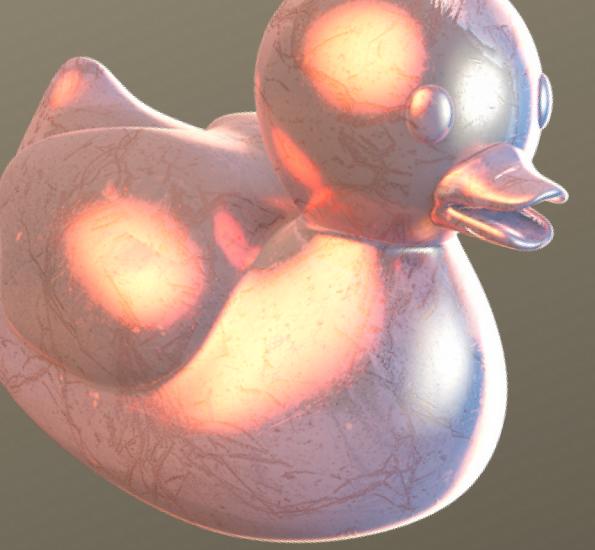
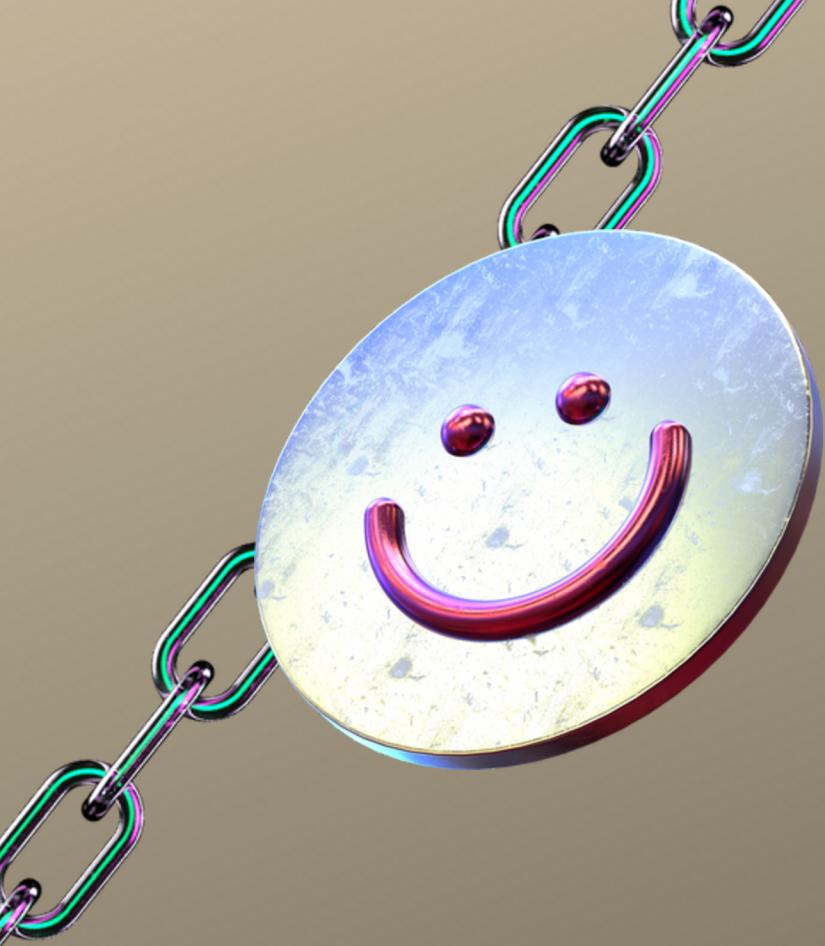
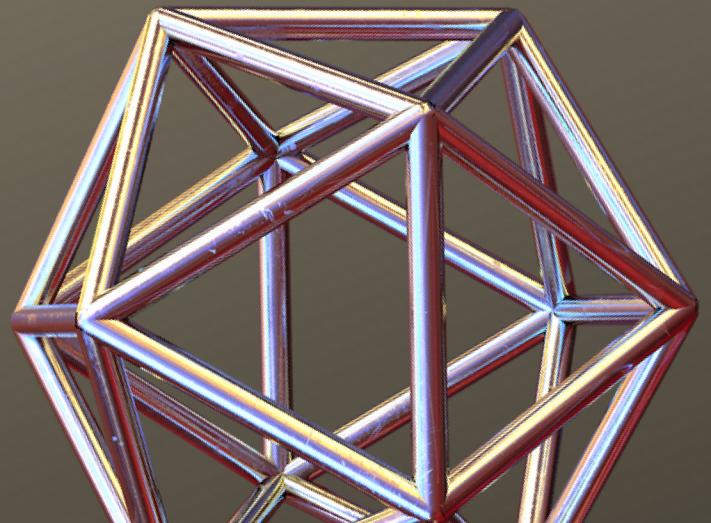


CAMPUS EXPENSE MANAGER





Thành Phong



Viết Tiến



Quang Minh



Khánh Huy



Mạnh Đức

Welcome to my Group

Presentation

Meeting Agenda

01

CampusExpense
Manager

02

What the
Parties Want
from the
System

03

Estimated
budget

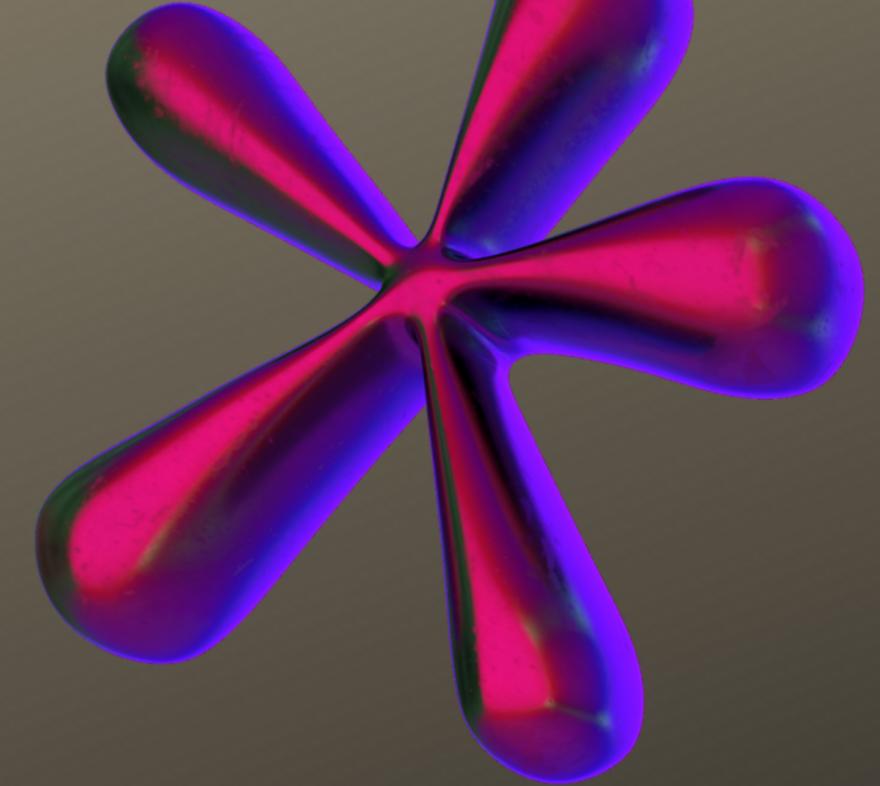
04

Difficulties of
CampusExpense
Manager Projectao

05

Functional
Requirements or
Non-functional
requirements

PROJECT INTRODUCTION AND GOALS



The CampusExpense Manager project is a mobile application specifically designed for university students to help them track and manage their expenses effectively. The application will assist students in controlling their personal budget through features such as entering expenses, categorizing expenses, setting budget alerts, and providing detailed reports on their financial situation. The project aims to provide a simple, easy-to-use tool for students to manage their personal finances in a smarter and more economical way.

IDENTIFY STAKEHOLDERS



Investor



Primary users
(Students)



Project Manager



Development Team

STAKEHOLDERS

1

Investor: This is the party that provides funding to develop the application. The investor can be the university management or an organization affiliated with the school. They want the application to meet the needs of students, contribute to improving reputation and support students in managing their finances effectively.

2

Primary users (Students): Students are the main users of the application, including students living in and outside the dormitory. They expect an easy-to-use application that helps track and manage personal expenses effectively without too many complicated operations.

3

Project Manager: The project manager is responsible for overseeing and ensuring that the project is completed on schedule, meets the budget and meets the requirements. They also link the teams and resolve issues that arise during the development process.

4

Development Team: Includes programmers, designers, and testers. They are responsible for building, developing, testing, and deploying the application. Since the team has limited experience in mobile development, they will face some technical challenges in the project.

WHAT THE PARTIES WANT FROM THE SYSTEM





Investor:
Role: Provide funding to implement and complete the project.
Want: Ensure that the application meets the needs of students and contributes to enhancing the image of the school. They also expect the application to be scalable or create long-term value such as enhancing the connection between students and the school.

Customer (Student):
Role: End users and the main target audience for the application.
Want: Need a simple, easy-to-use tool that meets the personal financial management needs of students. Essential features include expense management, statistical reporting, budget alerts and cost classification.

Project Manager:
Role: Oversee the entire project implementation process, ensuring completion on schedule and within budget.
Want: Define clear goals, allocate resources appropriately and promptly resolve technical or non-technical issues.

University Management:
Role: To be the entity that can deploy and promote the application within the student community.
Aspiration: To ensure that the application not only helps students but also strengthens the image of the university as an organization that supports students in developing personal financial management skills.

Estimated budget



CAMPUS EXPENSE MANAGER PROJECT

BUDGET

- + The total projected budget for the CampusExpense Manager project is [specific amount], allocated as follows: personnel costs ([x%]), equipment and software costs ([x%]), testing and security costs ([x%]), promotion and deployment costs ([x%]), and a contingency reserve ([x%]). Budget monitoring and adjustments will be made at each stage to maintain efficiency and quality while staying within the projected budget.



BUDGET ALLOCATION

1

Human Resources:

Includes salaries for developers, UI/UX designers, testers, and project managers. Ensure that the right team is hired and trained to develop high-quality applications.

2

Equipment and Software:

Costs for computers, development software, and supporting tools such as IDEs, source code management systems, and design software.

3

Testing and Security:

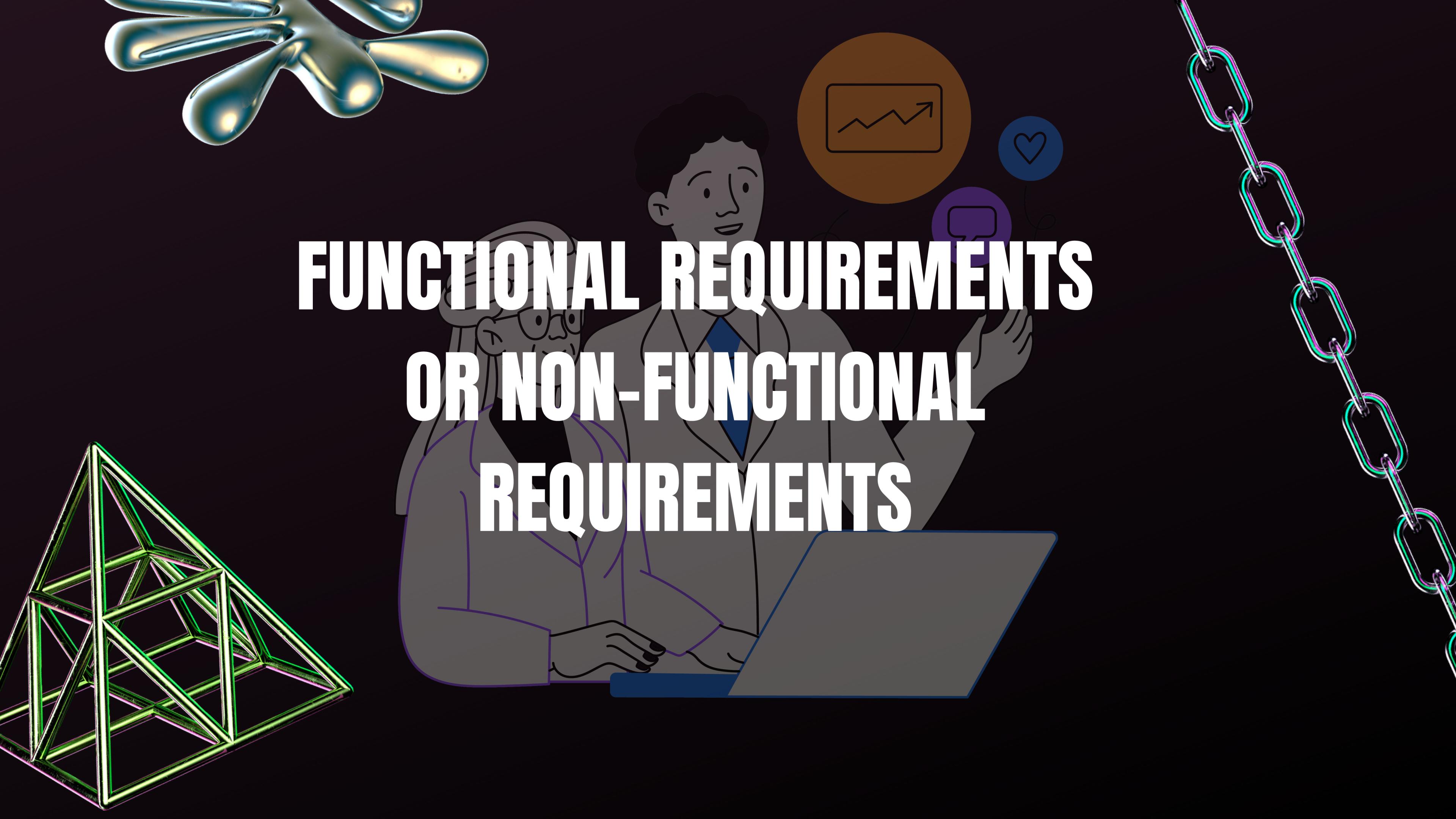
Includes costs for quality assurance (QA) testing, security testing, and possibly outsourcing to ensure the application is secure and stable.

4

Marketing and Promotion:

Budget for marketing activities within the student community, including online advertising, launch events, and media campaigns

FUNCTIONAL REQUIREMENTS OR NON-FUNCTIONAL REQUIREMENTS





PROJECT FUNCTIONS

14

USER REGISTRATION AND AUTHENTICATION

- Users should be able to create new accounts with usernames and passwords.
- The authentication process should be secure, allowing users to log in and access their spending data.

EXPENSE TRACKING

- Users should be able to add, edit, and categorize expenses (e.g. rent, food, transportation).
- Each expense item should include a description, date, amount, and category.

PROJECT FUNCTIONS

BUDGET SETTING

- Set up personal budgets: Users can create budgets for each expense category, helping to control finances more effectively.
- Budget over-warning: The application will send notifications to users when they are close to or exceed the set budget, helping them to adjust their spending in time.



EXPENSE OVERVIEW

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



PROJECT FUNCTIONS

Recurring Spending

- Users should be able to 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.

Spending Reports

- Users should be able to generate detailed spending reports for specific time periods (e.g. monthly, yearly).
- The reports should include a breakdown of spending by category.

Spending Notifications

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



NON-FUNCTIONAL REQUIREMENTS



EASY-TO-USE INTERFACE:

- Friendly design: The application should have an intuitive and easy-to-navigate user interface (UI), helping users quickly get acquainted and use it effectively.
- User experience (UX): Ensure a smooth and comfortable user experience, from registration and login to recording expenses and viewing reports.

SECURITY

- Protection of personal data: User information, including financial and personal information, should be encrypted and stored securely to protect against cyber attacks.
- Secure authentication: Provide multi-factor authentication (2FA) to enhance user account security, minimizing the risk of unauthorized access.

NON-FUNCTIONAL REQUIREMENTS



HIGH PERFORMANCE:

- Page loading speed: The application needs to ensure fast page loading speed, minimizing user waiting time when accessing features.
- Concurrent processing capability: Support multiple users to access and use the application at the same time without reducing performance, ensuring smooth operation even when there is a large amount of traffic.

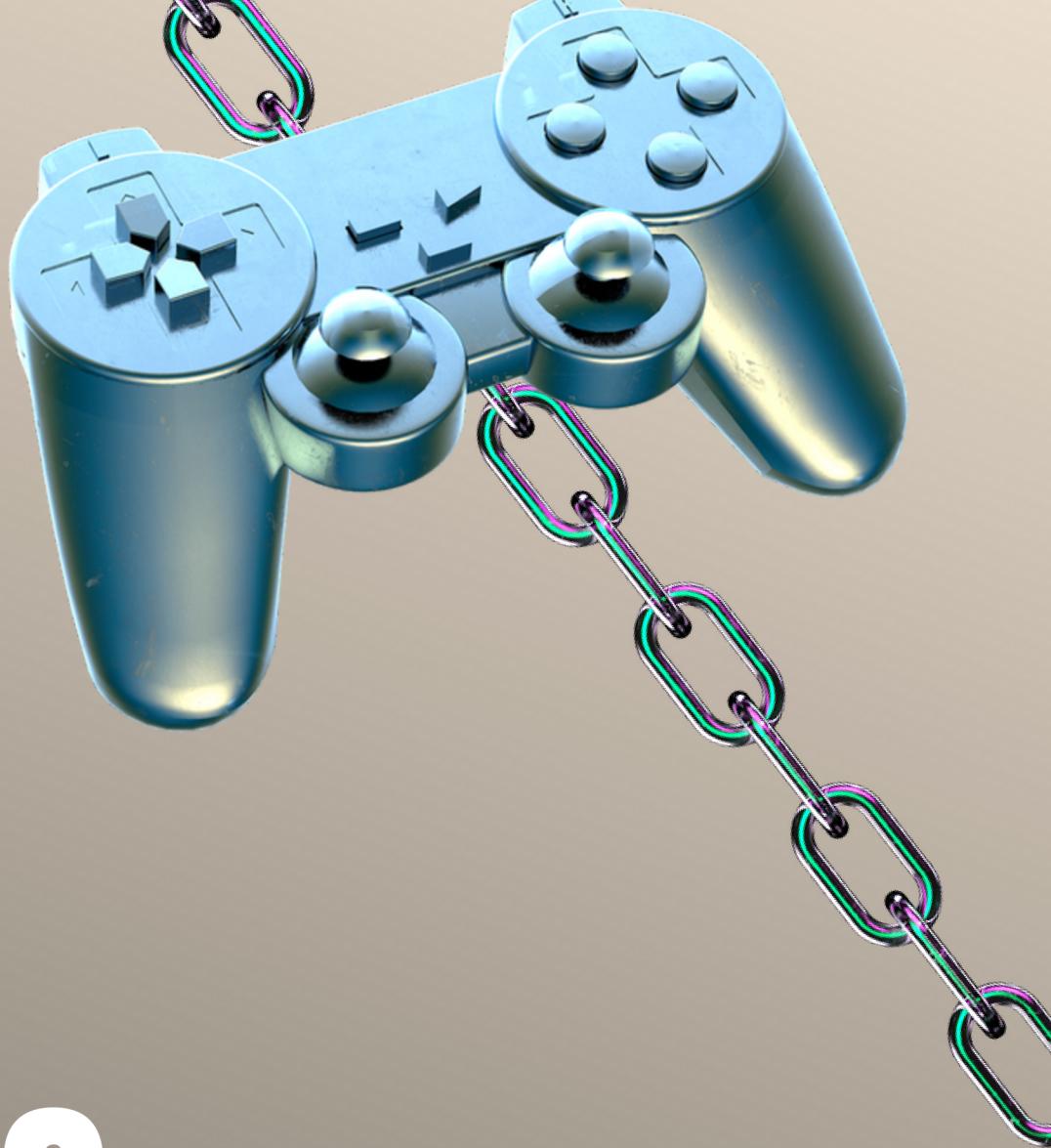
COMPATIBILITY

- Multi-platform: The application must operate effectively on multiple devices and operating systems (Android and iOS) to serve a large number of students.
- Compatible with the latest operating system versions: Ensure that the application can run on new operating system versions, as well as mobile devices with different configurations.

MAINTENANCE AND SUPPORT

- Ease of maintenance: The source code structure should be clear and easy to understand to ensure that future maintenance, upgrades and feature extensions can be done quickly.
- Customer support: Provide an online support channel for users to resolve issues or questions related to the use of the application.

DIFFICULTIES OF CAMPUS EXPENSE MANAGER PROJECTAO



DIFFICULTIES ENCOUNTERED IN THE PROJECT

Lack of experience of the development team

- The team may lack experience in developing mobile applications, which may result in the product not being optimized for performance and quality.

Limited budget

- The project budget may not be enough to cover the entire development costs, including software, hardware, and marketing, leading to feature cuts or delays.

Difficulty in gathering requirements

- Identifying and gathering requirements from stakeholders can be difficult, especially if there is no close coordination between the parties, leading to a lack of consensus on the project goals.

Time management

- Lack of a specific schedule or detailed plan can result in delays and failure to complete project phases on time.

Changing Stakeholder Requirements

- Requirements may change during development due to feedback from users or other stakeholders, making it difficult to adjust plans and budgets.

Technical Risks

- Technical issues, such as system errors or incompatibilities between platforms (Android, iOS), can affect the stability and performance of the application.

Market Competition

- The presence of similar applications can reduce the ability to attract users and affect the project's revenue, especially if the application does not have a clear point of differentiation.

Difficulty in promoting and marketing the application

- Building an effective marketing strategy to reach and attract the target audience is a challenge, especially in a fiercely competitive environment.

PLATFORM USED



android

- Ability to take full advantage of the features of the Android operating system.
- User interface optimized for Android devices.
- Reach a large number of Android users in the market.



iOS

- Smooth and user-friendly interface, optimized for Apple devices.
- Ability to integrate with iOS-specific services and features, such as push notifications, Apple Pay.
- Reach users in the Apple ecosystem, who tend to have higher spending.

POTENTIAL RISKS



Technical risks

- System errors: Errors in the source code or software bugs can cause problems for the application, affecting the user experience and reliability of the product.
- Lack of optimization: If the application is not optimized for performance and speed, this can lead to slow performance and user dissatisfaction.

Financial risks

- Increased budget: Development costs may be higher than expected due to unforeseen factors, such as additional feature requests or technology changes.
- Lack of additional funding: If the project does not receive sufficient funding during the development process, this can affect the ability to complete the project on time and with quality.

Personnel risks

- Limited experience: The development team may lack experience in building mobile applications, leading to difficulties in implementing features and troubleshooting.
- Unexpected resignations: The sudden resignation of one or more team members can disrupt the project progress and increase the pressure on the remaining members.



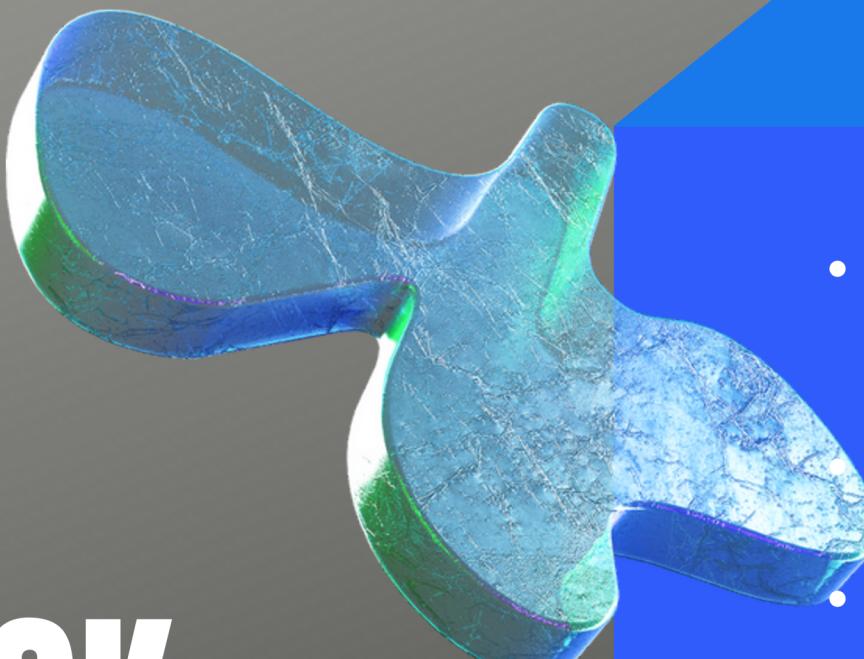
Risks from changing requirements

- Changes from stakeholders: Requirements from stakeholders can change during the development process, making it difficult to adjust the work plan and progress.
- Difficulty in managing requirements: The inability to control emerging requirements can lead to not completing planned features or increasing development time.

Market Risk

- Increased Competition: The emergence of similar or better applications could affect user acceptance of CampusExpense Manager.
- Meeting User Needs: If the application fails to adequately meet user needs or wants, this could result in a decrease in user numbers and revenue.

HOW TO OVERCOME RISK



Managing Human Resources Risks

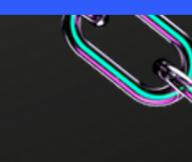
- Training and development: Provide training and mentoring to enhance the skills of less experienced team members.
- Contingency planning: Cross-assign tasks among team members to minimize reliance on a single person.
- Promoting motivation and team engagement: Creating a positive work environment and promoting team spirit to reduce the risk of turnover.

Managing Budget Risks

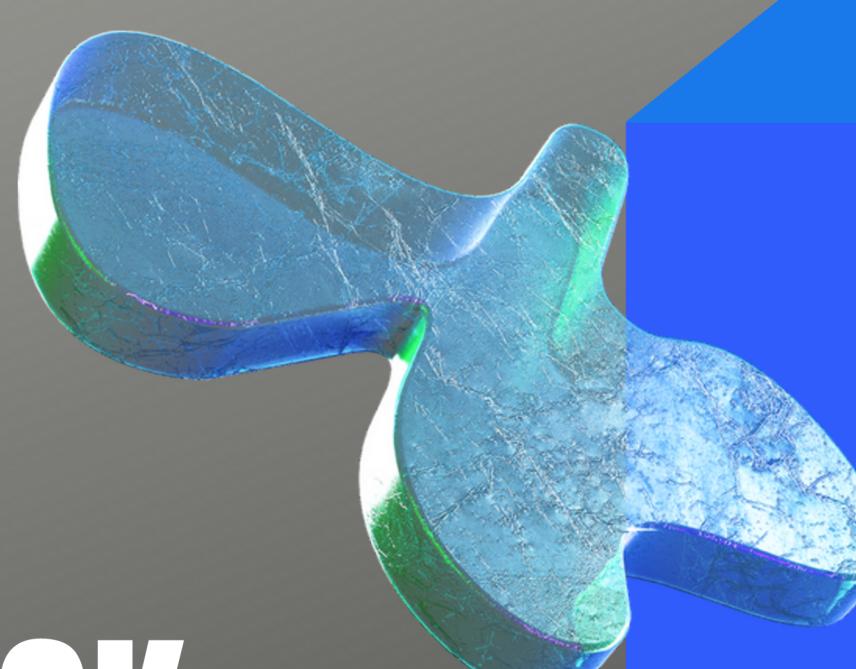
- Budgeting and monitoring: Regularly review and update the budget to ensure that expenses do not exceed the plan.
- Managing flexible costs: Review and cut unnecessary expenses if there are financial problems.
- Contingency funding: Establish a contingency budget to handle unexpected expenses.

Managing Technical Risks

- Comprehensive testing: Implement regular software testing (unit, integration, and user testing) to ensure that features are working properly.
- Use a bug tracking tool: Use tools like JIRA or Trello to manage and resolve bugs during development.
- Prepare a troubleshooting plan: Set up scenarios and contingencies for potential technical issues.



HOW TO OVERCOME RISK

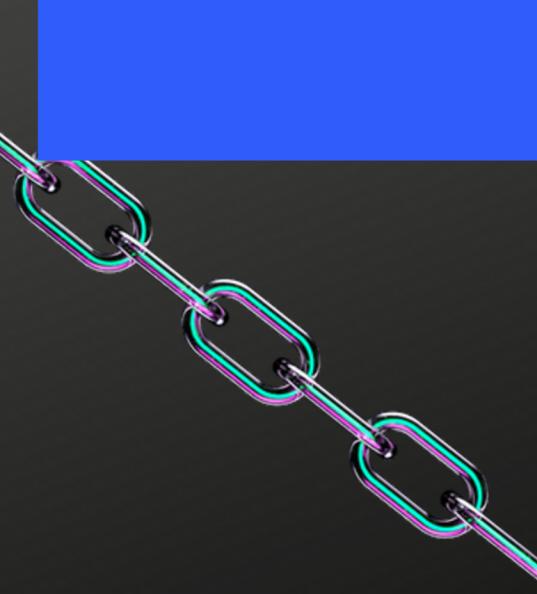


Manage Schedule Risks

- Create a detailed plan: Create a clear schedule with milestones to control progress.
- Prioritize important tasks: Focus on key project features first, ensuring they are completed on time.
- Regular progress checks: Hold weekly progress reviews to update the status and adjust the plan as needed.

Manage Partner or Third-Party Risks

- Choose trusted partners: Choose reputable partners when using third-party services, such as APIs or cloud services.
- Regular reviews and testing: Regularly check the quality of external services to ensure system stability.
- Redundancy and rapid changeover: Build in a quick backup plan if something goes wrong with a third-party service.



THANKS YOU!

See you in the next presentation.

