budget, and breakdown by category.
- Expense Trends: Users can analyze spending trends over time.

Recurring Expenses

- Add Recurring Expenses: Allow users to set up recurring expenses (e.g., monthly rent).
- Automatic Addition: The app should automatically include thes expenses in the monthly budget.

Expense Reports

- Detailed Expense Reports: Enable users to generate reports for specific time periods (monthly, annually).
- Expense Notifications
- Reminders: Send notifications when users approach or exceed budget limits for specific categories.

Essential Features of the Software

User-Friendly Interface

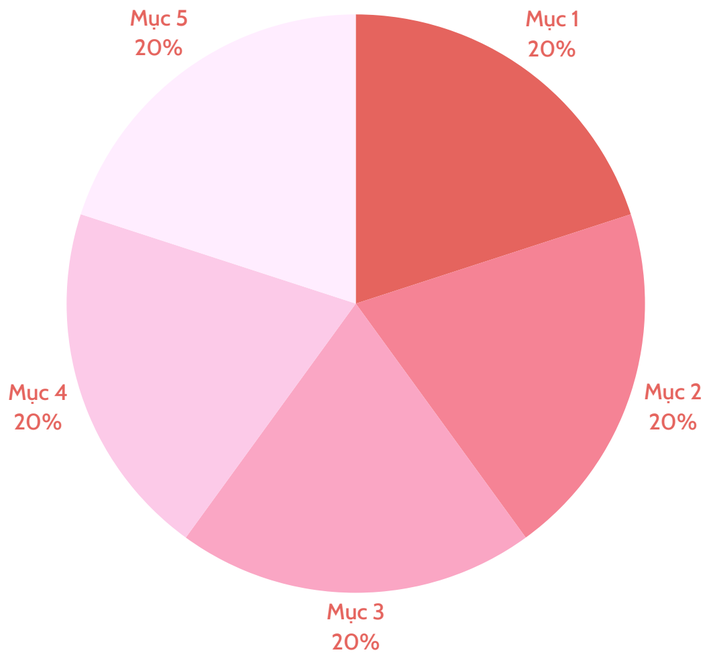
- Intuitive Design: The interface should be easy to use and navigate.

Data Security

- Secure Storage: User data must be encrypted and securely stored.

Feedback and Support

- Feedback Form: Provide a tool for users to submit feedback or report issues.



Strategies to manage risks

Potential Risks

Technical Risks

- Unfamiliar Technology: The development team may lack experience with certain technologies or frameworks.
- Integration Issues: Challenges in integrating various components or third-party services.

Schedule Risks

- Limited Expertise: Junior developers may face challenges that require more experienced guidance.
- Availability of Team Members: Team members may have conflicting commitments, leading to delays.

Resource Risks

- Tight Timeline: The 12-week development timeframe may not be sufficient for all required features.
- Scope Creep: Additional features or changes in project scope can extend timelines.

Budget Risks

- Cost Overruns: Unexpected expenses may arise, putting pressure on the budget.
- Inadequate Funding: Limited budget may restrict necessary resources or tools.

Data Security Risks

- Data Breaches: Vulnerabilities could expose sensitive user data.
- Non-compliance: Failure to adhere to data privacy regulations can lead to legal issues.

User Acceptance Risks

- Low User Adoption: The app may not meet user needs or expectations, leading to poor adoption.
- Negative Feedback: Users may report issues that could affect the app's reputation.

Monitoring and Review

Continuously monitor risks throughout the project lifecycle. Hold regular meetings to review the risk management plan and make adjustments as necessary.

Communication

Maintain transparent communication among team members and stakeholders about risks and mitigation strategies. This fosters a proactive culture where everyone is aware of potential issues.

Contingency Planning

Develop contingency plans for high-priority risks. This includes having backup strategies or resources ready to deploy if a risk materializes.

Documentation

Keep thorough documentation of identified risks, mitigation strategies, and outcomes. This will help in future projects and provide a reference for lessons learned

Thank You!!!

See You No Again