MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

MoneyEasy - Financial Management Application

GSP25SE47	
Group Members	Phạm Nhật Linh - Team Leader - SE173687 Đặng Ngọc Minh Trí - Team Member - SE173705 Nguyễn Thương Huyền - Team Member - SE161803 Dương Tôn Bảo - Team Member - SE171065 Đặng Phan Gia Đức - Team Member - SE171092
Supervisor	Mr. Nguyễn Trọng Tài
Capstone Project code	SP25SE065

- Ho Chi Minh, Jan 2025 -

Table of Contents

Acknowledgement

Four years of learning, growth, and collaboration have led us to this moment—our capstone project at FPT University. It's incredible to reflect on how we started as eager students, taking our first steps in software engineering, and now stand as aspiring professionals, ready to tackle real-world challenges.

None of this would have been possible without the unwavering support of our families. Through late nights, tough deadlines, and moments of doubt, their encouragement and belief in us kept us going. We are truly grateful for their sacrifices and constant motivation.

A special thank you to our project supervisor, Mr. Nguyễn Trọng Tài, whose guidance, expertise, and valuable feedback helped shape MoneyEasy into what it is today. We also deeply appreciate all the instructors at FPT University who have supported us throughout this journey, equipping us with the knowledge and skills to bring our ideas to life.

As a team, we couldn't be prouder of what we've built. MoneyEasy isn't just a project—it's a reflection of our dedication, hard work, and passion for creating meaningful solutions. The road wasn't always easy, but every challenge made us stronger, more resilient, and better prepared for what's ahead.

This milestone is just the beginning. As we step into the next chapter of our journey, we carry with us not only the technical expertise we've gained but also the friendships, experiences, and lessons that have shaped us. The future is full of possibilities, and we're excited to see where it takes us.

Definition and Acronyms

Table 1 - Definition and acronyms

Acronym	Definition
BR	Business Rule
ERD	Entity Relationship Diagram
GUI	Graphical User Interface
PM	Project Manager
SRS	Software Requirement Specification
UC	Use Case
API	Application Program Interface

I. Project Introduction

1. Overview

1.1 Project Information

- Project name: MoneyEasy - Financial Management Application
- Project code: GSP25SE47
- Group name: SP25SE065
- Software type: Web application and Mobile application

1.2 Project Team

Table 2 - Project team

Full Name	Role	Email	Mobile
Nguyễn Trọng Tài	Lecturer	TaiNT51@fe.edu.vn	0903346184
Phạm Nhật Linh	Leader	linhpnse173687@fpt.edu.vn	0854062409
Đặng Ngọc Minh Trí	Member	tridnmse173705@fpt.edu.vn	0901997815
Nguyễn Thương Huyền	Member	huyenntse161803@fpt.edu.vn	0979084700
Dương Tôn Bảo	Member	baodtse171065@fpt.edu.vn	0909251504
Đặng Phan Gia Đức	Member	ducdpgse171092@fpt.edu.vn	0384718905

2. Product Background

Managing money can be overwhelming, especially when juggling income, expenses, savings, and investments. Many people struggle with budgeting, keeping track of their spending, and staying financially disciplined. Traditional methods like spreadsheets or manual tracking are time-consuming and prone to mistakes, making it hard to stay on top of finances.

That's why we created MoneyEasy—a simple, intelligent, and automated way to take control of your finances. Whether you prefer structured budgeting models like 6 JARs, 50-30-20, or 80-20, or need automated tracking of your income and expenses through bank integration, MoneyEasy makes managing money effortless. With detailed reports and easy-to-read charts, you can see exactly where your money is going—daily, weekly, monthly, or yearly. Smart alerts keep you on track by notifying you when you're about to overspend.

By automating financial tracking, simplifying budgeting, and providing real-time insights, MoneyEasy takes the stress out of managing money. Whether you're saving for a big goal, cutting back on spending, or planning your next investment, MoneyEasy helps you stay in control—effortlessly.

3. Existing Systems

3.1 MISA Money Keeper

MISA Money Keeper is a free expense management app that helps users track income and expenses, set budgets, receive spending alerts, and calculate loan interest and taxes. It also supports data export (Excel, PDF) and quick search functions.

Pros:

- Free to use with many financial tracking features.
- Budget planning and spending limit alerts.
- Supports tax and loan interest calculation.
- Allows data export in Excel and PDF format.

Cons:

- No automatic invoice scanning—users must manually input transactions.
- Manual entry can be time-consuming and prone to errors.

3.2. Mint

Mint is a free, automated budgeting tool that syncs with bank accounts to track expenses, categorize spending, and provide financial insights. It offers bill reminders, alerts, and spending trends, making budgeting more accessible for everyday users.

Pros:

- Free to use with automated expense tracking
- Provides financial insights and reports
- Includes bill reminders and alerts

Cons:

- Limited customization for budgeting models
- Ad-based platform can be distracting
- Occasional syncing delays with bank accounts

4. Business Opportunity

Managing personal finances is a growing concern, especially as people seek better ways to budget, save, and invest efficiently. Traditional financial management methods, such as spreadsheets or manual tracking, are time-consuming and prone to errors. Many individuals struggle with spending discipline, financial planning, and investment optimization, leading to poor financial decisions and stress.

Market Trends:

- Rising demand for financial literacy – More people are seeking tools to manage their income, expenses, and investments effectively.
- Shift to digital finance – Consumers, especially younger generations, prefer mobile-first, automated solutions for financial tracking and planning.
- Growth of AI-powered financial assistants – AI-driven budgeting and investment recommendations are becoming mainstream, offering users smarter, data-driven insights.
- Increased focus on financial well-being – People are prioritizing savings, debt reduction, and financial independence, creating demand for tools that support these goals.

Opportunity for MoneyEasy: With these trends, MoneyEasy is positioned to revolutionize personal finance management by offering a smart, automated, and user-friendly solution. By integrating personalized budgeting models, automated expense tracking, financial goal setting, and AI-driven insights, MoneyEasy can empower users to take control of their finances with ease.

The increasing reliance on digital financial tools, combined with growing consumer demand for automated budgeting and smarter financial decision-making, presents a significant opportunity for MoneyEasy to become a go-to financial management app. By delivering an intuitive, insightful, and proactive financial experience, MoneyEasy can help users achieve long-term financial stability and success.

5. Software Product Vision

Managing money can feel overwhelming—juggling bills, savings, and expenses while trying to stay on top of your financial goals. We get it. That's why we created MoneyEasy, a financial management app designed to simplify the way you handle your money, so you can stress less and achieve more.

Think of MoneyEasy as your personal finance assistant, always keeping an eye on your budget, reminding you of upcoming expenses, and helping you make smarter financial decisions. No more complicated spreadsheets, no more guesswork—just clear insights, smart tracking, and effortless budgeting at your fingertips.

Whether you're saving for a dream vacation, managing daily expenses, or planning for the future, MoneyEasy adapts to your lifestyle. With automated tracking, personalized budgeting models, real-time spending insights, and goal-setting tools, we make financial management simple, smart, and stress-free.

What makes MoneyEasy different? It's built for real life. Our app doesn't just track numbers—it helps you build better money habits, avoid overspending, and stay on track with financial goals. With secure bank integration, smart alerts, and AI-powered insights, you'll always know exactly where your money is going.

At the end of the day, MoneyEasy isn't just an app—it's your partner in financial success. Whether you're just starting your financial journey or looking for better ways to manage your money, we're here to make it easier. Take control of your finances with confidence.

6. Project Scope & Limitations

6.1 Major Features

Table 3 - Major features

Feature ID	Feature Description
FE1: User Management	Manages user authentication and profile data, including sign-up, login (with email verification), password recovery, and user profile CRUD operations. Admins have full control over user accounts.

FE2: Transaction Management	Allows users to track income and expenses manually or automatically (via bank integration). Supports structured budgeting using various financial models (6 JARs, 80-20, 50-30-20, etc.), automatic updates on model expiration, and financial goal management.
FE3: Group Management	Supports shared financial goals by enabling users to create, join, and manage group funds. Features include member invitations (via email/QR code), group fund contributions, withdrawal requests, voting on expenses, and transaction history tracking. Groups can be archived when dissolved.
FE4: Statistic Management	Provides detailed financial reports and spending insights through visual charts (daily, weekly, monthly, yearly). Includes alerts for overspending, financial goal tracking, and budgeting tools to help users stay within set limits.
FE5: Category Management	Admins can create, update, view, and delete categories, ensuring that users have a well-structured system for classifying their expenses. By organizing expenses into relevant categories such as food, rent, and entertainment, users can track their spending habits more effectively.
FE6: Quiz Management	Admins can add and modify quiz questions, answers, and scoring rules. Additionally, they can review user quiz results to provide personalized spending model recommendations. This interactive tool helps users gain insights into their financial habits and make informed budgeting decisions.
FE7: Spending Model Management	Spending models help users adopt structured budgeting strategies based on their financial behavior. Admins can assign specific categories to spending models, ensuring that users receive relevant guidance on managing their expenses. This feature plays a crucial role in improving users' financial literacy and spending discipline.
FE08: Bank Account Management	This feature allows users to securely add, update, view, and remove their linked bank accounts within the system. By integrating bank accounts, users can track their transactions more efficiently and gain better insights into their financial activities. The system ensures secure data handling and provides a seamless experience for managing multiple accounts while maintaining privacy and security.

6.2 Limitations & Exclusions

6.2.1. Limitations

- **LI-01:** MoneyEasy currently supports only the **Android** platform; an **iOS** or web version is not yet available.
- **LI-02: AI-powered financial insights** rely on historical user transaction data and may not provide fully personalized financial predictions.
- **LI-03:** Financial reports and analytics **should not be considered a substitute for professional financial advice** and are intended only as a reference for spending decisions.
- **LI-04:** MoneyEasy does not currently support **multi-currency tracking**, meaning users cannot manage transactions in multiple currencies within a single account.

6.2.2. Exclusions

- **EX-01:** MoneyEasy **does not support cryptocurrency transactions or digital assets** such as Bitcoin or Ethereum.
- **EX-02:** The system **does not provide loan interest calculations**; users can only manually enter monthly payments for tracking purposes.
- **EX-03:** The application **does not offer investment advisory services** but solely focuses on financial tracking and management.
- **EX-04:** MoneyEasy does not provide **legal or financial protection** in case of disputes over group funds or shared expenses.

II. Project Management Plan

1. Overview

1.1 Scope & Estimation

Table 4 - Scope and estimation

No	WBS Item	Complexity	Est. Effort (man-days)
1	**Initiating**		**25**
1.1	Create a project management plan	Complex	8
1.2	Rent a server, buy a domain	Medium	5
1.3	Define required functionality	Medium	5
1.4	Define required screens	Medium	4
1.5	Define tables, relationships, and data schemas	Medium	7
2	**Planning**		**80**
2.1	Technology Training	Medium	15
2.2	Requirement Analysis	Complex	10
2.3	BRS – Business requirement specification	Complex	10
2.4	ERD Design	Complex	10
2.5	UI Design (Figma Design)	Medium	10
2.6	Database Design	Medium	20
2.7	Create test cases	Medium	5
2.8	Team meeting	Simple	5
2.9	Supervisor meeting	Medium	3
2.9	Create Source Base	Complex	12
3	**Executing**		**152**
3.1	**Iteration 1**		**42**
3.1.1	Set up project architecture (Frontend, Backend)	Complex	10
3.1.2	Develop authentication module (Login, Sign-up, Role Management)	Complex	5
3.1.3	Implement basic UI screens	Medium	4
3.1.4	Develop user profile & settings	Simple	4
3.1.5	Database integration	Simple	3
3.1.6	Implement API for user authentication & profile	Medium	2
3.1.7	Unit testing	Medium	10
3.2	**Iteration 2**		**35**
3.2.1	Develop Core Features (Transaction, User Management)	Medium	2
3.2.2	View list of transactions and view transaction details	Simple	2

3.2.3	Update transaction	Medium	3
3.2.4	Remove transaction	Medium	1
3.2.5	Create transaction for group	Medium	5
3.2.6	Update transaction for group	Simple	1
3.2.7	Get transaction - group(list, detail)	Medium	1
3.2.8	Remove transaction - group	Medium	2
3.2.9	Generate running balance for user	Medium	1
3.2.10	Generate running balance for group	Simple	2
3.2.11	Generate daily report	Medium	1
3.2.12	Auto update/add transaction when demo bank (EZB) updated	Medium	2
3.2.13	Add one or many spending model	Medium	3
3.2.14	Update model basic information	Complex	3
3.2.15	Add new SubCategories	Medium	2
3.2.16	Get list,details SubCategory	Simple	4
3.2.17	Create payment for customer	Medium	5
3.2.18	Get current / list history goal for group	Simple	5
3.3	**Iteration 3**		**28**
3.3.1	Add notification to user	Medium	2
3.3.2	Add notification to group	Medium	2
3.3.3	Remind finance	Simple	1
3.3.4	Create quiz	Medium	1
3.3.5	Update quiz	Medium	1
3.3.6	Get quiz (list, detail)	Complex	2
3.3.7	Calculate depreciation	Complex	1
3.3.8	Manage depreciation	Medium	2
3.3.9	Summarize by period and spending model	Medium	1
3.3.10	Basic statistic charts	Medium	3
3.3.11	Frequency of budgeting models	Simple	5
3.4	**Iteration 4**		**7**

3.4.1	Get list, detail index document from QdrantDb (admin)	Complex	10
3.4.2	Integration testing	Medium	10
3.4.3	System testing	Complex	12
3.4.4	Unit testing	Medium	3
3.4.5	Testing in staging environment	Medium	12
3.4.6	Deploy code in product environment	Medium	3
3.4.7	Testing in product environment	Medium	14
4	**Monitoring and Controlling**		**60**
4.1	Monitoring and Controlling	Complex	20
4.2	Scope of control	Medium	15
4.3	Tracking progress	Medium	10
4.4	Monitoring and controlling risks	Complex	15
5	**Closing**		**40**
5.1	Lesson learn	Simple	10
5.2	Final project report	Simple	15
5.3	Project archive	Simple	15
Total Estimated Effort (man-days)			***488***

1.2 Project Objectives

Table 5 - Project objective

No	Testing Stage	Test Coverage	No. of Defects	% of Defect	Notes
1	Unit Test	56	0	0%	N/A
2	System Test	313	0	0%	N/A

1.3 Project Risks

Table 6 - Project risks

#	Risk Description	Impact	Possibility	Response Plans
01	Regulatory compliance issues (violating financial laws, KYC/AML regulations)	Critical	Medium	Work with legal experts, ensure compliance with local financial regulations, implement automated KYC/AML verification.
02	Security vulnerabilities leading to fraud or hacking	Critical	Medium	Implement end-to-end encryption, conduct regular security audits, use multi-factor authentication (MFA).
03	Payment processing failures (delayed transactions, failed payments)	High	Medium	Partner with reliable payment gateways, implement failover mechanisms, monitor transaction logs in real time.
04	High chargeback rates or fraudulent transactions	High	Medium	Use AI-based fraud detection, implement dispute resolution protocols, work closely with financial institutions.
05	System downtime affecting app availability	High	Medium	Implement cloud-based scalable infrastructure, ensure 24/7 monitoring, set up redundancy systems.
06	Poor customer support response times	Medium	Medium	Implement chatbot support for instant responses, scale support team based on demand, provide clear FAQs and self-help guides.
07	Partnership issues with banks/payment providers	High	Medium	Diversify financial service providers, negotiate flexible contracts, have alternative providers as backup.

08	Cash flow issues due to delayed revenue streams	High	Medium	Optimize pricing models, introduce multiple revenue streams, maintain an emergency operating fund.

2. Management Approach

2.1 Project Process

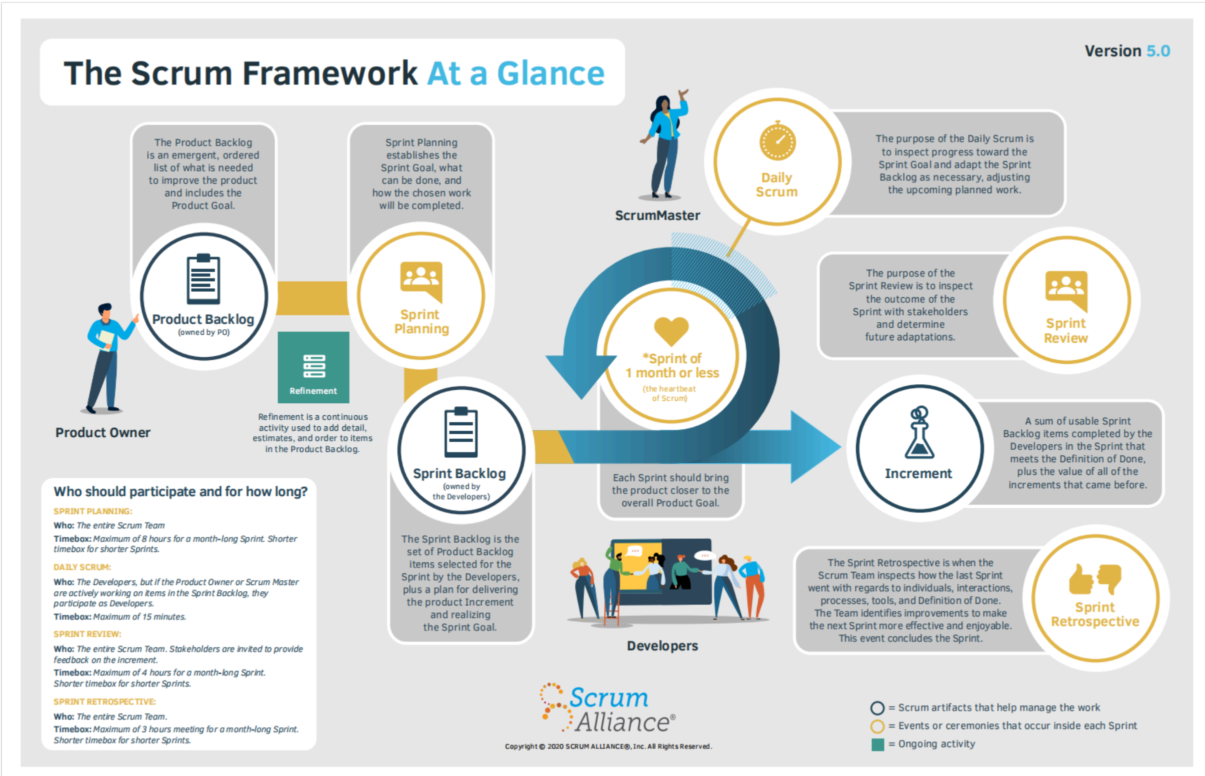


Figure 1 - Project process

Source: https://www.scrumalliance.org/about-scrum

We chose to build this project using the Scrum framework from Agile because:

It allows us to adapt to customers' unpredictable requirements. By breaking the development down into smaller, manageable parts, we can refine the process step by step, ensuring that the final system meets the customer's vision.

Team Roles:

- LinhPN - Mobile Developer
- TriDNM - Back-end developer, Mobile Developer
- HuyenNT - Back-end Developer
- BaoDT - Front-end Developer, Mobile Developer
- DucDPG - Back-end Developer

Scrum Process:

1. Scrum Sprints:

- Each Scrum sprint should be 4 weeks. This is a common sprint duration that allows for meaningful progress while keeping the development agile and responsive to changes.
- The team should aim for 2-3 week sprints, as this strikes a good balance between making tangible progress and maintaining flexibility.

2. Scrum Ceremonies:

- Sprint Planning: At the start of each sprint, the team gathers to plan the work for the upcoming sprint.
- Daily Scrum: The team has a brief (15 minute) daily standup meeting to discuss progress, blockers, and plan for the day.
- Sprint Review: At the end of each sprint, the team demonstrates the completed work to stakeholders and gets feedback.
- Sprint Retrospective: Immediately after the Sprint Review, the team reflects on what went well, what could be improved, and agrees on actionable items for the next sprint.

3. Roles and Responsibilities:

- Mobile Developer (LinhPN, BaoDT, TriDNM): Revolve around designing, building, and maintaining mobile applications for platforms like Android
- Back-end Developer (HuyenNT, TriDNM, DucDPG): Focuses on building the server-side logic and API.
- Front-end Developers (BaoDT): Collaborate on the client-side application development.
- The whole team participates in sprint planning, reviews, and retrospectives to ensure a collaborative approach.

2.2 Quality Management

In this project, we employ the following strategy to raise project quality:

- API convention:

  The response of APIs must be returned following:

  - 200, 400, 401, 403, 404, 500 are the status codes used in the system.
  - Message body must be returned in JSON format.
- Coding convention:
  - Set the meaningful names for variables.
  - Add comments to code can be confusing.
  - Use Pascal case for naming Class, Interface, Method, Property, Constant.
  - Use Camel case for naming Variables.
  - Use Screaming snake case for naming Enum, constants.
  - Use Pascal case for naming single UI
  - Use Camel case for naming hooks

2.3 Training Plan

Table 7 - Training plan

Training Area	Participants	When, Duration	Waiver Criteria
React Native - Basics & Navigation	TriDNM, LinhPN	Week 1, 3 days	Mandatory
React Native - State Management (Redux/Context)	TriDNM, LinhPN	Week 1, 4 days	Mandatory
Figma - UI Design Principles Figma - UI Design Principle	All members	Week 1, 3 days	Mandatory
Figma - Prototyping & Developer Handoff	All members	Week 1, 4 days	Mandatory
Firebase - Hosting & Security Rules	DucDPG	Week 1, 4 days	Mandatory
Front-end Architecture	BaoDT	Week 2, 7 days	Mandatory
CI/CD - GitHub Actions & Workflow Setup	DucDPG	Week 2, 3 days	Mandatory
Figma	All members	Week 1, 7 days	Mandatory
Firebase	All members	Week 1, 7 days	Mandatory
CI/CD	DucDPG	Week 2, 7 days	Mandatory
Redis	TriDMN	Week 3, 7 days	Mandatory

Restful API	HuyenNT	Week 3, 7 days	Mandatory
Redis - Session Storage & Pub/Sub	TriDNM	Week 3, 3 days	Mandatory
Restful API - Documentation with Swagger	HuyenNT	Week 3, 3 days	Mandatory

3. Project Deliverables

Table 8 - Project deliverables

Iteration	Sprint objective	Duration	Deliverable
1	Database Design, User Stories, SRS Document	14 days	Software Requirements and defined user stories Report 1 - Project Introduction
1	Project Initialization, UI/UX Design	14 days	Project planning Report 2 - Project Management Plan
2	Basic authentication and initial transaction module, User management module and group transaction	14 days	User management module, expanded transaction features, functional testing, working authentication module, basic statistic system, demo Report 3 - Software Requirement Specification
3	Chatbot and statistic module	14 days	Report 4 - Software Design Description
4	Testing and Bug Fixing	14 days	System testing, bug fixes, usability improvements, ready for demo Report 5 - Software Testing Documentation

4	Final Deployment and Documentation	14 days	User guides, final system testing, final report, and deployment Report 6 - Release Package & User Guides

4. Responsibility Assignments

* *D~Do; R~Review; S~Support; I~Informed; <blank>- Omitted*

Table 9 - Responsibility assignments

Responsibility	Linh PN	Huyen NT	Tri DNM	Bao DT	Duc DPG
Project Planning & Tracking	S	D	R	R	R
Prepare Project Introduction Document	S	S	D	R	I
Prepare SRS Document (Overview Part)	R	D	S	S	R
Prepare SRS Document (User Requirements)	D	R	S	S	R
Track backlog	S	D	D	D	D
Quality management	S	I	I	I	I
Design database	D	D	D	D	D
UI/UX design	D	S	D	D	D
Create test plan	R	S	R	R	R
Testing	R	S	I	I	I

API development	S	I	S	I	I
Monitor project progress and quality	S	I	I	I	I

5. Project Communications

Table 10 - Project communications

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
Supervisor communication	Mr. Nguyen Trong Tai and team members	Review documentation Demonstrate features Evaluate progress and result	2 times per week	Face to face, Google Meet, Zalo
Daily meeting	Team members	Raise opinions, problems Ask for help from other members Report working status to leader	6 times per week	Face to Face, Messenger, Google Meet, Zalo

6. Configuration Management

6.1 Document Management

We use Google Drive, Google Sheets, Jira to manage documents since they supply real-time data synchronization. Team members can easily collaborate in real-time thanks to Google Drive. Each participant can at once view the outcomes of the others. We can also evaluate document updates in Office 365 so that we can do so whenever necessary.

Link Google Drive: Link (Private)

6.2 Source Code Management

We decided to manage our source code on GitHub. The version control system makes it possible for team members to work together on source code efficiently, conveniently, and easily to resolve conflict. It is especially useful for looking back in time and at once recognizing changes a collaborator made.

Link GitHub BE: Source code BE (Public)

Link GitHub FE (mobile): Source code FE (Public)

Link GitHub FE (Web): Source code FE web (Public)

6.3 Tools & Infrastructures

6.3.1. Tools

Table 11 - Tools

Type	Tools
IDEs/Editors	Visual Studio Code, Visual Studio, Android Studio
UML tools	Visual Paradigm
Source Version Control	GitHub
Project Management Tool	Jira, Google Sheets
Database	SQL Server Management Studio, Azure Data Studio
Docs Management	Google Drive
UI/UX	Figma

6.3.2. Infrastructures

Table 12 - Infrastructures

Type	Infrastructure
Framework	Entity Framework Core 8 (Backend), React Native (Mobile), Nestjs (Web)
Programming Languages	Typescript, C#
DBMS	SQL Server
Deployment Server	Github Action, VPS (Digital Ocean) - Ubuntu server - NGINX
Storage	Firebase Storage, Redis

III. Software Requirement Specification

1. Product Overview

MoneyEasy is a financial management platform that helps users track expenses, manage budgets, and improve financial literacy through quizzes and analytics. It supports group-based financial management and detailed reporting. Admins can oversee accounts, spending models, and categories. MoneyEz offers a structured and insightful approach to personal and group finance management.

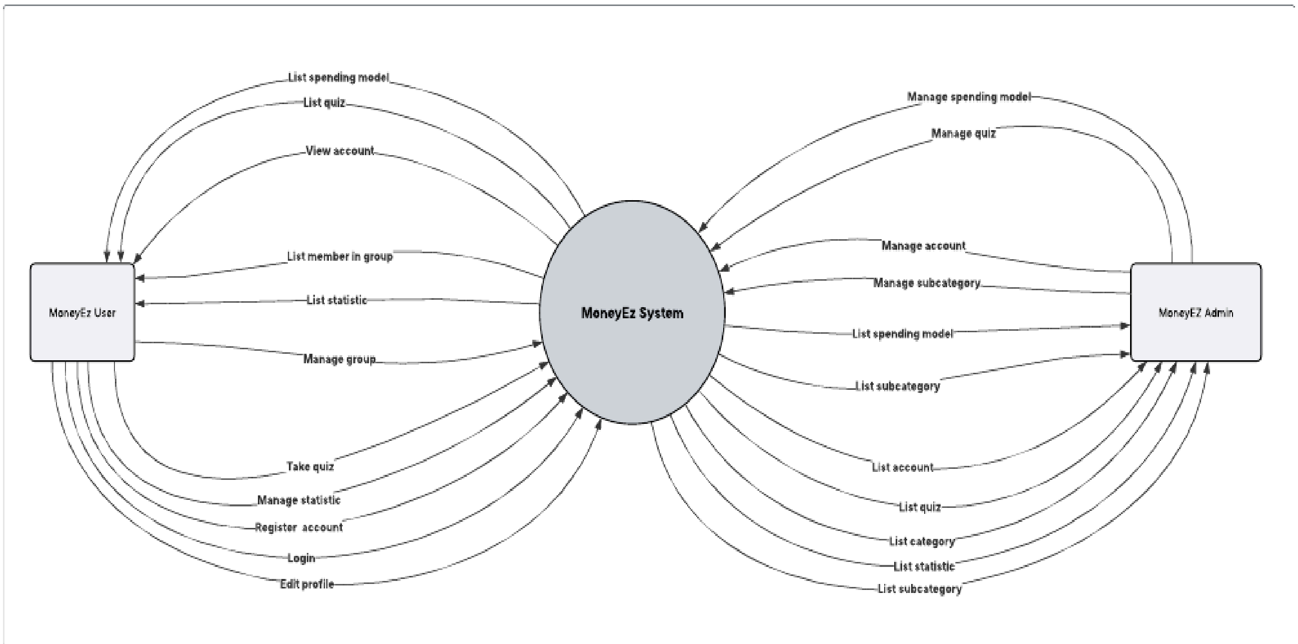


Figure 2 - Product overview

2. User Requirements

2.1 Actor

Table 13 - Actor

No	Actor	Description
1	Admin	The admin is responsible for overseeing the system, including account management, quiz management, and the administration of categories, subcategories, and the spending model.
2	User	The user is responsible for tracking and managing expenses, monitoring spending statistics, and exploring and utilizing categories and spending models within the system.
3	System handler	The system is responsible for suggesting spending models, classifying transactions, and sending notifications in real time using AI.

2.2. Use Case Diagram

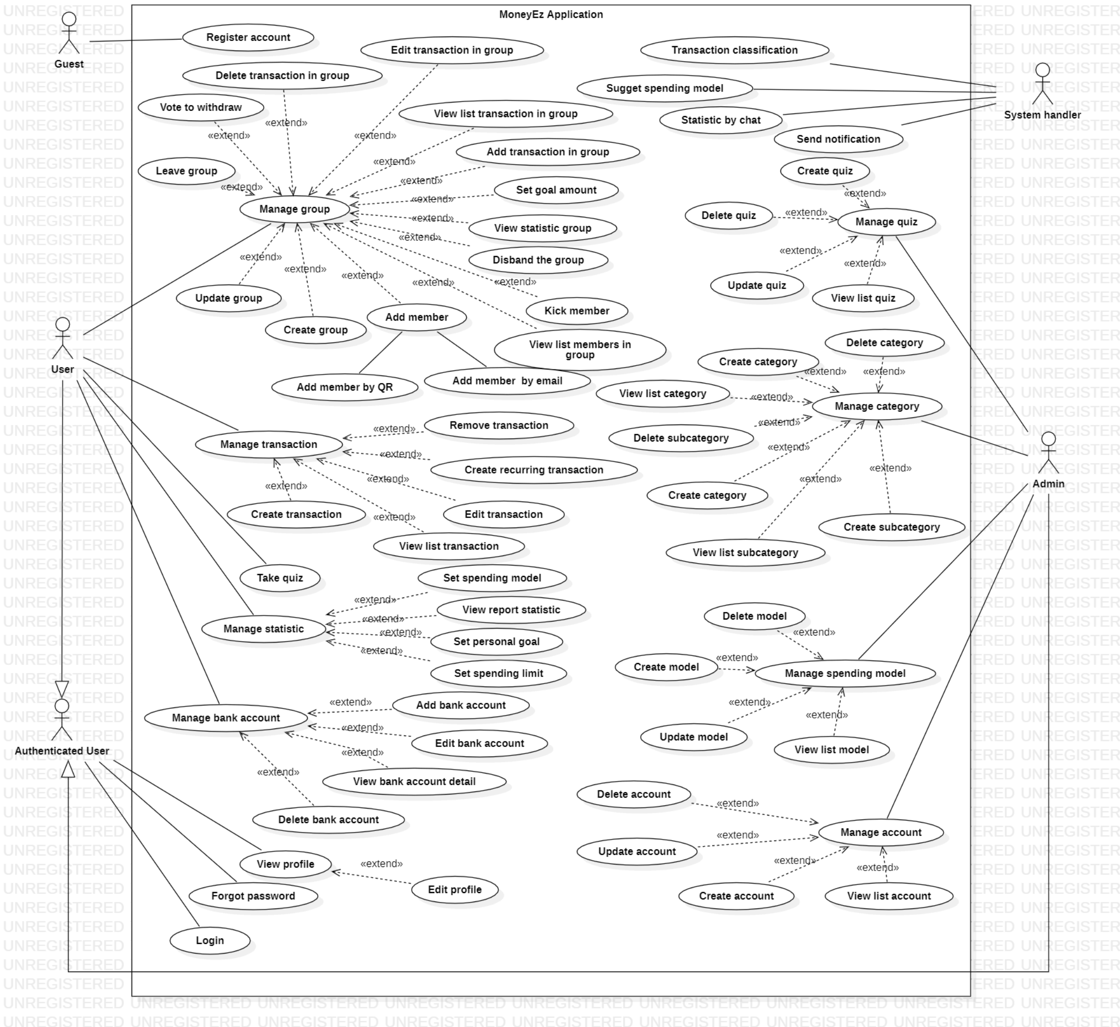


Figure 3 - Use case diagram

2.3. List of Use Case Descriptions

2.3.1. User Management Use Case Descriptions

Table 14 - User Management Use Case Descriptions

ID	Use Case	Actors	Use Case Description
01	View profile	User, Admin	The various users can view their own profile information, such as their personal details, contact information, and any roles or statuses associated with their account.
02	Edit profile	User	Users can update their profile information, such as their name, contact details, or preferences.

03	Login	User, Admin	Users can log in to the system using their assigned username and password.
04	Forgot password	User, Admin	Users can initiate a password reset process if they have forgotten their login credentials.
05	Create account	Admin	The admin creates a new admin account to manage the system.
06	Register account	Guest	Guests can create a new user account in the system, either as a customer or one of the other user roles.
07	Delete account	Admin	The admin removes an existing account from the system.
08	Update account	Admin	The admin updates an existing account's details.
09	View list account	Admin	The admin views a list of all registered accounts.

2.3.2. Transaction Management Use Case Descriptions

Table 15 - Transaction Management Use Case Descriptions

ID	Use Case	Actors	Use Case Description
01	Create transaction	User	The user records a new transaction in the system to track their expenses or income.
02	View list transactions	User	The user views a list of all recorded transactions to monitor their spending and income.
03	Remove transaction	User	The user deletes a previously recorded transaction from the system.
04	Edit transaction	User	The user updates the details of an existing transaction.
05	Create recurring transaction	User	The user sets up a recurring transaction that automatically records at specified intervals (e.g., weekly, monthly).

2.3.3. Group Management Use Case Descriptions

Table 16 - Group Management Use Case Descriptions

ID	Use Case	Actors	Use Case Description

01	Create group	User	The user creates a spending group to track and manage shared expenses.
02	Update group	User	The group admin modifies the group's details, including name and goal amount.
03	Request a withdrawal	User	Group members send withdrawal requests to the group and wait for leader approval.
04	Remind raising fund	User	Leader reminds group members to pay the fund.
05	View list transactions in group	User	Users view all group transactions with details like date and amount.
06	Raising funds request	User	Group members send payment requests to the group and wait for leader approval.
07	Disband the group	User	Admin deletes the group, removing all members and transactions.
08	Set goal amount	User	Admins set a spending or saving goal for the group.
09	View list members in group	User	Users can see all group members and their roles.
10	Add member by QR	User	Users join the group by scanning a shared QR code.
11	Add member by email	User	Admins invite users to join the group via email.
12	Kick member	User	The admin removes a member from the group.
13	View statistic group	User	Users access spending analytics and financial reports.
14	Leave group	User	A user exits the group, losing access to transactions.

2.3.4. Statistic Management Use Case Descriptions

Table 17 - Statistic Management Use Case Descriptions

ID	Use Case	Actors	Use Case Description
01	Set spending model	User	Users select a predefined spending model to categorize expenses and optimize financial management.
02	Set spending limit	User	Users set a spending limit to control expenses and receive alerts when nearing the defined threshold.

03	View report statistic	User	Users access detailed reports, visual charts, and spending trends to analyze financial habits effectively.
04	Set personal goal	User	Users define a financial goal, such as savings or budget targets, and tracks progress over time.

2.3.5. Quiz Management Use Case Descriptions

Table 17 - Quiz Management Use Case Descriptions

ID	Use Case	Actors	Use Case Description
01	Create quiz	Admin	The admin creates a new quiz with questions to help users choose a suitable spending model.
02	Update quiz	Admin	The admin modifies quiz details, such as questions, answers, and recommendations for spending models.
03	View list quizzes.	Admin	The admin views all created quizzes, including their questions and assigned spending models.
04	Delete quiz	Admin	The admin removes a quiz from the system, preventing users from accessing it.
05	Take quiz	User	The user answers quiz questions to receive personalized recommendations for a suitable spending model.

2.3.6. Category Management Use Case Descriptions

Table 17 - Category Management Use Case Descriptions

ID	Use Case	Actors	Use Case Description
01	Create category	Admin	The admin creates a new spending category, such as food, utilities, or shopping, for expense classification.
02	Update category	Admin	The admin modifies category details, including name or description, to better organize spending records.
03	View list category	Admin	The admin views all spending categories available in the system for better expense management.

04	Delete category	Admin	The admin removes an unnecessary spending category, ensuring a more streamlined classification system.
05	Create subcategory	Admin	The admin creates a subcategory linked to one or multiple categories.
06	Update subcategory	Admin	The admin modifies subcategory details, including name or associated categories, for better expense classification.
07	View list subcategory	Admin	The admin views all subcategories, their linked categories, and related spending records.
08	Delete subcategory	Admin	The admin removes a subcategory, updating all related expenses to maintain data consistency.

2.3.7. Spending Model Management Use Case Descriptions

Table 18 - Spending Model Management Use Case Descriptions

ID	Use Case	Actors	Use Case Description
01	Create spending model	Admin	The admin defines a new spending model with rules and guidelines to help users manage their expenses efficiently.
02	Update spending model	Admin	The admin modifies an existing spending model by adjusting its parameters, categories, or financial recommendations.
03	View list spending model	Admin	The admin reviews all available spending models, including their details and user adoption rates.
04	Delete spending model	Admin	The admin removes an outdated or unused spending model from the system to keep the database relevant.

2.3.8. Bank Account Management Use Case Descriptions

Table 19 - Bank Account Management Use Case Descriptions

ID	Use Case	Actors	Use Case Description
01	Add bank account	User	Allows users to securely link a new bank account to the system, enabling transaction tracking and financial management.

02	Edit bank account	User	Enables users to update bank account details, such as account name or linked information, ensuring accurate financial records.
03	View bank account detail	User	Provides users with a detailed overview of a specific bank account, including transaction history and current balance.
04	Delete bank account	User	Allows users to remove a linked bank account from the system, ensuring they have control over their financial data.

3. Functional Requirements

3.1 System Functional Overview

3.1.1 Screens Flow

3.1.1.1 Mobile module for user

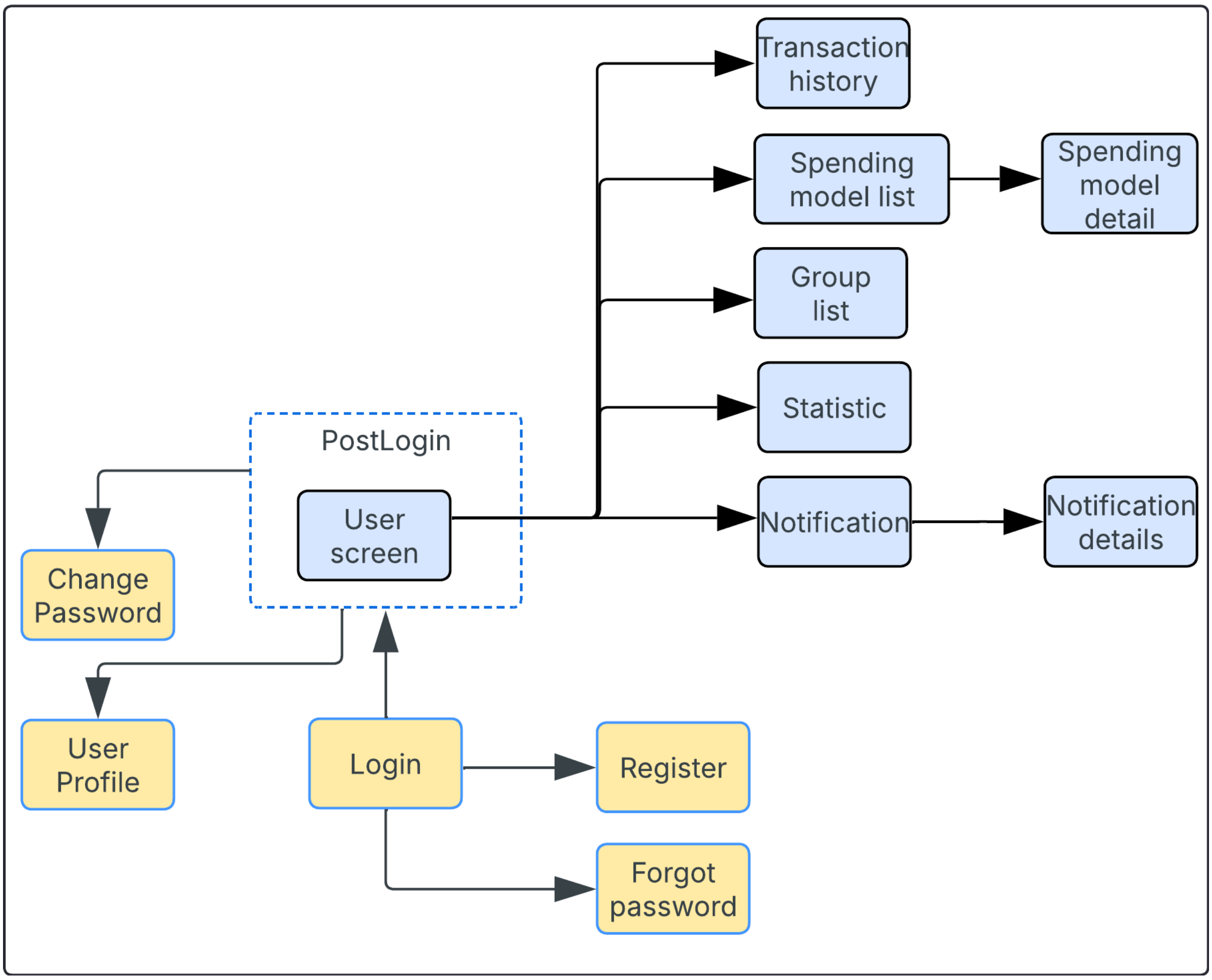


Figure 4 - Mobile module for user

3.1.1.2 Website module for admin

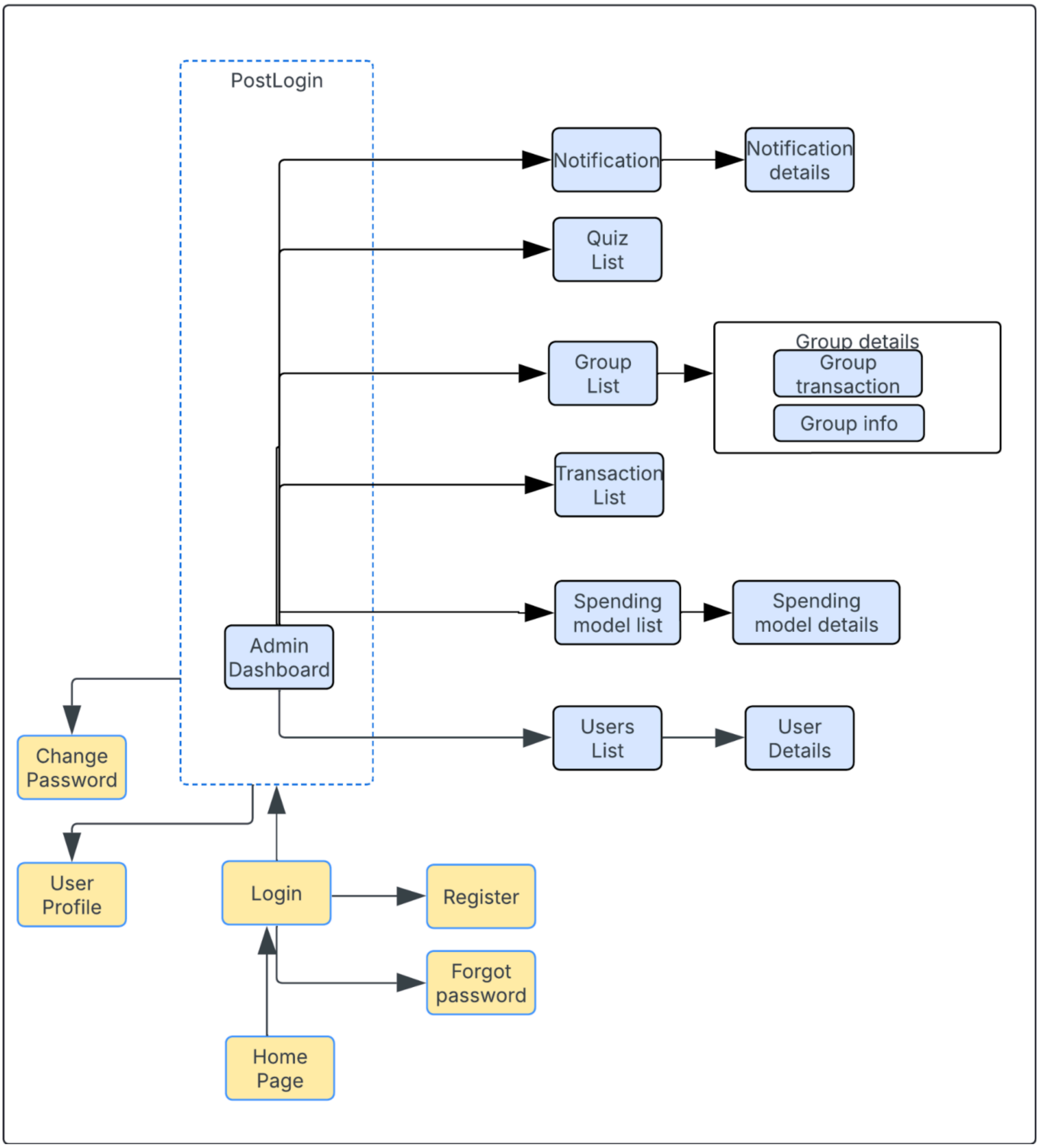


Figure 5 - Website module for admin

3.1.2 Screen Descriptions

3.1.2.1 Mobile module in user

Table 20 - Mobile module in user

#	Feature	Screen	Description

1	User Management	Login	Provides a mobile-optimized login interface for customers to access the application. Features simplified input fields for email and password, with touch-friendly design, remember me option, and quick access to forgot password functionality.
2	User Management	Register	Enables new users to create an account directly from the mobile app. Offers a streamlined registration process with minimal input fields, mobile-friendly form validation, and options to register using email or social media accounts.
3	User Management	Forgot password	Provides a mobile-specific password recovery mechanism. Allows users to reset their password by entering their registered email, receiving a reset link or verification code, and creating a new password.
4	User Management	User profile	Displays and allows editing of customer's personal information through a mobile-optimized interface. Includes options to update profile picture, contact details, preferences, and account settings
5	User Management	Change password	Offers a mobile-friendly screen for customers to change their account password. Includes current password verification, new password entry, and password strength indicators.
6	Transaction Management	Transaction history	Displays a detailed history of user transactions in a mobile-friendly format. Allows filtering and searching by date, category, and amount. Each transaction includes essential details like date, amount, category, and notes.
7	Group Management	Group list	Provides an overview of all user-created financial groups. Users can create, join, or manage groups, view member contributions, and track shared expenses in a structured format.
8	Spending model Management	Spending model list	Shows a categorized list of predefined and user-created spending models. Helps users plan

			budgets based on spending patterns, allowing easy selection and customization of spending categories.
9	Statistic Management	Statistic	Offers visual financial insights through charts and reports. Displays income vs. expenses, spending trends, and financial summaries over different time periods to help users analyze their financial habits.
10	Notification Management	Notification	Manages system and user-generated notifications. Alerts users about important updates, reminders, group activity, and financial thresholds, ensuring they stay informed about their finances.

3.1.2.2 Web module in admin

Table 21 - Web module in admin

#	Feature	Screen	Description
1	User Management	Login	Allows drivers, porters, and reviewers to access the system by entering their credentials. Provides a secure entry point with email and password fields, with options for remembering login and recovering forgotten passwords.
2	User Management	Register	Enables new users to create an account in the system. Collects essential user information such as name, contact details, email and creates a unique user profile with appropriate role assignment
3	User Management	Forgot password	Provides a mechanism for users to reset their forgotten passwords. Typically involves entering a registered email address, receiving a password reset link or temporary code, and creating a new password.
4	User Management	User profile	Displays and allows editing of user's personal information, contact details, profile picture, and account settings.

5	User Management	Change password	Allows users to modify their current password. Requires entering current password, new password, and password confirmation.
6	User Management	User list	Presents a comprehensive list of all users in the system, with filtering and sorting capabilities. Displays key user information such as name, role, contact details, and account status for administrators
7	User Management	User details	Shows detailed information about a specific user, including full profile, role, permissions, activity history, and associated records.
8	Notification Management	Notification list	Displays a list of all system notifications sent to the admin. Allows admins to view, search.
9	Notification Management	Notification details	Provides detailed information about a specific notification, including content, recipients, status (read/unread), and timestamp.
10	Quiz Management	Quiz list	Displays a list of all quizzes available in the system. Admins can view, edit, or delete existing quizzes.
11	Group Management	Group list	Provides a list of all user-created groups. Admins can view group information, the number of members, activity status, and recent activities.
12	Group Management	Group transaction	Shows a list of financial transactions within a group, including transaction amount, initiator, date, and status.
13	Group Management	Group info	Displays basic information about the group, such as group name, description, number of members, and other settings.

14	Transaction Management	Transaction list	Lists all financial transactions made by users in the system. Admins can filter transactions by date, type, and initiator.
15	Spending Model Management	Spending model list	Displays a list of spending models created by admin. Admins can add, edit, or delete these models.
16	Spending Model Management	Spending model details	Provides detailed information about a specific spending model, including spending categories, suggested budgets, and usage history.
17	User Management	Admin Dashboard	A summary dashboard for admins, providing an overview of the number of users, groups, transactions, notifications, and recent system activities.

3.1.3 Screen Authorization

3.1.3.1 Web module in admin

Table 22 - Web module in admin

Screen	Admin
User List	X
User Details	X
Spending model List	X
Spending model Details	X
Transaction List	X
Group List	X
Group Transaction	X

Group info	X
Quiz List	X
User Profile	X
Admin Dashboard	X
Settings List	X
Settings Details	X

3.1.3.1 Mobile module in user

Table 23 - Mobile module in user

Screen	Customer
Transaction List	X
Spending Model List	X
Spending Model Detail	X
Group List	X
Statistic	X
Notification	X
Notification details	X
User Profile	X

3.1.4 Non-Screen Functions

Table 24 - Non-screen functions

#	Feature	System Function	Description
01	Send notification	Send notification	The system will send notifications in the following cases: ● Create a successful spending transaction ● Create successful income transactions ● Exceed budget for a category ● Achieve a financial goal ● Track progress of a financial goal ● Request approval for a group transaction ● Successful group transaction ● Complete a financial goal ● System notifications from administrators ● Mark all notifications as read ● Mark a specific notification as read ● Delete a notification
02	Send mail	Send mail	The system will send mail in the following cases: ● Forget password ● Register new account ● Verify email ● Reset password confirmation ● Invitation to join a group

3.1.5 Entity Relationship Diagram

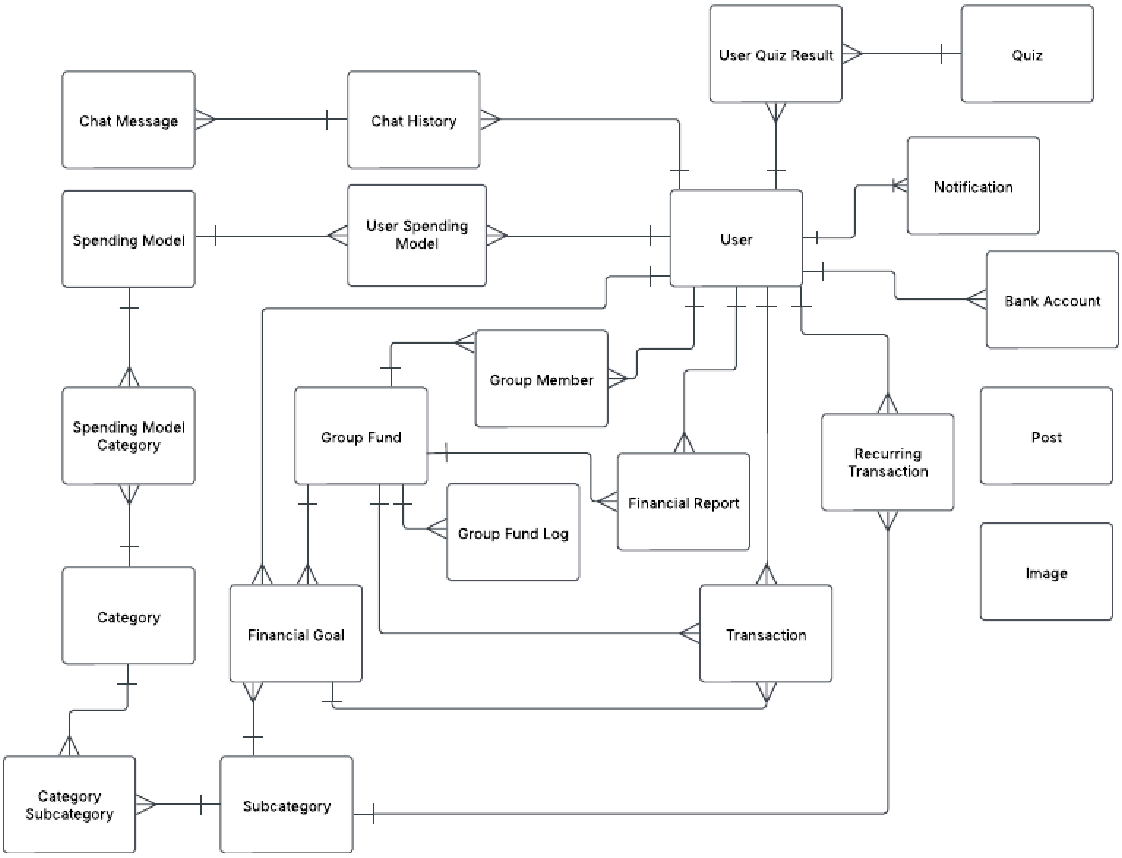


Figure 7 - Entity Relationship Diagram

Entities Description

Table 25 - Entities Description

#	Entity	Description
1	User	User account information, including personal details and authentication data.
2	Notification	Stores notifications sent to users, such as system alerts, order updates, reminders, messages, etc.
3	Bank Account	Contains users' linked bank accounts for tracking transactions and balances.
4	Recurring Transaction	Manages automated transactions that repeat on a scheduled basis, like rent or subscriptions.
5	Transaction	Stores individual expense or income records input by users for financial tracking.
6	Post	Stores information about posts created by users. Each post could represent a blog entry, a social media post, an announcement, or any form of content users generate.
7	Financial Report	Generates summaries and insights on user spending habits and financial trends.

8	Group Member	Manages user participation in spending groups, including roles and permissions.
9	Group Fund	Tracks shared financial resources within a group for collective spending and savings.
10	Group Fund Log	Record all transactions and updates related to the group's shared fund.
11	Financial Goal	User-defined savings or budget targets to track financial progress.
12	Subcategory	Represents a detailed classification of expenses within a broader spending category.
13	Category Subcategory	Defines many-to-many relationships between categories and subcategories.
14	Category	Represents major spending classifications, such as food, rent, or entertainment.
15	Spending Model Category	Links spending models to specific categories for structured budgeting.
16	Spending Model	Defines a financial management approach to guide user spending and budgeting.
17	User Spending Model	The spending model is selected or recommended for a specific user.
18	Quiz	Contains financial assessment quizzes that help users determine a suitable spending model.
19	User Quiz Result	Users' quiz responses and results for spending model recommendations.
20	Chat History	Logs conversations between users and the system for support or financial guidance.
21	Chat Message	Stores individual messages exchanged in chat conversations.
22	Image	Stores images uploaded by users, such as receipts or proof of transactions.

3.2 User Management Feature

3.2.1 Login by account

Function Trigger:

The user opens the application or web portal and selects the Login option.

Function Description:

Actor: User

Purpose: Allows users to securely access the system by entering their account credentials.

Preconditions: PRE-01, PRE-02

Post-conditions: POST-01

Function Details:

Data:

Credentials:

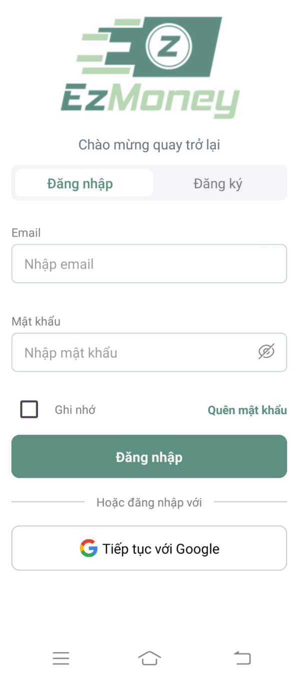
- Email (Text field)
- Password (Password input field)

Optional Features:

- Remember Me: Checkbox to save login credentials for future sessions.
- Forgot Password: Link for password recovery

Business Rules: BR-01, BR-02, BR-06

<u>Screen layout</u>



<u>Functionality</u>

Normal Cases:

The user navigates to the login screen.

The user enters their email and password.

The system validates the input:

- Checks if required fields are completed.
- Verifies email format and password complexity.

The system authenticates the credentials against the database.

If authentication succeeds:

- The system redirects the user to their role-specific dashboard:
  - User: Service browsing and booking page.

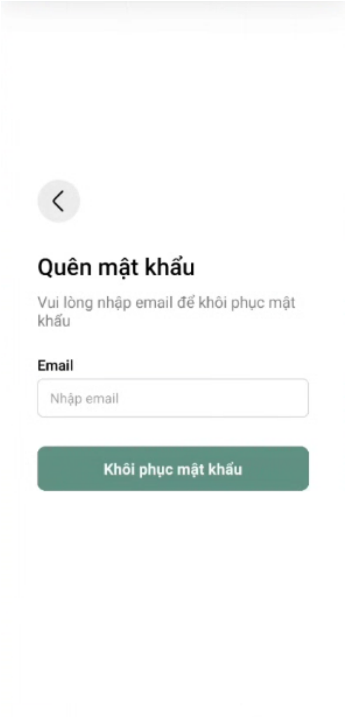
3.2.2 Forgot password

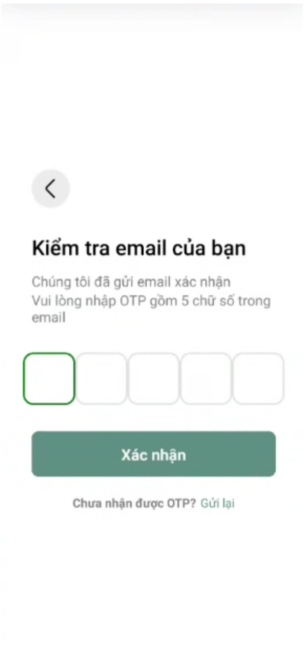
<u>Function Trigger:</u>

The user selects the Forgot Password option on the login screen.

<u>Function Description:</u>

Actor: Customer
Purpose: Allows users to reset their password securely via OTP sent to their registered email.
Preconditions: PRE-01
Post-conditions: POST-03
Function Details:
Data:
Email: Input field for registered email.
OTP (One-Time Password): Verification code sent via email to the provided email.
New Password: Input field for the new password.
Business Rules: BR-01, BR-06
Screen layout





Functionality

Normal Cases:

Step 1:

The user enters their registered email and clicks Send OTP.

The system verifies the email exists and sends an OTP via mail.

A message displays: "Mã OTP đã được gửi qua email."

Step 2:

The user enters the OTP and clicks Verify OTP.

The system validates the OTP.

If successful, the user is directed to the password reset screen.

A message displays: "Đã xác nhận OTP. Hãy đặt mật khẩu mới."

Step 3:

The user enters a new password, confirms it, and clicks Reset Password.

The system validates the password and updates it in the database.

A success message displays: "Mật khẩu đã được đặt lại thành công!"

Step 4:

The user is redirected to the login screen.

3.2.3 Login by Google

Function Trigger:

The user selects the Login with Google option on the login screen.

Function Description:

Actor: Customer

Purpose: To allow customers to log in using their Google account credentials, streamlining the authentication process.

Preconditions: PRE-03, PRE-04

Post-conditions: POST-05

Function Details:

Data:

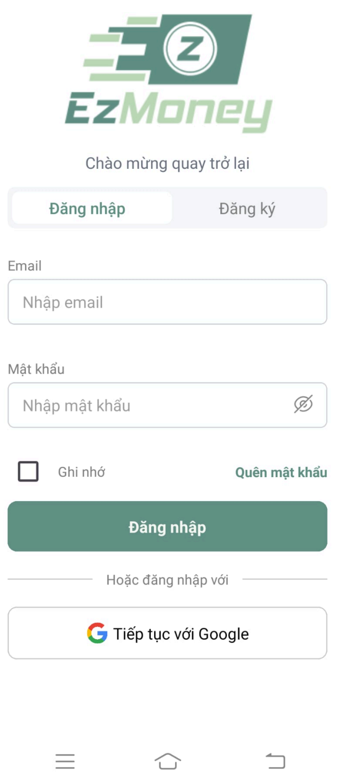
Google Account Email: Retrieved from Google’s OAuth 2.0 authentication process.

Phone Number: Input provided by the user during the first login.

Verification Code (OTP): Sent to the provided phone number for validation.

Business Rules: BR-07, BR-08

Screen layout



<u>Functionality</u>

Normal Cases:

Step 1:

The user clicks Login with Google.

The system redirects to Google’s OAuth 2.0 consent screen.

Step 2:

The user logs into their Google account and grants the necessary permissions.

Google sends an authentication token to the system.

Step 3:

The system validates the token using Google’s API.

If the token is valid:

1. The system checks if the Google email is linked to an existing customer account.
2. If linked, the user is redirected to their dashboard.

Step 4 (First Login):

The user enters their email.

The system sends an OTP to the provided email.

The user enters the OTP for verification.

If the OTP is valid, the system registers the email and completes the account setup.

Step 5:

The user is redirected to the home page.

3.2.4 Register account

Function Trigger:

The user accesses the registration screen and chooses to register as a guest.

Function Description:

Actor: Guest

Purpose: To allow new users to create a guest account to browse services, create bookings, or explore the application.

Preconditions: PRE-01

Post-conditions: POST-01

Function Details:

Data:

Name: The user's full name.

Email (optional): Used for verification and future upgrades to a full customer account.

Phone Number: Required for OTP-based verification.

Password: Secure input for account protection.

Verification Code (OTP): Sent to the provided email for validation.

Business Rules: BR-01, BR-02, BR-06

Screen layout

Functionality

Normal Cases:

Step 1:

The guest accesses the registration screen and enters their full name, phone number, and password.

The system validates the inputs.

Step 2:

The system sends an OTP to the provided email.

Step 3:

The guest enters the OTP on the verification screen.

The system validates the OTP.

Step 4:

If the OTP is valid, the system creates the guest account and stores the information in the database.

Step 5:

The guest is redirected to the dashboard or service browsing screen.

3.2.5 Update profile

Function Trigger:

The user selects the option to update their profile from the account settings

Function Description:

Actor: User

Purpose: To allow customers to modify their personal information, ensuring the profile details remain accurate and up-to-date

Preconditions: PRE-02

Post-conditions: POST-53, POST-02, POST-06

Function Details:

Data:

Full Name (Text field)

Email Address (Text field, must be unique and valid)

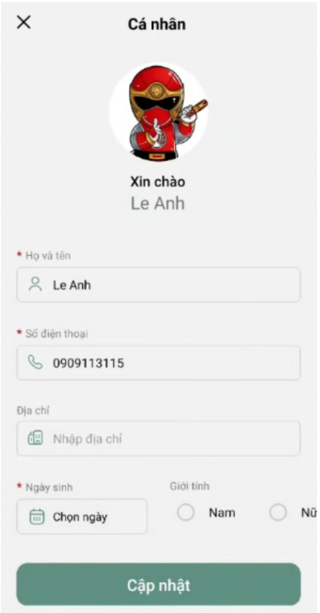
Phone Number (Numeric field, must be unique and valid)

Address (Text field)

Profile Picture (Image upload)

Business Rules: BR-05, BR-06, BR-09, BR-22, BR-23, BR-24

Screen layout



Functionality

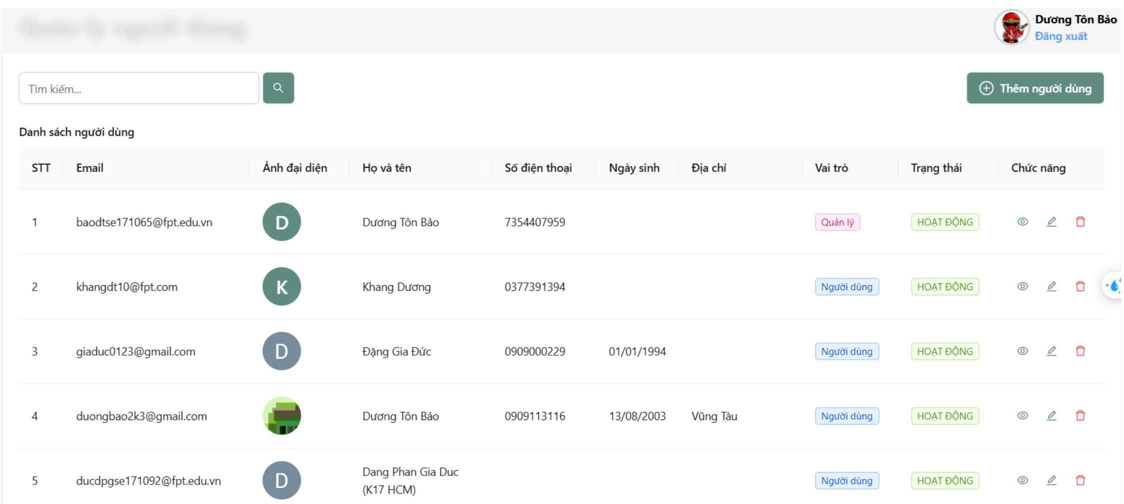
Normal Cases:
The user navigates to the profile management section.
The user edits their profile details using the provided input fields.
The system validates the inputs:
Ensures required fields are not empty.
Checks for unique email and phone numbers.
Validates the format of the uploaded profile picture.
The user touch save to submit the updates.
The system saves the changes and displays a success message: "Profile updated successfully."

3.2.6 View list user

Function Trigger:
The admin navigates to the customer management section to view the list of registered customers.
Function Description:
Actor: Admin
Purpose: To enable the admin to view and manage the list of all registered customers in the system.
Preconditions: PRE-25
Post-conditions: POST-53
Function Details:
Data:

Customer ID: Unique identifier for each customer.
Name: Full name of the customer.
Email: Email address (if provided).
Phone Number: Registered email.
Status: Indicates whether the account is active, suspended, or deleted.
Registration Date: Date when the customer registered.
Number of Bookings: Total bookings made by the customer.

Business Rules: BR-121
Screen layout



Functionality

Normal Cases:

Step 1:

The admin logs into the system and navigates to the customer management module.

Step 2:

The system displays a table with the list of customers and their details.

Step 3:

The admin can apply filters or search by specific criteria (e.g., phone number, name, status).

Step 4:

The admin selects a customer to view additional details or perform actions like editing status or updating information.

3.3 Group Management Feature

3.3.1 Create group

Function Trigger:

The user selects the option to create a new group from the group management section.

Function Description:

Actor: User

Purpose: To allow users to create a group for managing shared expenses, enabling collaborative financial tracking and goal setting.

Preconditions: PRE-10

Post-conditions: POST-65

Function Details:

Data:

Group Name (Text field, required)

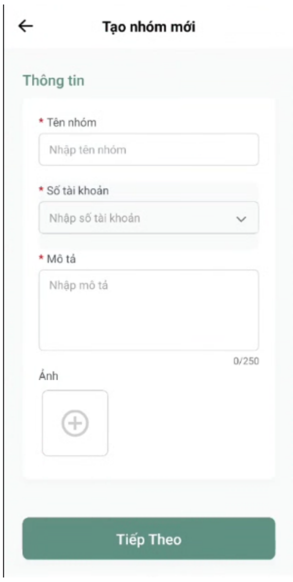
Bank Account Number (12 digits, required)

Group Description (Text field, required)

Profile Picture (Image upload, optional)

Business Rules: BR-10, BR-11, BR-12, BR-13, BR-14

Screen Layout:



<u>Functionality</u>:

Normal Cases:

Step 1:

The user navigates to the group management section and selects "Tạo nhóm."

Step 2:

The user fills in the group details using the provided input fields.

Step 3:

The system validates the inputs:

Ensures required fields (e.g., Group Name) are not empty.

Validates the format of the uploaded profile picture.

Step 4:

The user touches "Tiếp theo" to submit the group details.

Step 5:

The system saves the new group, assigns the creator as the admin, and displays a success message: "Tạo nhóm thành công."

3.3.2 Update Group

<u>Function Trigger:</u>

The user selects the option to update an existing group from the group management section.

<u>Function Description:</u>

Actor: User

Purpose: To allow users to update group details for managing shared expenses, enabling collaborative financial tracking and goal setting.

Preconditions: PRE-02, PRE-06, PRE-07

Post-conditions: POST-07

<u>Function Details:</u>

Data:

Group Name (Text field, required)

Bank Account Number (12 digits, required)

Group Description (Text field, required)

Profile Picture (Image upload, optional)

Business Rules: BR-10, BR-13, BR-15, BR-16, BR-17

Screen Layout:

Functionality:

Normal Cases:

Step 1:

The user navigates to the group management section and selects "Cập nhật nhóm."

Step 2:

The user fills in the updated group details using the provided input fields.

Step 3:

The system validates the inputs:

Ensures required fields (e.g., Group Name) are not empty.

Validates the format of the uploaded profile picture.

Step 4:

The user touches "Tiếp theo" to submit the updated group details.

Step 5:

The system saves the updated group information and displays a success message: "Cập nhật nhóm thành công."

3.3.3 Disband Group

Function Trigger:

The user selects the option to disband an existing group from the group management section.

Function Description:

Actor: User

Purpose: To allow users to disband an existing group for managing shared expenses, effectively ending collaborative financial tracking and goal setting.

Preconditions: PRE-02, PRE-06, PRE-07, PRE-08

Post-conditions: POST-08, POST-09, POST-10

Function Details:

Data:

Group ID (Hidden, required): Unique identifier for the group being disbanded.

Business Rules: BR-10, BR-16, BR-18, BR-19, BR-20

Screen Layout:

Confirmation Dialog: "Bạn có chắc muốn giải tán nhóm không ?"

Functionality:

Normal Cases:

Step 1:

Navigation: The user goes to the group management section and selects "Giải tán."

Step 2:

Confirmation: A dialog appears asking for confirmation to disband the group.

Step 3:

User Confirmation: The user confirms the action.

Step 4:

System Process:

The system removes the group from the database.

Sends notifications to all group members about the disbanding.

Step 5:

Success Message: A message appears: "Nhóm đã được giải tán thành công."

3.3.4 Leave Group

Function Trigger:

The user selects the option to leave a group from the group management section.

Function Description:

Actor: User

Purpose: To allow users to exit a group for managing shared expenses.

Preconditions: PRE-02, PRE-06, PRE-09, PRE-10

Post-conditions: POST-11, POST-12, POST-13, POST-14

Function Details:

Data:

Group ID (Identifier for the group, required)

Business Rules: BR-21, BR-22, BR-23, BR-24, BR-25

Screen Layout:

Functionality:

Normal Cases:

Step 1:

The user navigates to the group management section and selects "Rời nhóm."

Step 2:

The system prompts the user for confirmation to leave the group.

Step 3:

The user confirms their decision to leave.

Step 4:

The system removes the user from the group and displays a success message: "Bạn đã rời khỏi nhóm thành công."

3.3.5 Request a withdrawal

<u>Function Trigger:</u>

User selects the option to add a transaction within a specific group from the group management section.

<u>Function Description:</u>

Actor: User

Purpose: Enable collaborative financial tracking and expense management within shared groups.

Preconditions: PRE-02, PRE-06, PRE-10

Post-conditions: POST-66, POST-67

<u>Function Details:</u>

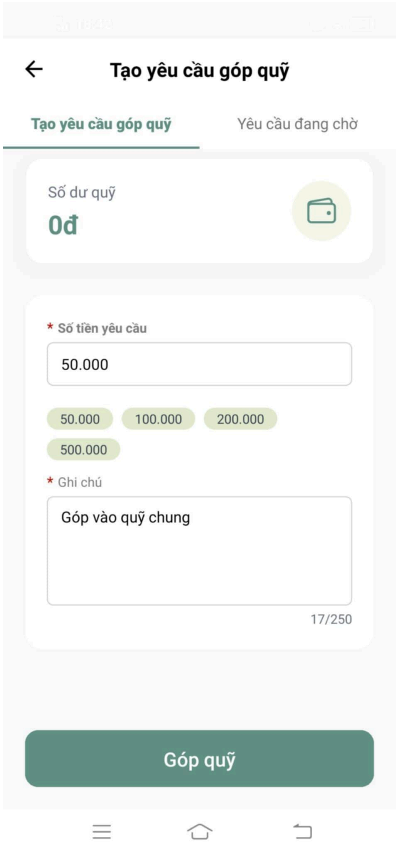
Data:

Transaction Amount: Numeric field, required

Transaction Description: Text field, required

Business Rules: BR-93

<u>Screen Layout:</u>



<u>Functionality:</u>

Normal Cases:

Step 1:

User navigates to the group management section and selects "Góp quỹ"

Step 2:

User fills in the transaction details using provided input fields.

Step 3:

System validates inputs:

Ensures required fields (Transaction Amount, Description) are not empty.

Step 4:

User touches "Góp quỹ" to add the transaction.

Step 5:

System saves the transaction, adds it to the group's expense list, and displays a success message: "Góp quỹ thành công."

3.3.6 Raising funds request

<u>Function Trigger:</u>

The user selects the option to raise funds within a specific group from the group management section.

<u>Function Description:</u>

Actor: User

Purpose: To allow users to raise funds in shared groups, enabling collaborative financial tracking and contribution management.

Preconditions: PRE-02, PRE-06, PRE-10

Post-conditions: POST-66, POST-67

<u>Function Details:</u>

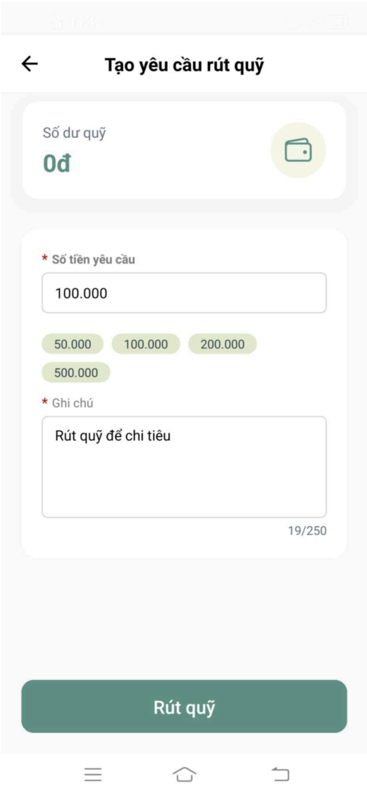
Data:

Transaction Amount (Numeric field, required)

Transaction Description (Text field, required)

Business Rules: BR-93

<u>Screen Layout:</u>



<u>Functionality:</u>

Normal Cases:

Step 1:

The user navigates to the group management section and selects "Rút quỹ"

Step 2:

The user fills in the transaction details using the provided input fields.

Step 3:

The system validates the inputs:

Ensuring required fields (Transaction Amount, Description) are not empty.

Step 4:

The user touches "Rút quỹ" to raise the funds.

Step 5:

The system saves the transaction, adds it to the group's financial records, and displays a success message: "Tạo giao dịch thành công."

3.3.7 Remind raising fund

<u>Function Trigger</u>

Action: The user selects the option to add a transaction within a specific group from the group management section.

<u>Function Description</u>

Actor: User

Purpose: To enable users to manage group transactions for collaborative financial tracking.

Preconditions: PRE-02, PRE-06, PRE-10

Post-conditions: POST-68

<u>Function Details</u>

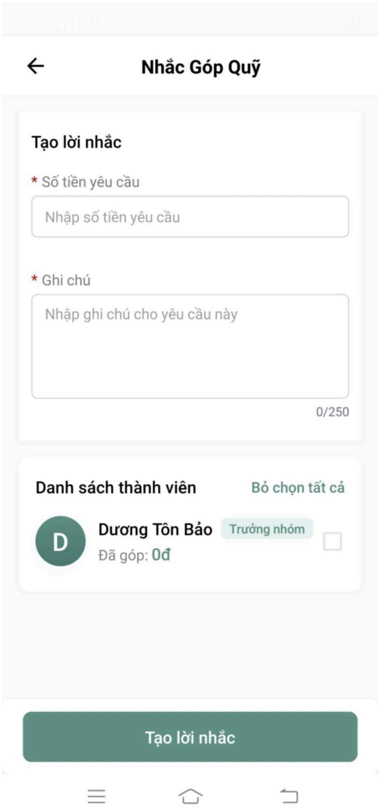
Data:

Transaction Amount: Numeric field, required

Transaction Description: Text field, required

Business Rules: BR-93

<u>Screen Layout:</u>



Functionality:

Normal Cases:

Step 1:

User navigates to the group management section and selects "Nhắc góp quỹ"

Step 2:

User fills in transaction details using the provided fields.

Step 3:

System validates inputs ensuring required fields (Transaction Amount, Description) are filled.

Step 4:

User taps "Tạo lời nhắc" to add the transaction.

Step 5:

System saves the transaction, adds it to the group's expense list, and displays a success message: "Tạo lời nhắc thành công."

3.3.8 Set goal amount

Function Trigger:

The user selects the option to set a financial goal for a specific group in the group management section.

Function Description:

Actor: User

Purpose: To allow users to establish financial goals for expenses or income within shared groups, enhancing financial planning and accountability.

Preconditions: PRE-02, PRE-06, PRE-07, PRE-15

Post-conditions: POST-23, POST-24

Function Details:

Data:

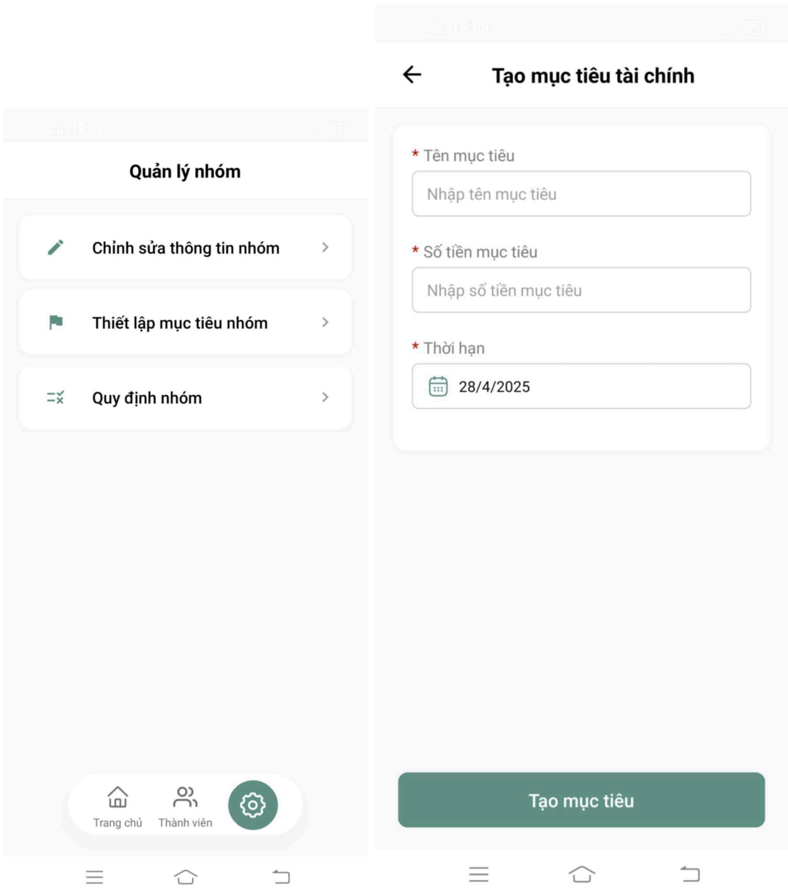
Amount for the goal

Description of the goal (e.g., "Save for vacation")

Current Date

Business Rules: BR-15, BR-16, BR-39, BR-40, BR-121

Screen Layout:



Functionality:

Normal Cases:

Step 1:

The user navigates to the group management section and selects "Thiết lập mục tiêu nhóm".

Step 2:

The user enters the goal description, amount and date.

Step 3:

The system saves the goals and confirms successful entry.

Step 4:

User taps "Tạo mục tiêu" to set a goal.

Step 5:

Displays a success message: "Tạo mục tiêu thành công."

3.3.9 Kick member

Function Trigger:

The user selects the option to kick a member from a specific group in the group management section.

Function Description:

Actor: User

Purpose: To allow users to remove members from a group, ensuring better management and accountability.

Preconditions: PRE-06, PRE-10

Post-conditions: POST-27, POST-28

Function Details:

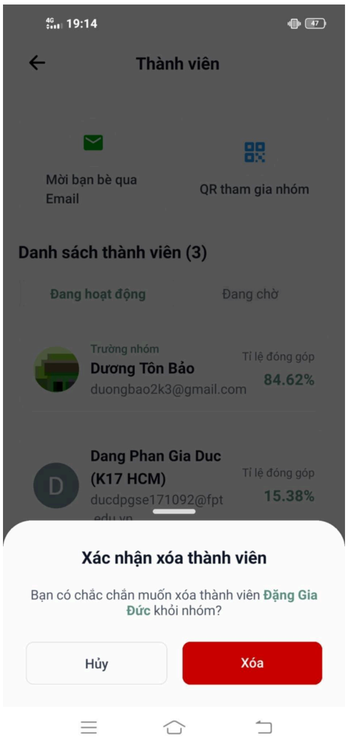
Data:

Member to be removed

Reason for removal (optional)

Business Rules: BR-15, BR-45, BR-46

Screen Layout:



Functionality:

Normal Cases:

Step 1:

The user navigates to the group management section and selects "Xóa thành viên."

Step 2:

The user selects the member to be removed.

Step 3:

The system processes the removal and confirms the action.

Step 4:

All group members will be notified of the member's removal (if applicable).

Step 5:

The kicked member will be removed from the group list and functionalities.

3.3.10 View statistic group

Function Trigger:

The user selects the option to view statistics for a specific group in the group management section.

Function Description:

Actor: User

Purpose: To allow users to view key statistics and metrics related to the group's activities and membership.

Preconditions: PRE-02, PRE-06, PRE-10

Post-conditions: POST-29, POST-30

Function Details:

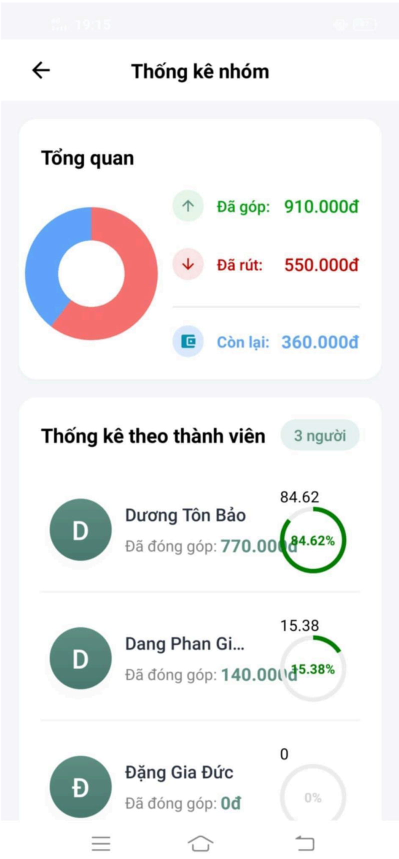
Data:

Group activity metrics

Member participation data

Business Rules: BR-47

Screen Layout:



Functionality:

Normal Cases:

Step 1:

The user navigates to the group management section and selects "Thống kê."

Step 2:

The system processes the request and displays relevant group statistics.

Step 3:

The user can explore detailed metrics related to group activities and member engagement.

3.3.11 Invite member by QR

Function Trigger:

The user selects the option to invite a new member to the group.

Function Description:

Actor: User

Purpose: To allow users to add new members to a group by scanning a QR code.

Preconditions: PRE-02, PRE-10

Post-conditions: POST-31, POST-32, POST-33

Function Details:

Data:

Member Name (Input field)

QR Code (Scan field)

Role in Group (Dropdown: Admin, Member, etc.)

Invitation Status (Pending/Accepted)

Business Rules: BR-48, BR-49

Screen Layout:

Functionality:

Normal Cases:

Step 1:

The user selects the option to invite a new member.

Step 2:

The system prompts the user to scan the new member's QR code.

Step 3:

After scanning, the system retrieves member details from the QR code.

Step 4:

The user submits the form.

Step 5:

The system checks if the member details are valid and unique.

If valid, the system sends an invitation to the new member.

If the details are already registered, the system shows an error.

Step 6:

The member is added, and the system updates the member list in real-time.

3.3.12 View list members in group

<u>Function Trigger:</u>

The user selects the option to view group members from the group management section.

<u>Function Description:</u>

Actor: User

Purpose: To allow users to view the list of members in a group, including their roles and contribution percentages.

Preconditions: PRE-02, PRE-06- PRE-10

Post-conditions: POST-34, POST-35

<u>Function Details:</u>

Data:

Member Name (Displayed as text)

Email Address (Displayed as text)

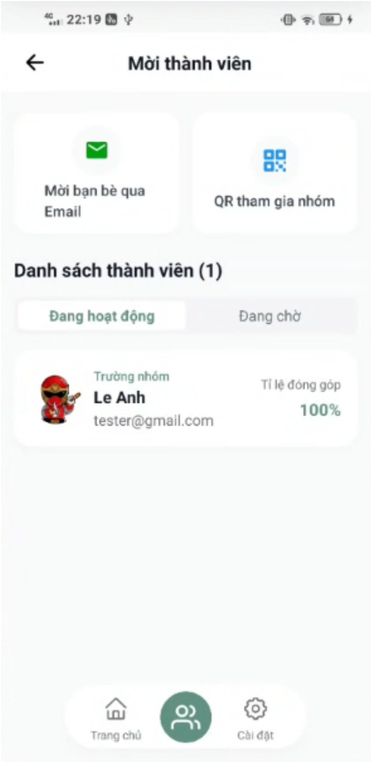
Role in Group (E.g., Admin, Member)

Contribution Percentage (Displayed as a percentage)

Status (Active/Pending)

Business Rules: BR-15, BR-50, BR-51

<u>Screen Layout:</u>



A list displaying all group members with their profile pictures, names, roles, and contribution percentages.

Tabs for filtering between Đang hoạt động and Đang chờ members.

Options to invite new members or remove existing ones (if the user has permissions).

<u>Functionality:</u>

Normal Cases:

Step 1:

The user navigates to the group management section and selects "Thành Viên"

Step 2:

The system retrieves and displays the list of members with their details.

Step 3:

The user can:

View detailed information of each member.

Switch between Đang hoạt động and Đang chờ members.

Invite new members or remove existing ones (if authorized).

Step 4:

The system updates the list in real-time if changes occur.

3.3.13 Invite member by email

<u>Function Trigger:</u>

The user selects the option to invite a new member to the group.

<u>Function Description:</u>

Actor: User

Purpose: To allow users to add new members to a group by entering their email address or selecting them from existing contacts.

Preconditions: PRE-02, PRE-10
Post-conditions: POST-36, POST-37, POST-38
<u>Function Details:</u>
Data:
Member Name (Input field)
Email Address (Input field)
Role in Group (Dropdown: Admin, Member, etc.)
Invitation Status (Pending/Accepted)
Business Rules: BR-48, BR-49
<u>Screen Layout:</u>

A form for entering new member details:
Email Address field.
Plus button to send the invitation.
Functionality:
Normal Cases:
Step 1:
The user selects the option to invite a new member.
Step 2:
The system prompts the user to enter the new member's email address.
Step 3:
The user submits the form.
Step 4:
The system checks if the email address is valid and unique.
If the email address is valid, the system sends an invitation to the user.
If the email address is already registered, the system shows an error.
Step 5:
The member is added, and the system updates the member list in real-time.

3.3.14 View list transaction

Function Trigger:

The user selects the option to view the transaction list in the group management section.

Function Description:

Actor: User

Purpose: To allow users to view the list of transactions within shared groups for better financial tracking and accountability.

Preconditions: PRE-02, PRE-10

Post-conditions: POST-39, POST-40

Function Details:

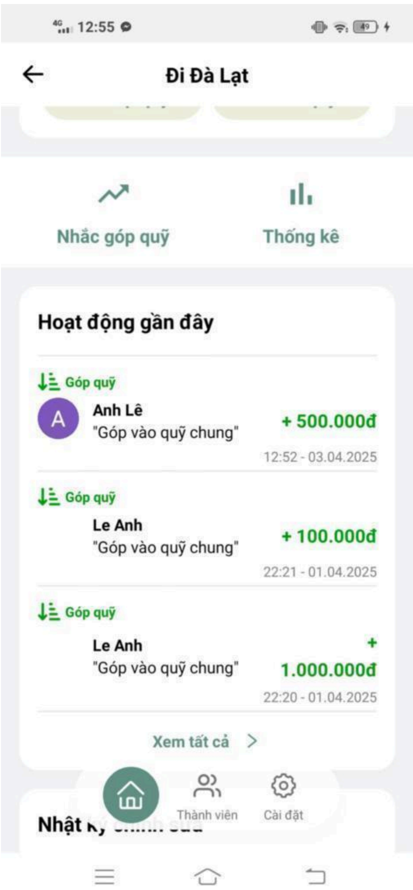
Data:

Transaction details, including:

- Amount
- Description of the transaction
- Date and time

Business Rules: BR-47

Screen Layout:



Functionality:

Normal Cases:

Step 1:

The user navigates to the group management section and selects "Xem thêm."

Step 2:

The system retrieves and displays the transactions for the selected date range.

Step 3:

Users can scroll through the list to review transaction details.

Step 4:

Users have the option to filter transactions based on criteria (e.g., date, type).

Step 5:

Users can observe total income and total expenses during the selected date range.

3.4 Bank Account Management Feature

3.4.1 Add bank account

<u>Function Trigger:</u>

User selects the "Add Bank Account" option from the main menu or relevant section.

<u>Function Description:</u>

To allow users to add and manage their bank account information in the app.

Preconditions: PRE-02

Post-conditions: POST-41, POST-42

POST-80: User successfully adds a bank account, and the information is saved in the system.

<u>Function Details:</u>

Data:

Account Number: User inputs their bank account number.

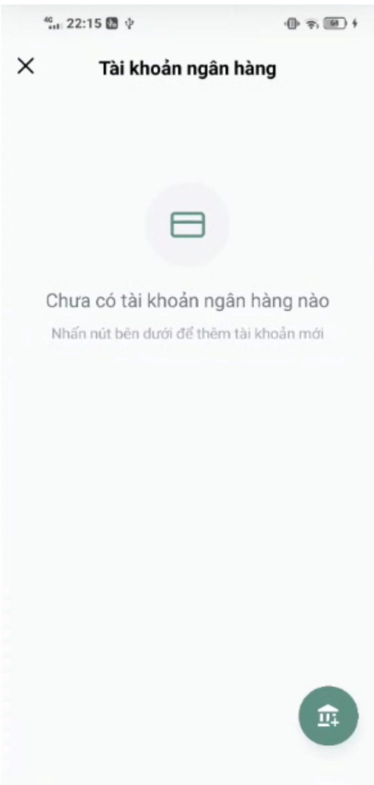
Bank Name: User selects or inputs the name of the bank.

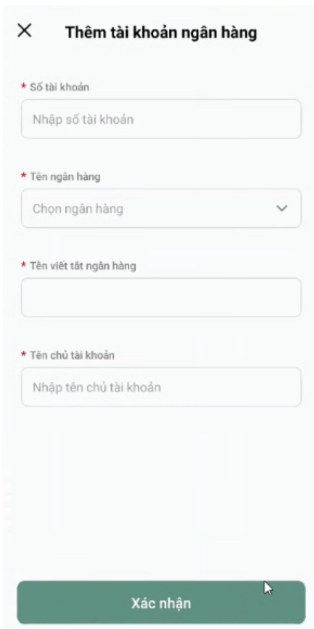
Bank Abbreviation: Enter the abbreviated name of the bank.

Account Holder Name: User inputs the name of the account holder.

Business Rules: BR-52, BR-53, BR-54, BR-55

<u>Screen Layout:</u>





Step-by-Step Display:
Progress Indicator: Indicates the saving process for the bank account.
Input Data Section:
Account Number: Text field for inputting the number.
Bank Name: Dropdown menu to select the bank.
Bank Abbreviation: Text field for entering the abbreviation.
Account Holder Name: Text field for inputting the holder's name.
Submit Action:
Submit Button: To confirm and save the bank account details.

Functionality:

Normal Cases:
Step 1:
User selects the "Thêm ngân hàng" option.
Step 2:
System displays the input form.
Step 3:
User fills in the details and submits.
Step 4:
System saves the account and displays a confirmation.

3.4.2 Edit bank account

Function Trigger:

User selects the "Edit Bank Account" option from the main menu or the relevant section.

Function Description:

To allow users to modify their existing bank account information in the app.
Preconditions: PRE-02, PRE-18
Post-conditions: POST-43

Function Details:

Data:
Account Number: User inputs the bank account number to be edited.
Bank Name: User selects or inputs the name of the bank.
Bank Abbreviation: User provides the abbreviated name of the bank.
Account Holder Name: User inputs the name of the account holder.
Business Rules: BR-56, BR-57, BR-58, BR-59

Screen Layout:

Functionality:

Normal Cases:
Step 1:
User selects the "Edit Bank Account" option.
Step 2:
System displays the input form with existing account details.
Step 3:
User updates the details and submits.
Step 4:

System saves the edited account and displays a confirmation.

3.4.3 View bank account detail

Function Trigger:

User selects the "View Bank Account" option from the main menu or the relevant section.

Function Description:

To allow users to view details of their existing bank accounts within the app.

Preconditions: PRE-02, PRE-18

Post-conditions: POST-44, POST-45

Function Details

Data:

Account Number: System fetches the bank account number for display.

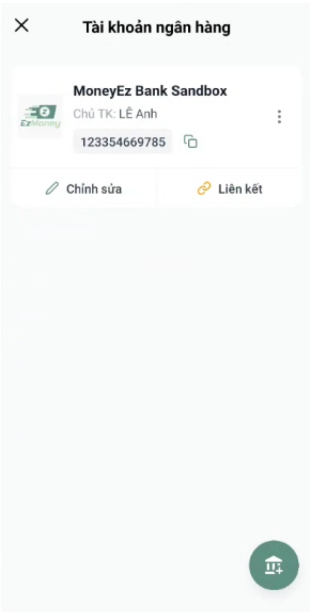
Bank Name: Displays the name of the bank associated with the account.

Bank Abbreviation: Displays the abbreviated name of the bank.

Account Holder Name: Displays the name of the account holder.

Business Rules: BR-60, BR-61, BR-62

Screen Layout:



Step-by-Step Display:

Progress Indicator: Indicates loading process for bank account details.

Display Data Section:

Account Number: Display only.

Bank Name: Display only.

Bank Abbreviation: Display only.

Account Holder Name: Display only.

Functionality:

Normal Cases:

Step 1:

User selects the "Chi tiết" option.

Step 2:

System displays the details of the selected bank account.

Step 3:

User can choose to edit or delete the account.

3.4.4 Delete bank account

Function Trigger:

User selects the "Xóa liên kết" option from the relevant menu section.

Function Description:

To allow users to delete an existing bank account within the app.

Preconditions: PRE-02, PRE-18

Post-conditions: POST-46, POST-47

Function Details:

Data:

Account Number: System fetches the bank account number to be deleted.

Bank Name: Displays the name of the bank associated with the account being deleted.

Account Holder Name: Displays the name of the account holder.

Business Rules: BR-63

Screen Layout:

Step-by-Step Display:

Progress Indicator: Indicates loading process for the deletion request.

Display Data Section:

Account Number: Display only.

Bank Name: Display only.

Account Holder Name: Display only.

Functionality:

Normal Cases:

Step 1:

User selects the "Xóa liên kết" option.

Step 2:

System prompts for confirmation before deletion.

Step 3:

User confirms deletion, and the bank account is deleted successfully.

3.5 Statistic Management Feature

3.5.1 Set spending model

Function Trigger:

User selects the spending model option from the main menu or relevant section.

Function Description:

To help users determine a suitable consumer spending model based on their provided preferences.

Preconditions:PRE-02, PRE-19

Post-conditions: POST-48, POST-49, POST-50

POST-80: User receives a suggested spending model based on their selections.

Function Details:

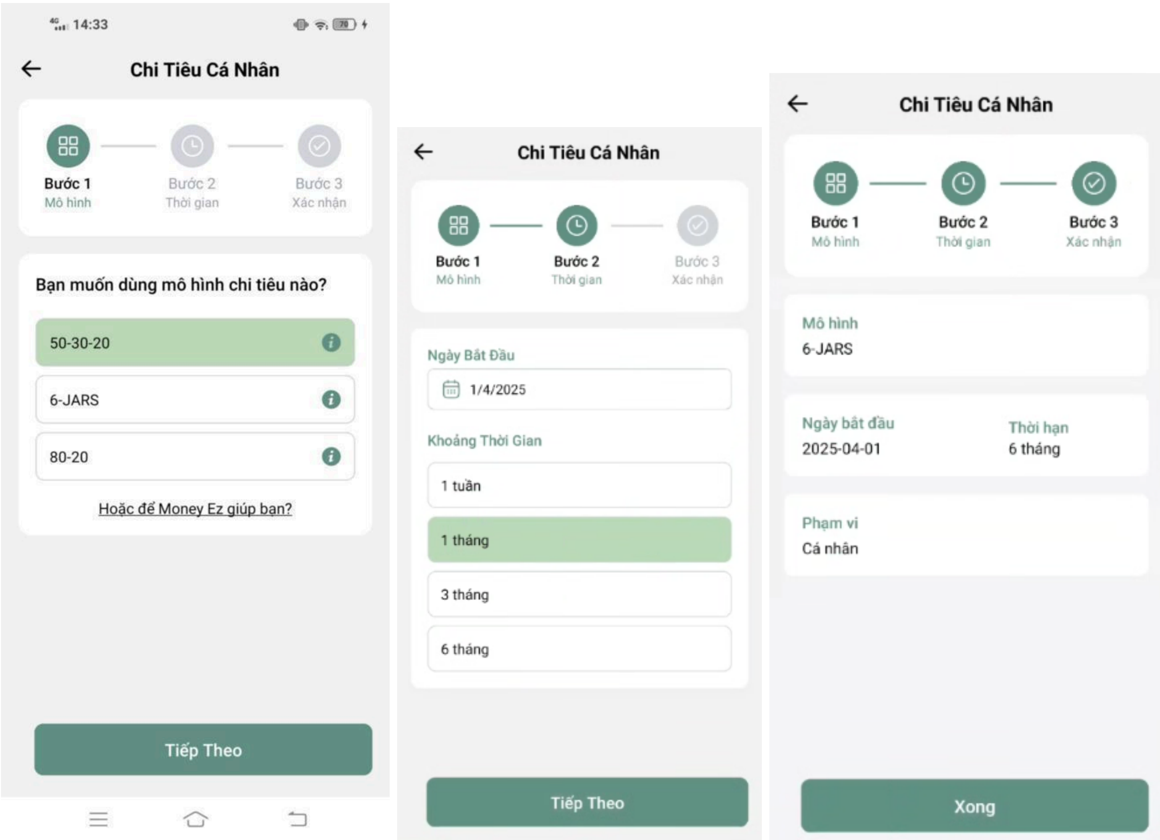
Data:

Question: Displayed as text.

Answer Choices: Multiple-choice options presented relevant to spending habits.
User Selection: Captured for processing results.
Suggested Spending Model: Displayed after selection completion.

Business Rules: BR-64, BR-65, BR-66

Screen Layout:



Step-by-Step Display:

Progress Indicator: To show the selection process.

Multiple-choice Answer Selection: Users can choose their preferred spending habit.

Input Data Section:

Start Date: User inputs the start date, e.g., “2/4/2025.”

Time Frame Selection:

Options:

1 tuần (1 week)

1 tháng (1 month) *(selected option highlighted)*

3 tháng (3 months)

6 tháng (6 months)

<u>Functionality:</u>

Normal Cases:

Step 1:

User selects the spending model option.

Step 2:

System displays the available choices.

Step 3:

User selects an option and submits.

Step 4:

After analysis, the system provides a suggested spending model.

Step 5:

User can review results and select again if desired.

3.5.2 View report statistic

<u>Function Trigger:</u>

User selects the report statistic option from the main menu or relevant section.

<u>Function Description:</u>

To provide users with an overview of their income and expenditure over a selected time frame.

Preconditions: PRE-02, PRE-19, PRE-20

Post-conditions: POST-51, POST-52

<u>Function Details:</u>

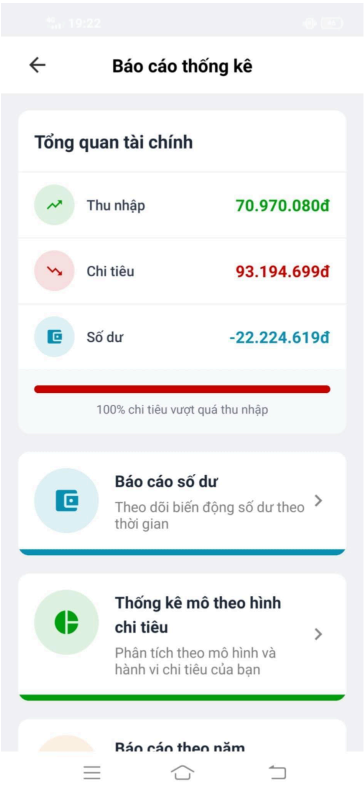
Data:

Selected Date: User specifies which date to view statistics for.

Income and Expenditure Data: Displayed in a clear format.

Business Rules: BR-67, BR-68, BR-69

<u>Screen Layout:</u>



<u>Functionality:</u>

Normal Cases:

Step 1:

User selects the "Thống kê" option.

Step 2:

System displays available reporting options.

Step 3:

User selects a date and submits.

Step 4:

System provides a detailed report based on the criteria.

3.5.3 Set spending limit

<u>Function Trigger:</u>

User selects the "Set Spending Limit" option from the main menu or the relevant section.

<u>Function Description:</u>

To allow users to set spending limits on their categories within the app.

Preconditions: PRE-02, PRE-21

Post-conditions: POST-54, POST-55

<u>Function Details:</u>

Data:

Category Name: The system displays the name of the category for which the limit is to be set.

Maximum Amount: Input field for users to specify the maximum spending limit.

Business Rules: BR-70, BR-71

<u>Screen Layout:</u>

Step-by-Step Display:

Progress Indicator: Indicates the loading process for setting the spending limit.

Display Data Section:

Category Name: Display only.

Maximum Amount: Input field for user entry.

<u>Functionality:</u>

Normal Cases:

Step 1:

User selects the "Đặt hạn mức chi tiêu" option.

Step 2:

System prompts for the category and amount.

Step 3:

User enters the maximum amount and confirms.

Step 4:

System saves and confirms the spending limit successfully.

3.5.4 Set spending goal

Function Trigger:

User selects the planned spending option from the main menu or relevant section.

Function Description:

To assist users in establishing a planned consumer spending model based on their preferences.

Preconditions: PRE-02, PRE-22

Post-conditions: POST-56, POST-57

Function Details:

Data:

Question: Displayed as text prompting user engagement.

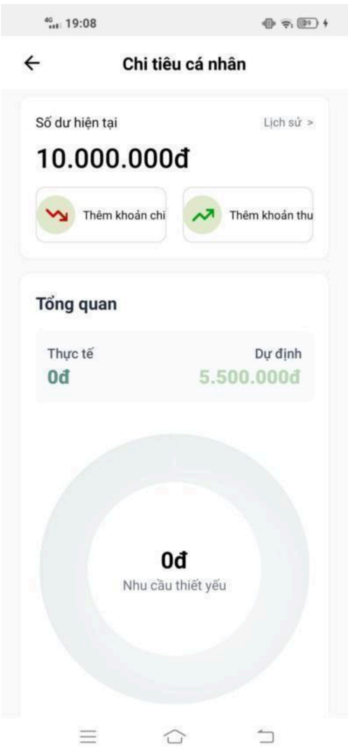
Answer Choices: Multiple-choice options related to financial planning.

User Selection: Captured for further analysis.

Suggested Planned Spending Model: Presented after selections are made.

Business Rules: BR-72, BR-73, BR-74

Screen Layout:



Step-by-Step Display:

Progress Indicator: Shows the user's advancement through the selection process.
Multiple-choice Answer Selection: Users choose their preferred spending plan.
Input Data Section:
Start Date: Users enter the start date, e.g., “2/4/2025.”
Time Frame Selection:
Options for duration:
1 tuần (1 week)
1 tháng (1 month) (selected option highlighted)
3 tháng (3 months)
6 tháng (6 months)

Functionality:

Normal Cases:

Step 1:
User initiates the planned spending option.

Step 2:
System presents available choices.

Step 3:
User selects an option and submits.

Step 4:
After analysis, the system provides a suggested planned spending model.

Step 5:
User can review and make new selections if desired.

3.5.5 Quiz

Function Trigger:

The user selects the quiz option from the main menu or relevant section.

Function Description:

Actor: User

Purpose: To allow users to answer multiple-choice questions that help determine a suitable consumer model based on their responses.

Preconditions: PRE-02, PRE-23, PRE-24

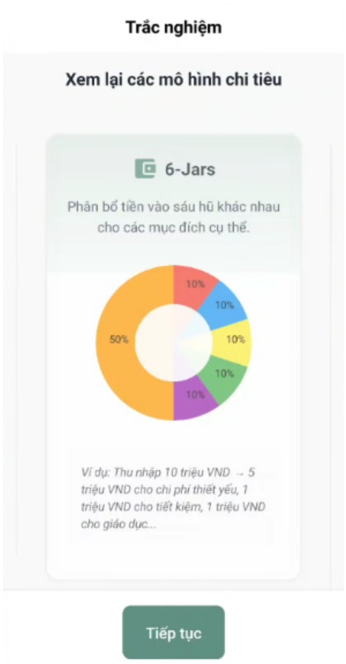
Post-conditions: POST-58, POST-59

Function Details:

Data:
Question (Displayed as text)
Answer Choices (Displayed as multiple-choice options)
User Selection (Stored for processing results)
Suggested Consumer Model (Displayed after quiz completion)

Business Rules: BR-75, BR-76, BR-77, BR-78

Screen Layout:



A step-by-step question display format with:

A progress indicator.

Multiple-choice answer selection.

A submit button to proceed to the next question.

A results screen displaying the suggested consumer model with an explanation.

<u>Functionality:</u>

Normal Cases:

Step 1:

The user selects the quiz option.

Step 2:

The system displays the first question with multiple-choice options.

Step 3:

The user selects an answer and proceeds to the next question.

Step 4:

After answering all questions, the system analyzes the responses.

Step 5:

The system provides a suggested consumer model based on the user's choices.

Step 6:

The user can review the result and retake the quiz if desired.

3.6 Transaction Management Feature

3.6.1 Create transaction

<u>Function Trigger:</u>

User selects the option to add a spending transaction from the main menu or relevant section.

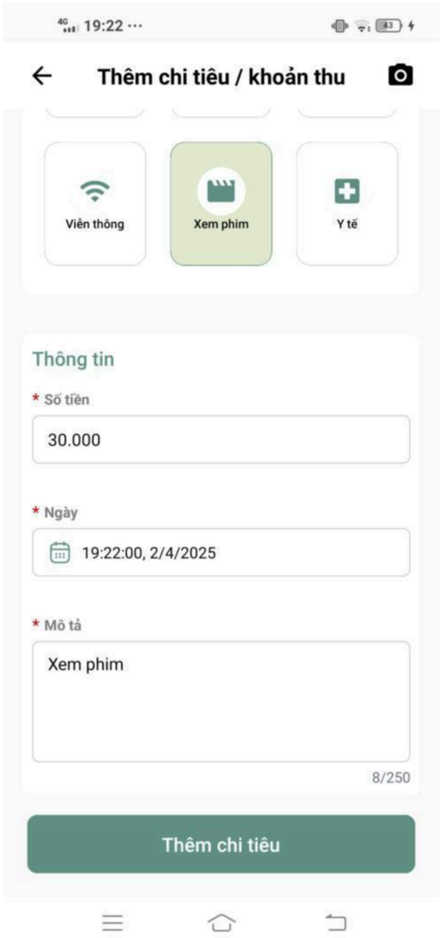
<u>Function Description:</u>

To enable users to record and categorize their spending efficiently.

Preconditions: PRE-02, PRE-26

Post-conditions: POST-60, POST-61

Function Details:
Data
Amount: The user specifies the amount of the transaction.
Date: The user specifies the date and time of the transaction.
Description: The user can provide details regarding the spending (optional).
Business Rules: BR-30, BR-79, BR-80, BR-81
Screen Layout:



Step-by-Step Display
Information Section:

Input Field for Amount: E.g., "30.000"
Date and Time Selector: Users select the date from a calendar and set the time.
Input Field for Description: E.g., "Xem phim".

Functionality:
Normal Cases:

Step 1:
User selects the option to add a spending transaction.
Step 2:
User enters all required information (amount, date, description).
Step 3:
User submits the information.

Step 4:

System confirms the recording of the transaction.

3.6.2 View list transactions

<u>Function Trigger</u>:

User selects the "View Transactions" option from the main menu or relevant section.

<u>Function Description:</u>

To provide users with a detailed list of their transactions over a specified time frame.

Preconditions: PRE-02, PRE-19

Post-conditions: POST-62

<u>Function Details:</u>

Data:

Selected Date Range: User specifies the start and end dates to view transactions.

Transaction Data:

Displayed in a clear format, including:

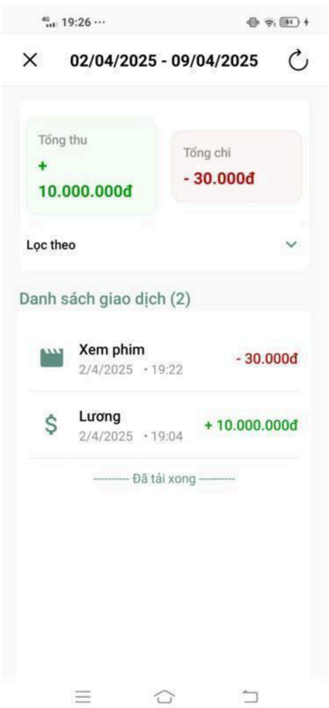
Transaction date

Description

Amount

Business Rules: BR-82, BR-83, BR-84

<u>Screen Layout:</u>



Step-by-Step Display:

Progress Indicator: Shows the transaction retrieval progress.

Date Selection:

Users can click on a calendar to select a specific date range.

Input Data Section:

Start Date: Displayed date (e.g., "02/04/2025").

End Date: Displayed date (e.g., "09/04/2025").
Statistics Display:
Total Income: Highlighted format (e.g., "+ 10.000.000đ").
Total Expenditure: Shown in red (e.g., "- 30.000đ").
Net Savings: Calculated and displayed below.

Functionality:

Normal Cases:

Step 1:
User selects the "View Transactions" option.

Step 2:
System displays available date range options.

Step 3:
User selects a date range and submits.

Step 4:
System provides a detailed list of transactions based on the criteria.

3.6.3 Edit transaction

Function Trigger:

User selects the option to edit a spending transaction from the main menu or relevant section.

Function Description:

To enable users to modify and categorize their spending efficiently.

Preconditions: PRE-02, PRE-27

Post-conditions: POST-63

Function Details:

Data

Amount: The user specifies the amount of the transaction.

Date: The user specifies the date and time of the transaction.

Description: The user can provide details regarding the spending (optional).

Business Rules: BR-85, BR-86, BR-87

Screen Layout:

Step-by-Step Display

Information Section:

Input Field for Amount: E.g., "30.000"

Date and Time Selector: Users select the date from a calendar and set the time.

Input Field for Description: E.g., "Viewing Movie".

Submit Action:

Submit Button: To finalize and save the transaction.

Results Screen

Displays a confirmation that the transaction has been edited successfully.

Users can view a summary of their recent transactions.

Functionality:

Normal Cases:

Step 1:
User selects the option to edit a spending transaction.
Step 2:
User enters all required information (amount, date, description).
Step 3:
User submits the information.
Step 4:
System confirms the recording of the edited transaction.

3.6.4 Delete transaction

Function Trigger:

User selects the option to delete a spending transaction from the main menu or relevant section.

Function Description:

To enable users to efficiently remove unwanted or erroneous spending transactions.

Preconditions: PRE-02, PRE-27

Post-conditions: POST-64

Function Details:

Data
Transaction ID: The user selects the transaction to delete.

Business Rules: BR-88, BR-89

Screen Layout:

Step-by-Step Display

Information Section:
Transaction List: Displays all transactions with a delete option next to each.

Submit Action:
Delete Button: To confirm the deletion of the selected transaction.

Functionality:

Normal Cases:
Step 1:
User selects the option to delete a spending transaction.
Step 2:
User confirms the deletion.
Step 3:
System confirms the deletion of the transaction.

3.6.5 Create recurring transaction

Function Trigger:

User selects the option to create a recurring spending transaction from the main menu or relevant section.

Function Description:

To enable users to efficiently set up transactions that occur on a regular basis.

Preconditions: PRE-02

Post-conditions: POST-64, POST-65

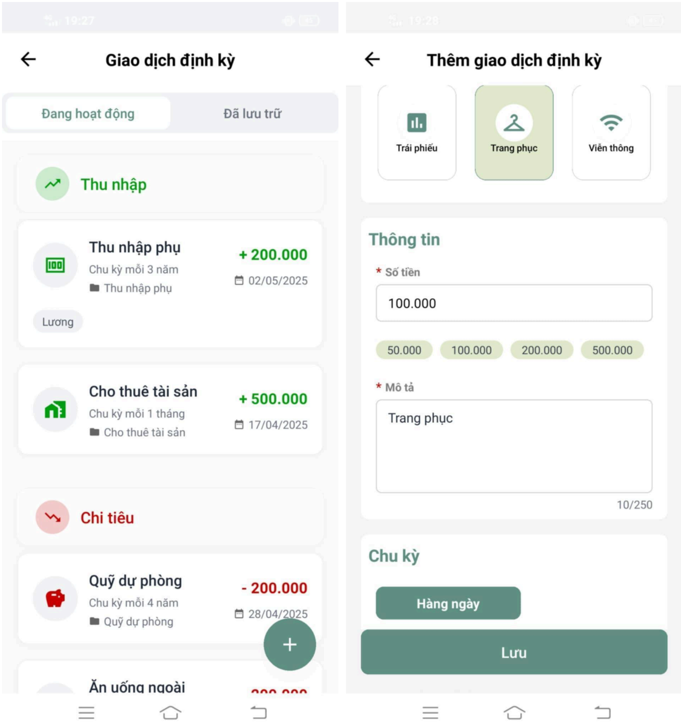
<u>Function Details:</u>

Data

Transaction ID: A new unique identifier is generated for the recurring transaction.

Business Rules: BR-90, BR-91, BR-92

<u>Screen Layout:</u>



<u>Functionality:</u>

Normal Cases:

Step 1:

User selects the option to create a recurring transaction.

Step 2:

User enters necessary details and confirms the creation.

Step 3:

The system validates the inputs:

Ensures required fields (e.g., Amount, Description, Cycle, Date) are not empty.

Step 4:

The user touches "Lưu" to create a recurring transaction.

Step 5:

Displays a success message: "Tạo giao dịch định kì thành công."

4. Non-Functional Requirements

4.1: Usability

- Intuitive user interface with clear navigation
- Responsive design compatible with mobile and desktop devices
- Accessibility features for users

4.2: Scalability With Disabilities

- Cloud infrastructure (Digital Ocean) will be used to dynamically scale server resources
- Horizontal scaling capabilities
- Load balancing implementation

4.3: Security

- User authentication will be managed with OAuth 2.0, ensuring secure login with third-party services like Google.
- End-to-end encryption for sensitive data
- Regular security audits and vulnerability assessments

4.4: Testability

- Comprehensive unit and integration test coverage
- Logging mechanisms for error tracking

4.5: Availability

- System uptime target of 99.9%

4.6: Maintainability

- Modular and clean code architecture
- Comprehensive documentation
- Easy configuration and update processes

4.7: Portability

- Platform-independent design
- Support for multiple operating systems
- Containerization using Docker

5. Requirement Appendix

5.1 Business Rules

Table 26 - Business Rules

No	Business Rule
BR-01	Users must have a valid account with the system to use this login method.
BR-02	The system should securely handle and store user credentials.
BR-03	Account lockout measures may be implemented to enhance security.
BR-04	Only authenticated users can access profile information.
BR-05	Profile data must be securely handled and displayed.
BR-06	Password length must be between 8 and 50 characters. It must include at least one uppercase letter, one lowercase letter, one digit, and one special character

BR-07	The system must securely manage and handle Google via Firebase Authentication.
BR-08	The system should request only the necessary permissions from Google to authenticate the user.
BR-09	Profile information must be valid (e.g., email format, phone number format).
BR-10	The system must ensure profile updates are secure and validated before saving.
BR-11	Each group name must be unique within the system to prevent confusion and ensure accurate tracking.
BR-12	Be exactly 12 digits. Contain only numeric characters. Pass any additional checksum or formatting validation required by the financial institution.
BR-13	Be in JPG, JPEG, or PNG format. Not exceed the maximum allowed size (e.g., 5MB), which has appropriate resolution (optional business-specific constraint).
BR-14	When a group is created, the user who created the group is automatically assigned the Leader role and has elevated permissions within the group.
BR-15	The system must not allow updates to non-existent groups under any circumstances
BR-16	Only users with the role of Leader can update the group's details.
BR-17	Each group update must be logged, recording who updated it, what was changed, and when.
BR-18	The system must notify all members that the group has been disbanded.
BR-19	Once disbanded, the group cannot be restored unless backed up or archived first.
BR-20	All associated data (transactions, members, goals) must either be archived or permanently deleted.
BR-21	The user’s transaction history remains intact but is marked as inactive or past member.
BR-22	The system must notify remaining group members of the user’s departure.

BR-23	Upon leaving, the user's future access to group data and interactions is revoked.
BR-24	If the user is the sole group admin, they must assign admin rights to another member before leaving.
BR-25	A user can only leave a group they are a member of.
BR-26	The group's total balance is recalculated upon adding the transaction.
BR-27	Transactions must be immutable after submission unless edited explicitly.
BR-28	The system records who added the transaction and when.
BR-29	The system must categorize the transaction (e.g., food, transport) if applicable.
BR-30	The amount must be a positive number.
BR-31	Each transaction must include a valid amount, description, and date.
BR-32	Only members of the group can add transactions to it.
BR-33	Validations apply to edited data (e.g., amount must be positive, required fields must be filled).
BR-34	Changes must trigger a recalculation of group totals if relevant.
BR-35	The system logs the modification, including the user who made the change and timestamp.
BR-36	Only the transaction creator or a group admin can edit a transaction.
BR-37	The system logs the deletion activity (who deleted, when, which transaction).
BR-38	Deleted transactions are either permanently removed or archived based on retention policies.
BR-39	Target amount must be a positive number.

BR-40	A financial goal must include a valid type (e.g., savings, spending cap), target amount, and duration.
BR-45	Kicked members lose access to all group data and transactions.
BR-46	Only members with admin rights can kick other members.
BR-47	Statistics must be updated in real-time or based on the latest available data
BR-48	New invitees must have an account or register before joining.
BR-49	QR codes must expire after a predefined time for security (e.g., 24 hours).
BR-50	Roles (e.g., Admin, Member) must be clearly shown.
BR-51	The list should be updated in real-time or reflect the latest group data.
BR-52	Sensitive banking data must be stored securely and comply with data protection standards.
BR-53	Duplicate bank accounts (same number and bank) cannot be added by the same user.
BR-54	The account name must match the registered name with the bank (if verified).
BR-55	The bank account number must be exactly 12 digits (or follow local validation standards).
BR-56	Editing a bank account does not bypass the maximum number of allowed accounts.
BR-57	Duplicate bank account entries are not allowed.
BR-58	The updated bank account number must pass validation (e.g., 12 digits).
BR-59	Only authenticated users can modify bank account information.
BR-60	Read-only access is enforced – no edits can be made from the view screen.
BR-61	Viewing sensitive bank account details may require an additional authentication step (e.g., password or OTP).

BR-62	Bank account information must be displayed securely (e.g., partially masked account number).
BR-63	A confirmation prompt is required before the deletion is finalized.
BR-64	Any model change must be logged and timestamped for audit or future references.
BR-65	Models must allocate 100% of income (sum of all categories must equal total income).
BR-66	Users can choose from predefined models (6-Jars,80/20,50/30/20).
BR-67	Report data must be fetched in real-time or from the most recent sync.
BR-68	Data must be displayed in both numeric and visual format (e.g., pie chart, bar graph).
BR-69	Reports must include income, expenses, and balances for selected time ranges.
BR-70	The system should visually indicate progress toward the limit (e.g., progress bar).
BR-71	Spending limits must be defined per category and time period (e.g., monthly, weekly).
BR-72	Notifications must be triggered when spending approaches or exceeds the goal.
BR-73	The system must track actual vs. planned spending and provide visual indicators.
BR-74	Goals should be measurable (specific amount) and adjustable at any time.
BR-75	The quiz results must be saved to the user's profile for future reference.
BR–76	Free-text responses are collected from the user for all financial behavior questions.
BR-77	The final consumer model is determined by an LLM based on the full set of user responses.
BR-78	The quiz must consist of a fixed number of multiple-choice questions.
BR-79	The category must be selected from predefined categories.

BR-80	The transaction date cannot be in the future.
BR-81	The transaction must include required fields: amount, date, and category.
BR-82	The list must include both income and expense transactions with clear visual distinctions.
BR-83	Each transaction entry must display: category, amount, date, and notes (if available).
BR-84	Users can filter transactions by date range, category, or amount.
BR-85	All required fields must be validated before saving (e.g., amount > 0, valid date).
BR-86	Only the user who created the transaction can edit it, unless shared permissions apply.
BR-87	Users may edit transaction fields such as amount, category, date, notes, and payment method.
BR-88	Deletion updates any affected reports, limits, or goals.
BR-89	Deletion must be confirmed by the user through a confirmation dialog.
BR-90	Users can cancel or modify recurring transactions at any time.
BR-91	Recurring transactions must have a valid start date and optionally an end date or occurrence count.
BR-92	Recurring transactions must include frequency (daily, weekly, monthly, etc.).
BR-93	Group admins must approve the transaction before it affects the fund balance.
BR-94	Users must grant EzMoney permission to access transaction information via Open Banking APIs.
BR-95	Transactions must be confirmed via webhook from the bank to ensure synchronization and transparency.
BR-96	Users are responsible for protecting their login information; EzMoney only uses the information for bank linking purposes.

BR-97	Automatic updating of wallet transactions (deposit/withdraw) is required after confirmation through API.
BR-98	App will not hold group funds; the group leader manages funds directly.
BR-99	App only acts as a recording tool to ensure transparency for group transactions.
BR-100	App will not intervene in disputes between group members.
BR-101	Group leaders must use a separate management account for group fund management.
BR-102	Group leaders have full authority to manage all deposit/withdraw transactions for the group account.
BR-103	Group fund transactions must be recorded by the system and cannot be deleted.
BR-104	Joining a group means agreeing to all terms set by the group leader.
BR-105	Only leaders can manage group goals; they can update goals before the deadline.
BR-106	Group goals automatically move to 'COMPLETED' if achieved, or 'ARCHIVED' when expired.
BR-107	All models must have a 'Revenue' category with 0% allocation.
BR-108	'Saving' is treated as an expense in the model.
BR-109	The total of all categories in a model must equal 100%.
BR-110	Users can only use predefined models from the system.
BR-111	Users must cancel the current model before switching to a new one.
BR-112	Personal budgets are set based on subcategories selected from the active model.
BR-113	Personal budgets must be validated against the corresponding model category percentage and user's income.

BR-114	Personal budgets can be updated before the model's time frame ends.
BR-115	Personal budgets automatically move to 'COMPLETED' when achieved, or 'ARCHIVED' when expired.
BR-116	Categories of type 'EXPENSE' can have an 'isSaving' attribute to identify saving-related expenses.
BR-117	A spending model must be selected before creating any transaction.
BR-118	The system only supports linking and verification with EzMoney Bank Sandbox accounts.
BR-119	Bank account linking requires correct verification of account number and account holder with the Sandbox bank.
BR-120	Each user can only link up to three bank accounts.
BR-121	The target duration for any group goal must not exceed 1 year, and the deadline must be set to a date at least 3 days after the current date.

5.2 Application Messages List

Table 27 - Application Messages List

No	Message Code	Message Type	Context	Content
1	MEZ-01	Toast message	Incorrect login account or password	Email hoặc mật khẩu không hợp lệ.
2	MEZ-02	Toast message	Account Does Not Verify Email	Tài khoản không xác minh email
3	MEZ-03	Toast message	Password Is Incorrect	Mật khẩu không đúng
4	MEZ-04	Toast message	Token Not Valid	Mã thông báo không hợp lệ
5	MEZ-05	Toast message	Otp Invalid	Otp không hợp lệ

6	MEZ-06	Toast message	Old Password Invalid	Mật khẩu cũ không hợp lệ
7	MEZ-07	Toast message	Can Not Reset Password	Không thể đặt lại mật khẩu
8	MEZ-08	Toast message	Account Verified	Tài khoản đã xác minh
9	MEZ-09	Toast message	Otp Has Sent	Otp đã gửi
10	MEZ-10	Toast message	Email Not Request Otp Code	Email không yêu cầu mã Otp
11	MEZ-11	Toast message	Login successfully	Đăng nhập thành công
12	MEZ-12	Toast message	Login with google successfully	Đăng nhập bằng google thành công
13	MEZ-13	Toast message	Token refresh successfully	Làm mới mã thông báo thành công
14	MEZ-14	Toast message	The OTP has been sent to your email. Please verify it to log in.	OTP đã được gửi đến email của bạn. Vui lòng xác minh để đăng nhập.
15	MEZ-15	Toast message	Change password successfully	Đổi mật khẩu thành công
16	MEZ-16	Toast message	The OTP reset password has been sent to your email	Mật khẩu đặt lại OTP đã được gửi đến email của bạn
17	MEZ-17	Toast message	You can reset password now	Bạn có thể đặt lại mật khẩu ngay bây giờ
18	MEZ-18	Toast message	Account created	Tài khoản đã được tạo
19	MEZ-19	Toast message	Updated user successfully	Cập nhật người dùng thành công

20	MEZ-20	Toast message	Deleted user successfully	Xóa người dùng thành công
21	MEZ-21	Toast message	Banned user successfully	Cấm người dùng thành công
22	MEZ-22	Toast message	Update device token successfully	Cập nhật mã thông báo thiết bị thành công
23	MEZ-23	Toast message	Financial report generated successfully.	Báo cáo tài chính đã được tạo thành công.
24	MEZ-24	Toast message	Financial report retrieved successfully.	Đã truy xuất báo cáo tài chính thành công.
25	MEZ-25	Toast message	Financial report list retrieved successfully.	Đã truy xuất danh sách báo cáo tài chính thành công.
26	MEZ-26	Toast message	Financial report updated successfully.	Đã cập nhật báo cáo tài chính thành công.
27	MEZ-27	Toast message	Financial report deleted successfully.	Đã xóa báo cáo tài chính thành công.
28	MEZ-28	Toast message	Report name is required.	Tên báo cáo là bắt buộc.
29	MEZ-29	Toast message	Start date is required.	Ngày bắt đầu là bắt buộc.
30	MEZ-30	Toast message	End date is required.	Ngày kết thúc là bắt buộc.
31	MEZ-31	Toast message	Report type is required.	Loại báo cáo là bắt buộc.
32	MEZ-32	Toast message	Group ID is required for group reports.	ID nhóm là bắt buộc đối với báo cáo nhóm.
33	MEZ-33	Toast message	You do not have permission to access this report.	Bạn không có quyền truy cập vào báo cáo này.
34	MEZ-34	Toast message	Group created successfully	Nhóm đã tạo thành công

35	MEZ-35	Toast message	Group get all successfully	Nhóm đã nhận được tất cả thành công
36	MEZ-36	Toast message	Group closed successfully	Nhóm đã đóng thành công
37	MEZ-37	Toast message	Group member removed successfully	Thành viên nhóm đã xóa thành công
38	MEZ-38	Toast message	Group member role updated successfully	Vai trò thành viên nhóm đã cập nhật thành công
39	MEZ-39	Toast message	Only the group leader can invite members.	Chỉ có người đứng đầu nhóm mới có thể mời thành viên.
40	MEZ-40	Toast message	Invalid invitation token.	Mã thông báo mời không hợp lệ.
41	MEZ-41	Toast message	Invitation accepted successfully.	Lời mời đã được chấp nhận thành công.
42	MEZ-42	Toast message	You have left the group successfully.	Bạn đã rời khỏi nhóm thành công.
43	MEZ-43	Toast message	Fundraising request list fetched successfully	Danh sách yêu cầu gây quỹ đã được tải thành công
44	MEZ-44	Toast message	Fundraising request details fetched successfully	Chi tiết yêu cầu gây quỹ đã được tải thành công
45	MEZ-45	Toast message	Bank account list fetched successfully	Danh sách tài khoản ngân hàng đã được tải thành công
46	MEZ-46	Toast message	Bank account details fetched successfully	Chi tiết tài khoản ngân hàng đã được tải thành công
47	MEZ-47	Toast message	Bank account created successfully	Tài khoản ngân hàng đã được tạo thành công
48	MEZ-48	Toast message	Bank account updated successfully	Tài khoản ngân hàng đã được cập nhật thành công
49	MEZ-49	Toast message	Bank account deleted successfully	Đã xóa tài khoản ngân hàng thành công

5.3 Other Requirements

5.3.1 Pre-condition

Table 28 - Pre-condition

No	Pre-condition
PRE-01	The user has a valid email, verified phone number and password.
PRE-02	The user is logged into the system.
PRE-03	The customer has a valid Google account.
PRE-04	The MoneyEZ system is integrated with Google OAuth for authentication.
PRE-05	The user has permission to delete the transaction (e.g., creator or admin).
PRE-06	The group must already exist in the system.
PRE-07	The user must be the group’s Leader or have been granted permission to edit group information.
PRE-08	There must be no unresolved financial transactions or pending settlements in the group
PRE-09	If the user is the only admin, they must transfer admin rights before leaving.
PRE-10	The user is currently a member of the group.
PRE-11	Required fields for the transaction (e.g., amount, description) must be provided.
PRE-12	The user has permission to edit the transaction (e.g., they created it or are an admin).
PRE-13	A withdrawal request has been initiated and is open for voting.
PRE-14	The transaction exists and belongs to the specified group.
PRE-15	Required goal input fields (e.g., target amount, type, duration) must be filled.
PRE-16	The user has not already voted on this withdrawal request.
PRE-17	A withdrawal request has been initiated and is open for voting.
PRE-18	The user has already added at least one bank account.
PRE-19	The user has already provided basic financial data (e.g., income, goals, group affiliations if any).

PRE-20	The time range or filters (e.g., category, account, group) are selected by the user or default to the current month.
PRE-21	Categories have been created or are available in the system.
PRE-22	The user has enabled access to financial tracking or budgeting features.
PRE-23	The user has not taken the quiz in the current session, or has chosen to retake it.
PRE-24	The quiz module must be available and properly loaded.
PRE-25	Admin is authenticated and authorized to access the customer management module.
PRE-26	The user must have at least one spending category created or available by default.
PRE-27	A valid transaction must already exist and be accessible by the user.

5.3.2 Post-condition

Table 29 - Post-condition

No	Post-condition
POST-01	The user is successfully authenticated and logged into the system using their account credentials.
POST-02	The user successfully views their personal profile information.
POST-03	The user successfully resets their password and can log into the system with the new password.
POST-04	The customer is successfully authenticated and logged into the MoneyEz system.
POST-05	If it’s the customer’s first time logging in, the system may prompt for additional registration information.
POST-06	The customer successfully updates their profile information, and the system stores the changes.
POST-07	The group’s information (e.g., name, description, bank account, profile picture) is updated in the database.
POST-08	All associated data (transactions, history, goals) is archived or deleted.

POST-09	All group members are notified that the group has been disbanded.
POST-10	The group is removed from the active groups list.
POST-11	The system logs the leave action (who left, from which group, and when).
POST-12	The user no longer has access to group data, activities, or notifications.
POST-13	Remaining group members are notified of the user's exit.
POST-14	The user is removed from the group’s active member list.
POST-15	A log entry is created indicating who added the transaction and when.
POST-16	The user and other members are notified
POST-17	The group balance or financial summary is updated.
POST-18	The new transaction is saved to the group’s transaction history.
POST-19	The transaction details are updated in the database.
POST-20	A log or history record of the edit is saved.
POST-21	An audit log of the deleted transaction is saved (if applicable).
POST-22	The transaction is removed from the group’s transaction history.
POST-23	The system begins tracking progress against the goal.
POST-24	The financial goal is saved and associated with the specified group.
POST-25	The system updates the current vote tally.
POST-26	The user’s vote is recorded and associated with the withdrawal request.
POST-27	The removed member receives a notification or alert (if applicable).
POST-28	The selected member is removed from the group.

POST-29	A log entry is created indicating the user viewed group statistics (optional audit).
POST-30	The user can optionally filter or customize the data view (e.g., by date, category).
POST-31	A success message is shown to both the inviter and the invited user.
POST-32	The invited user, upon scanning the QR code, receives a group join prompt.
POST-33	A QR code is generated and displayed to the inviting user.
POST-34	Each member is displayed with their role and available contribution data.
POST-35	The user is presented with a list of current group members.
POST-36	An invitation email is sent to the provided address.
POST-37	A success message is displayed to the inviting user.
POST-38	The invited member receives a prompt to accept and join the group.
POST-39	The user may interact with the list (e.g., filter, search, sort).
POST-40	The user sees the full list of transactions for the selected group.
POST-41	The new account appears in the user's list of saved accounts.
POST-42	The bank account is successfully linked to the user’s profile.
POST-43	The bank account information is updated and stored securely.
POST-44	The user may proceed to edit or delete the account if needed from this screen.
POST-45	The user successfully views the bank account details.
POST-46	The UI refreshes to reflect the updated list of linked accounts.
POST-47	The selected bank account is successfully deleted from the user’s profile.
POST-48	The UI updates with a confirmation and displays the active spending model.

POST-49	The system may trigger insights or suggestions based on the chosen model.
POST-50	The selected or customized spending model is saved to the user's profile.
POST-51	The user can export or share the report (e.g., via csv, if allowed).
POST-52	A report is generated and displayed based on the user's selected filters.
POST-53	The admin can view customer details, including contact information and status.
POST-54	The system monitors and compares transactions against the set limit.
POST-55	A spending limit is saved and applied to the selected category.
POST-56	The system begins tracking expenses against the defined goal.
POST-57	A spending goal is saved and linked to the selected category and time range.
POST-58	The quiz result is stored in the user's profile.
POST-59	The user's consumer spending model is updated based on quiz results.
POST-60	The app updates spending summaries and reports accordingly.
POST-61	A new spending transaction is saved to the user's transaction list.
POST-62	The filtered/sorted transaction list is displayed to the user.
POST-63	The system updates the transaction details in the database.
POST-64	The system automatically handles future entries according to the user-defined frequency.
POST-65	A new recurring transaction schedule is saved to the system.
POST-66	The request appears in the group's pending transaction list.
POST-67	The group admin(s) are notified of the request for review and approval.
POST-68	Group members (or authorized users) receive a notification about remind raising fund in group.

IV. Software Design Description

1. System Design

1.1 System Architecture

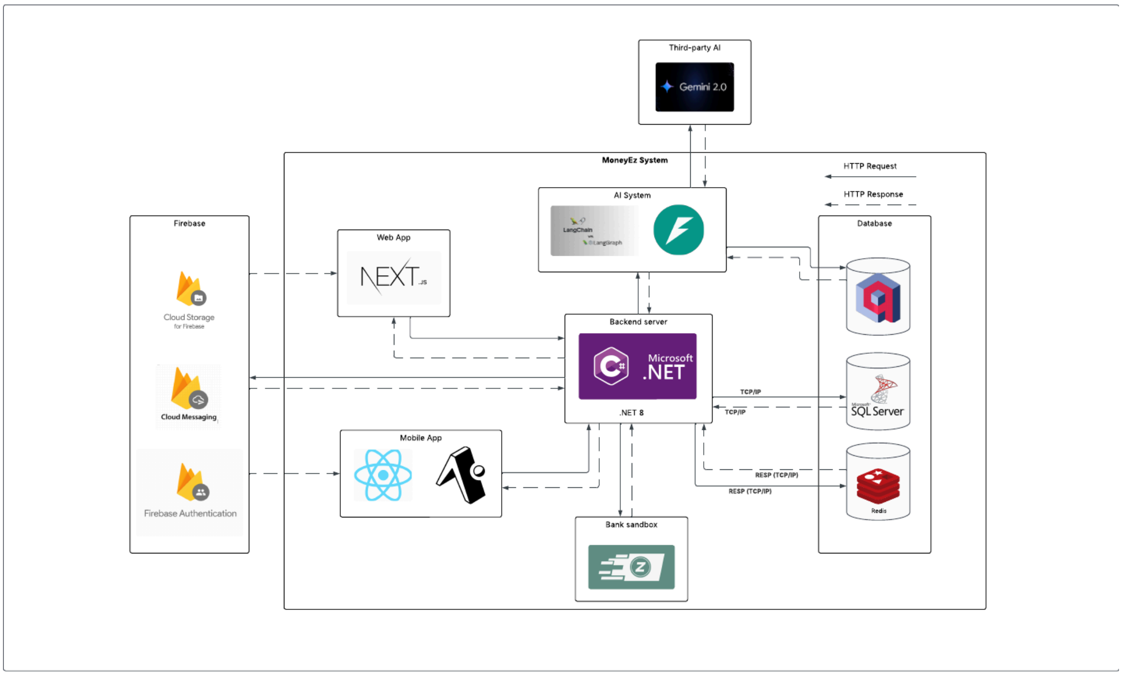


Figure 8 - System Overview Architecture

1.2 Package Diagram

1.2.1 Back-end Package Diagram

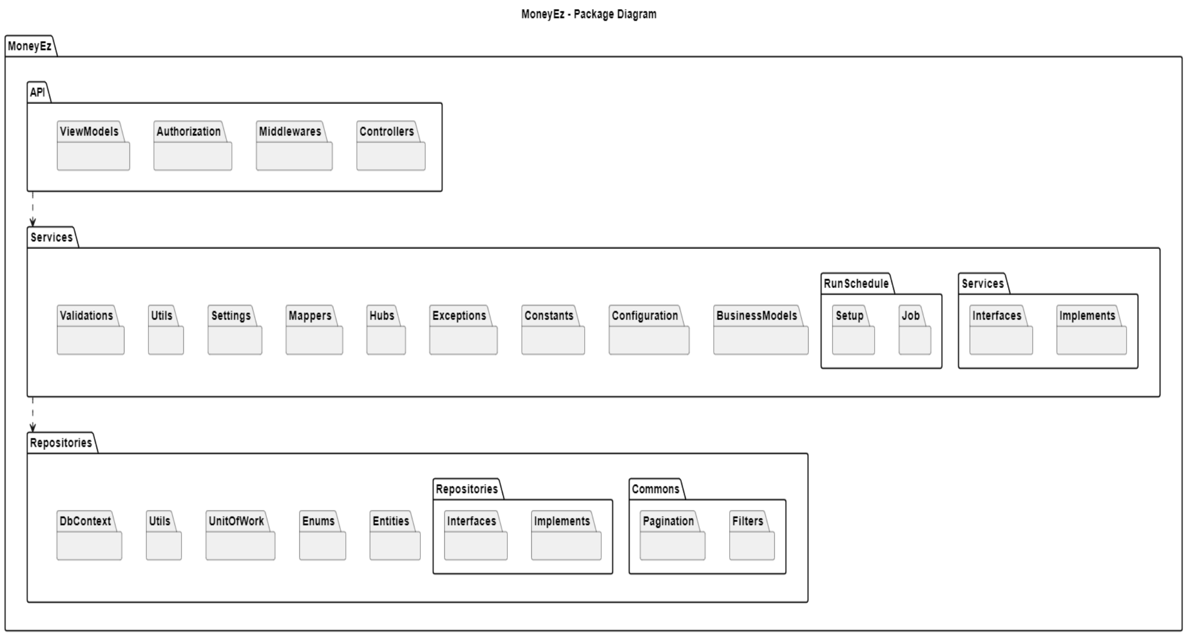


Figure 9 - Back-end Package Diagram

1.2.2 Web Application front-end package diagram

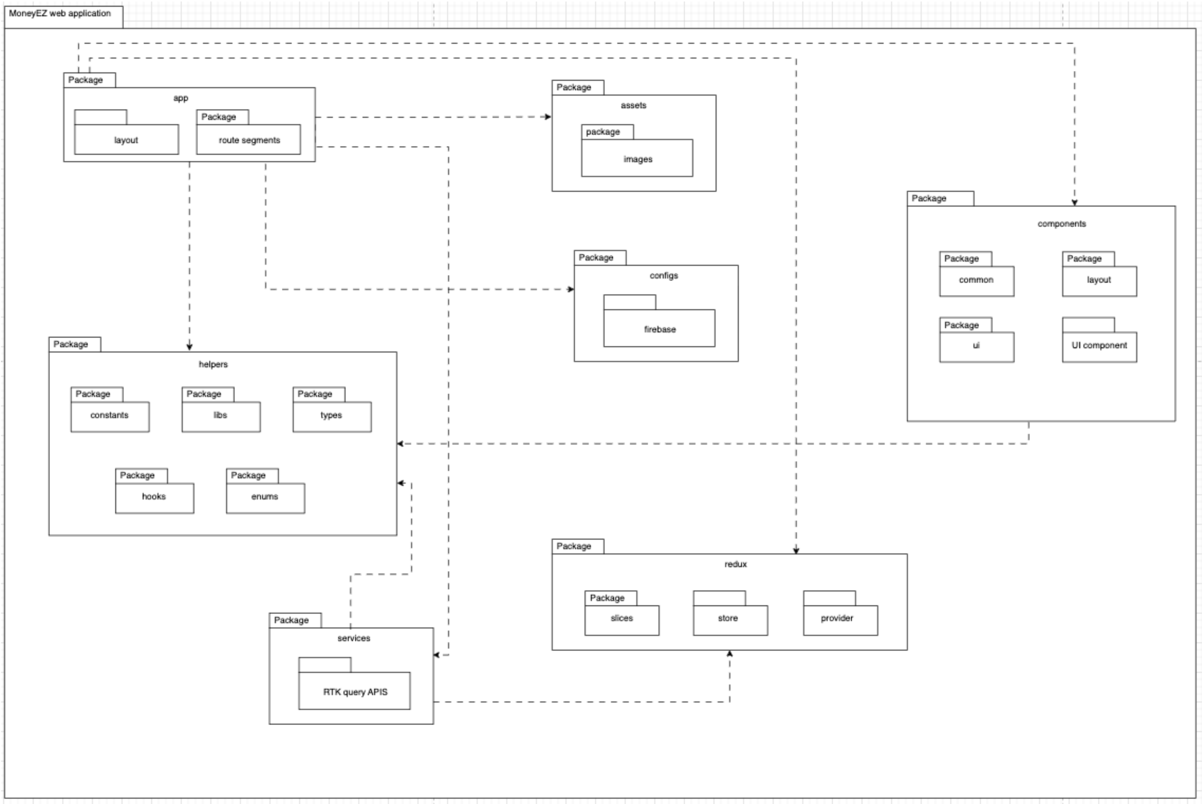


Figure 10 - Web Application Front end Package Diagram

1.2.3 Mobile Application front-end package diagram

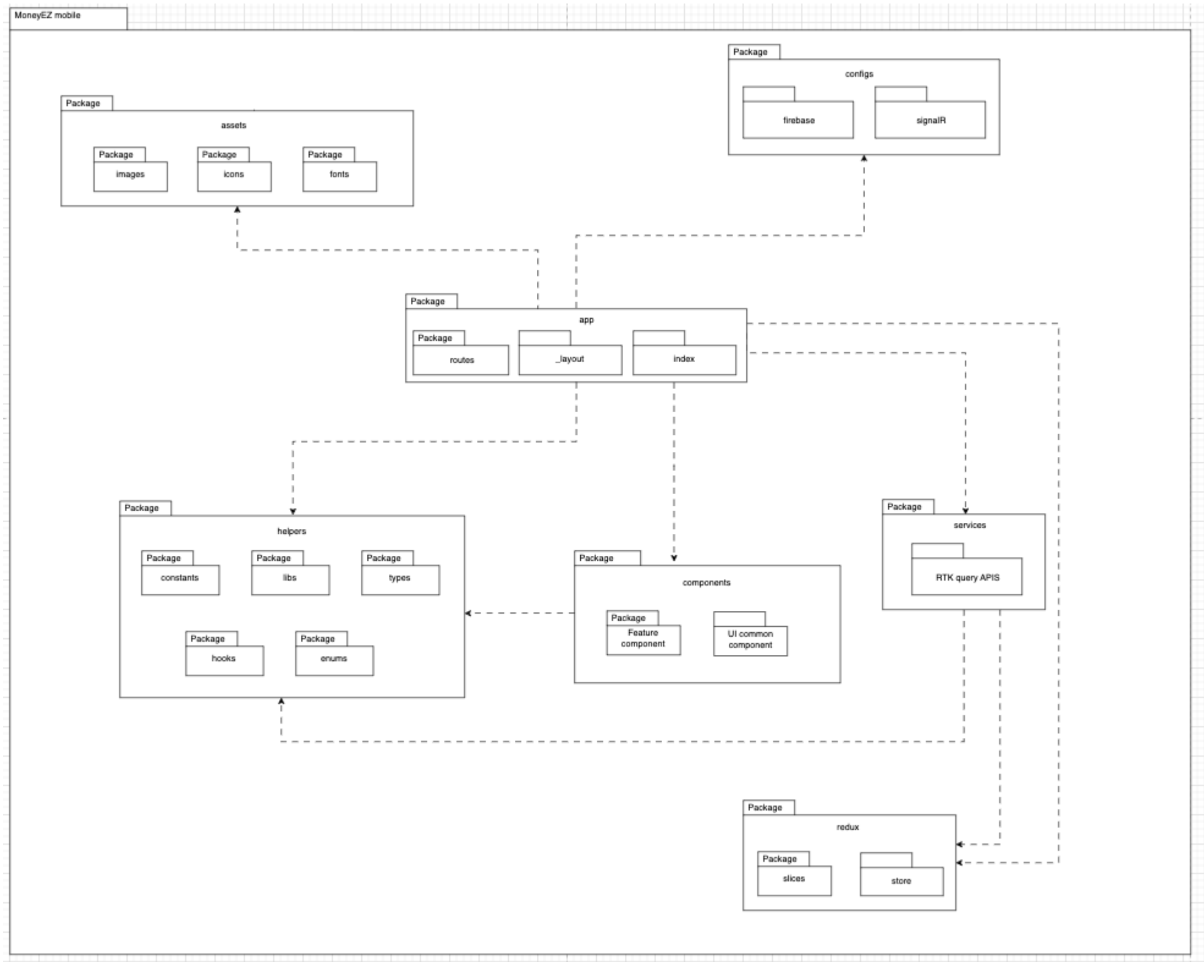


Figure 11 - Mobile Application Front end Package Diagram

2. Database Design

2.1. Physical Database

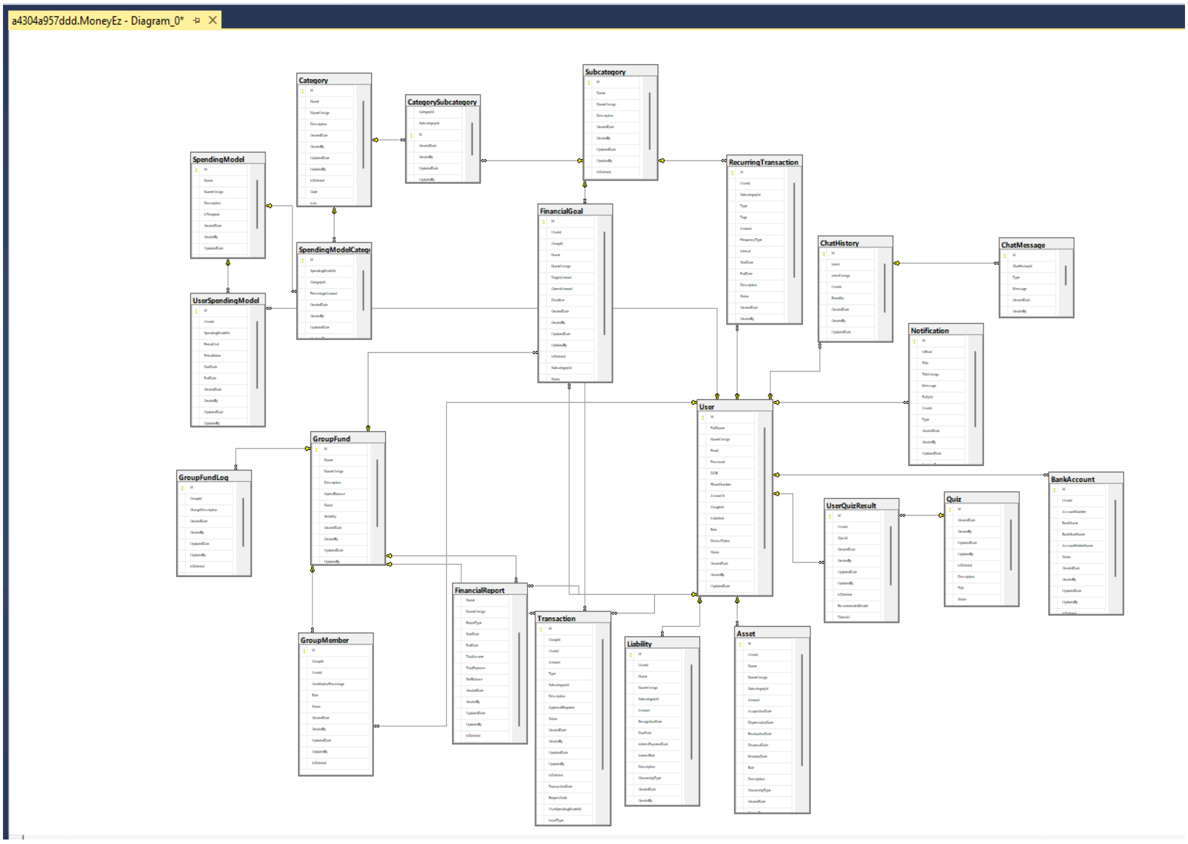


Figure 12 - Database

2.2. Data Dictionary

Table 30 - Data dictionary

No	Table Name	Description
01	User	Stores user information including name, email, password, personal details, status, and address. - Primary keys: Id - Foreign keys: None
02	BankAccount	Stores user bank account information. - Primary keys: Id - Foreign keys: UserId
03	Category	Stores general spending category information. - Primary keys: Id - Foreign keys: None

04	CategorySubcategory	Links Category and Subcategory tables for many-to-many relationships. - Primary keys: Id - Foreign keys: CategoryId, SubcategoryId
05	ChatHistory	Stores user chat history within the application. - Primary keys: Id - Foreign keys: UserId
06	ChatMessage	Stores messages in a chat conversation. - Primary keys: Id - Foreign keys: ChatHistoryId
07	FinancialGoal	Stores user's financial goals, including name, target amount, deadline, and subcategory reference. - Primary keys: Id - Foreign keys: UserId, GroupId, SubcategoryId
08	FinancialReport	Stores financial reports, calculating total income, expenses, and net balance over a time period. - Primary keys: Id - Foreign keys: UserId, GroupId
09	GroupFund	History log of actions performed on group funds. - Primary keys: Id - Foreign keys: GroupId
10	GroupFundLog	History log of actions performed on group funds. - Primary keys: Id - Foreign keys: GroupId
11	GroupMember	Stores information about group fund members including contribution percentage, role, and status. - Primary keys: Id - Foreign keys: GroupId, User
12	Image	Stores images related to entities like bills, receipts, etc. - Primary keys: Id - Foreign keys: EntityId
13	Notification	Stores user notifications including title, message, read status, and categorization info. - Primary keys: Id - Foreign keys: UserId

14	Post	Store information post relate to expense trend - Primary keys: None - Foreign keys: None
15	Quiz	Stores information about quizzes related to spending models or financial knowledge. - Primary keys: Id - Foreign keys: SpendingModelId
16	RecurringTransaction	Stores recurring transactions including amount, frequency, and description. - Primary keys: Id - Foreign keys: UserId, SubcategoryId
17	SpendingModel	Stores information about a spending model, including name, description, and template flag. - Primary keys: Id - Foreign keys: None
18	SpendingModelCategory	Links SpendingModel and Category, storing the percentage allocated to each category. - Primary keys: Id - Foreign keys: SpendingModelId, CategoryId
19	Subcategory	Stores detailed subcategories under each category. - Primary keys: Id - Foreign keys: None
20	Transaction	Stores user financial transactions including amount, type, description, and approval info. - Primary keys: Id - Foreign keys: GroupId, UserId, SubcategoryId
21	UserQuizResult	Stores user's quiz results including answer data and completion time. - Primary keys: Id - Foreign keys: UserId, QuizId

22	UserSpendingModel	Stores user's quiz results including answer data and completion time. - Primary keys: Id - Foreign keys: UserId, QuizId

3. Detailed Design

3.1 Create Transaction

3.1.1 Class Diagram

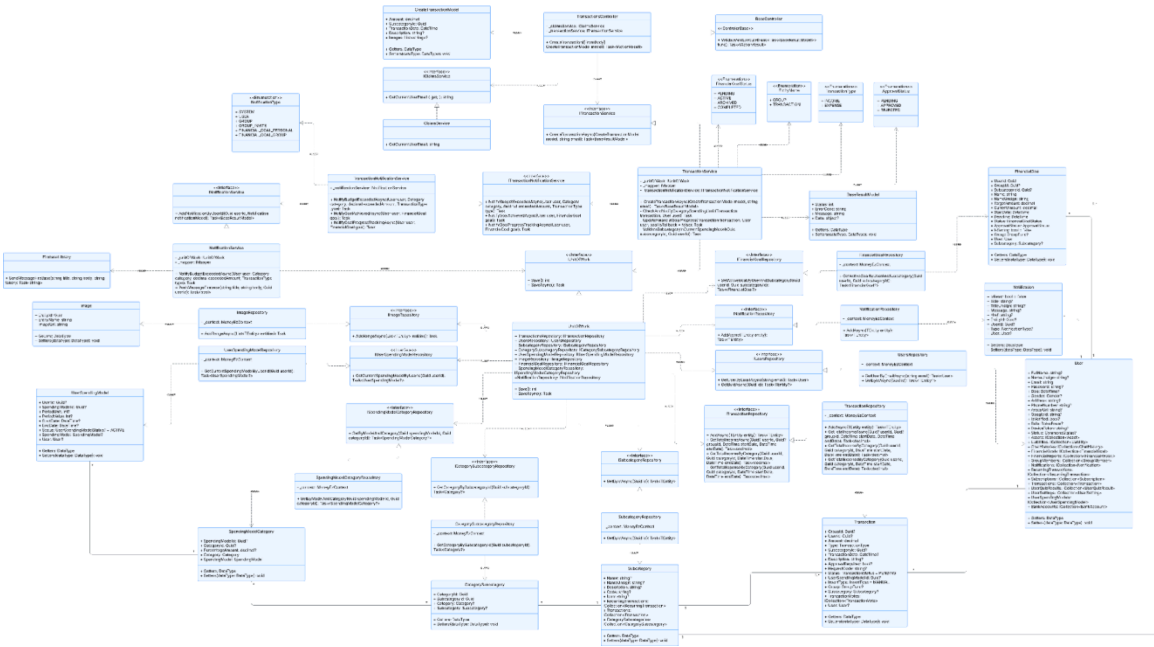


Figure 14 - Class Diagram

3.1.1.1 TransactionsController class

No	Attribute	Data Type	Visibility	Description
1	_claimsService	IClaimsService	Private	Service for handling user claims extracted from the token.
2	_transactionService	ITransactionService	Private	Service responsible for processing transaction logic.

No	Method	Data Type	Visibility	Description
1	CreateTransaction	Task<IActionResult>	Public	API endpoint that receives transaction

				data from the request body to create a new transaction.

3.1.1.2 BaseController class

No	Method	Data Type	Visibility	Description
1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.1.1.3 CreateTransactionModel class

No	Attribute	Data Type	Visibility	Description
1	Amount	decimal	Public	Transaction amount.
2	SubcategoryId	Guid	Public	ID of the subcategory.
3	TransactionDate	DateTime	Public	Date of the transaction.
4	Description	string?	Public	Optional transaction description.
5	Images	List<string>?	Public	Optional list of image URLs.

3.1.1.4 ITransactionService class

No	Method	Data Type	Visibility	Description
1	CreateTransactionAsync	Task<BaseResultModel>	Public	Creates a transaction using given data.

3.1.1.5 TransactionService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.
3	_transactionNotificationService	ITransactionNotificationService	Private	Sends transaction-related notifications.

No	Method	Data Type	Visibility	Description
1	CheckAndNotifyCategorySpendingLimit	Task	Private	Checks spending limit and sends notification.

2	UpdateFinancialGoalProgress	Task	Private	Updates progress toward financial goals.
3	ValidateSubcategoryInCurrentSpendingModel	Task	Private	Validates subcategory in user's spending model.
4	CreateTransactionAsync	Task<BaseResultModel>	Public	Creates a transaction with given data.

3.1.1.6 ClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Email extracted from current user claims.

3.1.1.7 IClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Gets the email of the current user.

3.1.1.8 ITransactionNotificationService class

No	Method	Data Type	Visibility	Description
1	NotifyBudgetExceededAsync	Task	Public	Notifies user when their budget limit is exceeded.
2	NotifyGoalAchievedAsync	Task	Public	Notifies user when a financial goal is achieved.
3	NotifyGoalProgressTrackingAsync	Task	Public	Notifies user about the progress of their financial goal.

3.1.1.9 TransactionNotificationService class

No	Attribute	Data Type	Visibility	Description
1	_notificationService	INotificationService	Private	Service for sending notifications to the user.

No	Method	Data Type	Visibility	Description

1	NotifyBudgetExceededAsync	Task	Public	Notifies user when their budget limit is exceeded.
2	NotifyGoalAchievedAsync	Task	Public	Notifies user when a financial goal is achieved.
3	NotifyGoalProgressTrackingAsync	Task	Public	Notifies user about the progress of their financial goal.

3.1.1.10 INotificationService class

No	Method	Data Type	Visibility	Description
1	AddNotificationByUserId	Task<BaseResultModel>	Public	Adds a notification for a user by their ID.

3.1.1.11 NotificationService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.

No	Method	Data Type	Visibility	Description
1	NotifyBudgetExceededAsync	Task	Public	Notifies user when their budget limit is exceeded.
2	PushMessageFirebase	Task<bool>	Public	Sends a push notification to the user via Firebase.

3.1.1.12 FirebaseLibrary class

No	Method	Data Type	Visibility	Description
1	SendMessageFireBase	Task<string>	Public	Sends a Firebase message with the specified title, body, and token.

3.1.1.13 IUnitOfWork class

No	Method	Data Type	Visibility	Description

1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.1.1.14 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	TransactionsRepository	ITransactionRepository	Public	Repository for managing transaction data.
2	UsersRepository	IUsersRepository	Public	Repository for managing user data.
3	SubcategoryRepository	ISubcategoryRepository	Public	Repository for managing subcategory data.
4	CategorySubcategoryRepository	ICategorySubcategoryRepository	Public	Repository for managing category-subcategory relationships.
5	UserSpendingModelRepository	IUserSpendingModelRepository	Public	Repository for managing user spending models.
6	ImageRepository	IImageRepository	Public	Repository for managing image data.
7	FinancialGoalRepository	IFinancialGoalRepository	Public	Repository for managing financial goal data.
8	SpendingModelCategoryRepository	ISpendingModelCategoryRepository	Public	Repository for managing spending model-category relationships.
9	NotificationRepository	INotificationRepository	Public	Repository for managing notification data.

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.

2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.1.1.15 IImageRepository class

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.

3.1.1.16 IUserSpendingModelRepository class

No	Method	Data Type	Visibility	Description
1	GetCurrentSpendingModelByUserId	Task<UserSpendingModel?>	Public	Asynchronously retrieves the current spending model for a user by their ID.

3.1.1.17 ISpendingModelCategoryRepository class

No	Method	Data Type	Visibility	Description
1	GetByModelAndCategory	Task<SpendingModelCategory?>	Public	Asynchronously retrieves the spending model category by its spending model and category IDs.

3.1.1.18 ICategorySubcategoryRepository class

No	Method	Data Type	Visibility	Description
1	GetCategoryBySubcategoryId	Task<Category?>	Public	Asynchronously retrieves the category associated with a subcategory ID.

3.1.1.19 ISubcategoryRepository class

No	Method	Data Type	Visibility	Description
1	GetByIdAsync	Task<TEntity>	Public	Asynchronously retrieves an entity by its ID.

3.1.1.20 ITransactionRepository class

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity and returns the added entity.

2	GetTotalIncomeAsync	Task<decimal>	Public	Asynchronously calculates the total income for a user or group within a date range.
3	GetTotalIncomeByCategory	Task<decimal>	Public	Asynchronously calculates the total income for a user and category within a date range.
4	GetTotalExpenseByCategory	Task<decimal>	Public	Asynchronously calculates the total expense for a user and category within a date range.

3.1.1.21 IUsersRepository class

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.
2	GetByIdAsync	Task<TEntity?>	Public	Asynchronously retrieves an entity by its ID, returning null if not found.

3.1.1.22 INotificationRepository class

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity to the repository and returns the added entity.

3.1.1.23 IFinancialGoalRepository class

No	Method	Data Type	Visibility	Description
1	GetActiveGoalByUserAndSubcategory	Task<FinancialGoal?>	Public	Asynchronously retrieves the active financial goal for a user and subcategory.

3.1.1.24 FinancialGoalRepository class

No	Attribute	Data Type	Visibility	Description

1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetActiveGoalByUserAndSubcategory	Task<FinancialGoal?>	Public	Asynchronously retrieves the active financial goal for a user and subcategory.

3.1.1.25 NotificationRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity to the repository and returns the added entity.

3.1.1.26 UsersRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.
2	GetByIdAsync	Task<TEntity?>	Public	Asynchronously retrieves an entity by its ID, returning null if not found.

3.1.1.27 TransactionRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity and returns the added entity.
2	GetTotalIncomeAsync	Task<decimal>	Public	Asynchronously calculates the total income for a user or group within a date range.
3	GetTotalIncomeByCategory	Task<decimal>	Public	Asynchronously calculates the total income for a user and category within a date range.
4	GetTotalExpenseByCategory	Task<decimal>	Public	Asynchronously calculates the total expense for a user and category within a date range.

3.1.1.28 SubcategoryRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByIdAsync	Task<TEntity>	Public	Asynchronously retrieves an entity by its ID.

3.1.1.29 CategorySubcategoryRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetCategoryBySubcategoryId	Task<Category?>	Public	Asynchronously retrieves the category associated with a subcategory ID.

3.1.1.30 SpendingModelCategoryRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByModelAndCategory	Task<SpendingModelCategory?>	Public	Asynchronously retrieves the spending model category by its spending model and category IDs.

3.1.1.31 ImageRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.

3.1.1.32 UserSpendingModelRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetCurrentSpendingModelByUserId	Task<UserSpendingModel?>	Public	Asynchronously retrieves the current spending model for a user by their ID.

3.1.1.33 Image class

No	Attribute	Data Type	Visibility	Description
1	EntityId	Guid	Public	The unique identifier for the entity.
2	EntityName	string	Public	The name of the entity.
3	ImageUrl	string	Public	The URL of the image associated with the entity.

3.1.1.34 UserSpendingModel class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid?	Public	The unique identifier for the user.
2	SpendingModelId	Guid?	Public	The unique identifier for the spending model.
3	PeriodUnit	int?	Public	The unit of the period.
4	PeriodValue	int?	Public	The value of the period.
5	StartDate	DateTime?	Public	The start date of the spending model.
6	EndDate	DateTime?	Public	The end date of the spending model.
7	Status	UserSpendingModelStatus? = ACTIVE	Public	The status of the user spending model, default is ACTIVE.

8	SpendingModel	SpendingModel?	Public	The spending model associated with the user.
9	User	User?	Public	The user associated with the spending model.

3.1.1.35 SpendingModelCategory class

No	Attribute	Data Type	Visibility	Description
1	SpendingModelId	Guid?	Public	The ID of the associated spending model.
2	CategoryId	Guid?	Public	The ID of the associated category.
3	PercentageAmount	decimal?	Public	The allocated percentage amount for the category.
4	Category	Category	Public	The category linked to the spending model.
5	SpendingModel	SpendingModel	Public	The spending model that includes this category.

3.1.1.36 CategorySubcategory class

No	Attribute	Data Type	Visibility	Description
1	CategoryId	Guid?	Public	The ID of the associated category.
2	SubcategoryId	Guid?	Public	The ID of the associated subcategory.
3	Category	Category?	Public	Navigation property to the category.
4	Subcategory	Subcategory?	Public	Navigation property to the subcategory.

3.1.1.37 Subcategory class

No	Attribute	Data Type	Visibility	Description
1	Name	string?	Public	The name of the subcategory
2	NameUnsign	string?	Public	The unsigned version of the name
3	Description	string?	Public	A short description of the subcategory

4	Code	string?	Public	The code representing the subcategory
5	Icon	string?	Public	Icon associated with the subcategory
6	RecurringTransactions	ICollection<RecurringTransaction>	Public	Related recurring transactions.
7	Transactions	ICollection<Transaction>	Public	Related one-time transactions.
8	CategorySubcategories	ICollection<CategorySubcategory>	Public	Links to categories through relationships.

3.1.1.38 Transaction class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid?	Public	ID of the group fund.
2	UserId	Guid?	Public	ID of the user who made the transaction.
3	Amount	decimal	Public	The amount of the transaction.
4	Type	TransactionType	Public	Type of transaction.
5	SubcategoryId	Guid?	Public	ID of the subcategory linked to this transaction.
6	TransactionDate	DateTime?	Public	The date the transaction occurred.
7	Description	string?	Public	Description of the transaction.
8	ApprovalRequired	bool?	Public	Whether approval is required.
9	RequestCode	string?	Public	Unique request code for approval tracking.
10	Status	TransactionStatus	Public	Current status of the transaction.
11	UserSpendingModelId	Guid?	Public	Related user's spending model ID.
12	InsertType	InsertType	Public	Indicates how the transaction was added.
13	Group	GroupFund?	Public	Navigation property to the related group.

14	Subcategory	Subcategory?	Public	Navigation to the subcategory.
15	TransactionVotes	ICollection<TransactionVote>	Public	Related transaction votes for approval.
16	User	User?	Public	Navigation property to the user

3.1.1.39 User class

No	Attribute	Data Type	Visibility	Description
1	FullName	string?	Public	Full name of the user.
2	NameUnsign	string?	Public	Full name without diacritics
3	Email	string?	Public	Email address of the user.
4	Password	string?	Public	User's password (if not using Google login).
5	Dob	DateTime?	Public	Date of birth.
6	Gender	Gender?	Public	Gender of the user.
7	Address	string?	Public	User’s address.
8	PhoneNumber	string?	Public	User’s phone number.
9	AvatarUrl	string?	Public	URL to the user's avatar image.
10	GoogleId	string?	Public	Google account ID (if signed in with Google).
11	IsVerified	bool?	Public	Whether the user's email is verified.
12	Role	RolesEnum?	Public	Role of the user
13	DeviceToken	string?	Public	Device token for push notifications.
14	Status	CommonsStatus?	Public	Current status of the user.
15	Assets	ICollection<Asset>	Public	List of user's assets.
16	Liabilities	ICollection<Liability>	Public	List of user's liabilities.

17	ChatHistories	ICollection<ChatHistory>	Public	Chat history records.
18	FinancialGoals	ICollection<FinancialGoal>	Public	User’s financial goals.
19	FinancialReports	ICollection<FinancialReport>	Public	Generated financial reports.
20	GroupMembers	ICollection<GroupMember>	Public	User’s membership in groups.
21	Notifications	ICollection<Notification>	Public	Notifications received by the user.
22	RecurringTransactions	ICollection<RecurringTransaction>	Public	Scheduled recurring transactions.
23	Subscriptions	ICollection<Subscription>	Public	User’s active subscriptions.
24	Transactions	ICollection<Transaction>	Public	All transactions linked to the user.
25	UserQuizResults	ICollection<UserQuizResult>	Public	Results from financial quizzes.
26	UserSettings	ICollection<UserSetting>	Public	Custom user settings/preferences.
27	UserSpendingModels	ICollection<UserSpendingModel>	Public	Linked spending models.
28	BankAccounts	ICollection<BankAccount>	Public	Bank accounts linked to the user.

3.1.1.40 Notification class

No	Attribute	Data Type	Visibility	Description
1	IsRead	bool	Public	Indicates whether the notification has been read.
2	Title	string?	Public	Title of the notification.
3	TitleUnsign	string?	Public	Title without diacritics (for search and filtering).
4	Message	string?	Public	Content/message of the notification.
5	Href	string?	Public	Link to redirect when the notification is clicked.

6	EntityId	Guid?	Public	Related entity ID
7	UserId	Guid?	Public	ID of the user who receives this notification.
8	Type	NotificationType?	Public	Type/category of the notification
9	User	User?	Public	Navigation property to the related User entity.

3.1.1.41 FinancialGoal class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid?	Public	ID of the user who owns the financial goal.
2	GroupId	Guid?	Public	Optional ID of the group associated with the goal.
3	SubcategoryId	Guid?	Public	Optional subcategory linked to the goal.
4	Name	string	Public	Name/title of the financial goal.
5	NameUnsign	string	Public	Name without diacritics.
6	TargetAmount	decimal	Public	The total amount to be saved or achieved.
7	CurrentAmount	decimal	Public	Current saved amount toward the goal.
8	StartDate	DateTime	Public	Date the goal was started.
9	Deadline	DateTime	Public	Deadline to reach the financial goal.
10	Status	FinancialGoalStatus	Public	Current status of the goal.
11	ApprovalStatus	ApprovalStatus	Public	Approval status for goals that require validation.
12	IsSaving	bool	Public	Whether the goal is currently in saving mode.
13	Group	GroupFund?	Public	Navigation property to related group, if any.

14	User	User	Public	Navigation property to the associated user.
15	Subcategory	Subcategory?	Public	Navigation property to linked subcategory.

3.1.1.42 NotificationType enum

No	Enum	Visibility	Description
1	SYSTEM	Public	System-generated notification for all users
2	USER	Public	Personal notification targeted to an individual user
3	GROUP	Public	Notification related to group activity
4	GROUP_INVITE	Public	Invitation to join a group
5	FINANCIAL_GOAL_PERSONAL	Public	Notification regarding a personal financial goal
6	FINANCIAL_GOAL_GROUP	Public	Notification related to a group financial goal

3.1.1.43 EntityName enum

No	Enum	Visibility	Description
1	GROUP	Public	Refers to a group entity
2	TRANSACTION	Public	Refers to a transaction entity

3.1.1.44 TransactionType enum

No	Enum	Visibility	Description
1	INCOME	Public	Refers to an income transaction
2	EXPENSE	Public	Refers to an expense transaction

3.1.1.45 ApprovalStatus enum

No	Enum	Visibility	Description
1	PENDING	Public	The entity is awaiting approval.
2	APPROVED	Public	The entity has been approved.

3	REJECTED	Public	The entity has been rejected.

3.1.1.46 FinancialGoalStatus enum

No	Enum	Visibility	Description
1	PENDING	Public	The financial goal is awaiting action or approval.
2	ACTIVE	Public	The financial goal is currently being pursued or worked on.
3	ARCHIVED	Public	The financial goal has been archived and is no longer active.
4	COMPLETED	Public	The financial goal has been successfully achieved.

3.1.1.47 BaseResultModel class

No	Attribute	Data Type	Visibility	Description
1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies the specific error if any occurred.
3	Message	string	Public	A message that describes the result or error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.1.2 Sequence Diagram

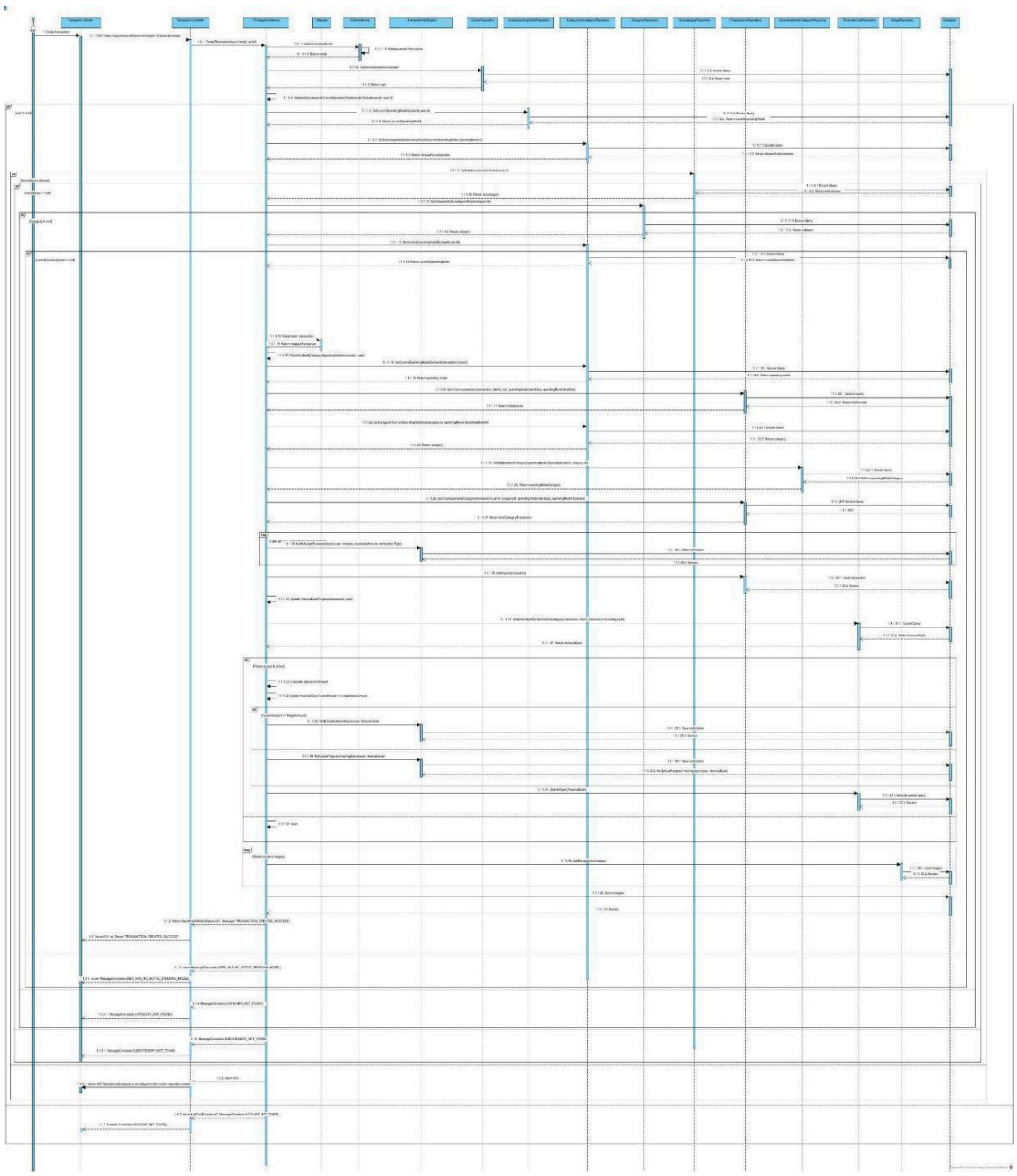


Figure 15 - Sequence Diagram

3.2 Update Transaction

3.2.1 Class Diagram

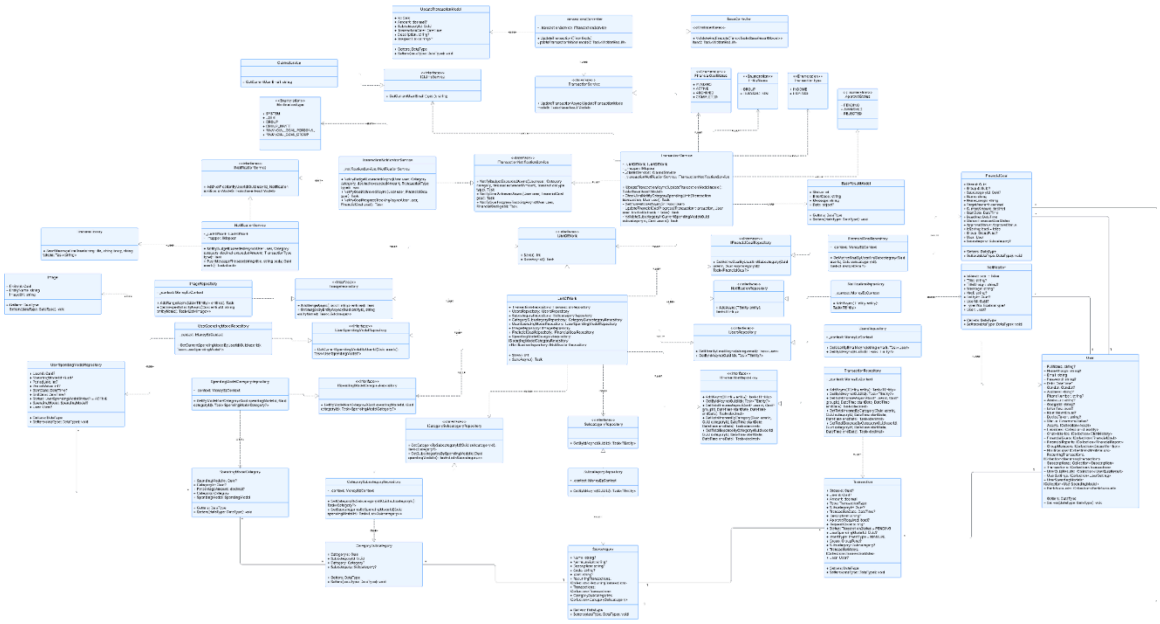


Figure 16 - Class Diagram

3.2.1.1 TransactionsController class

No	Attribute	Data Type	Visibility	Description
1	_transactionService	ITransactionService	Private	Service responsible for processing transaction logic.

No	Method	Data Type	Visibility	Description
1	UpdateTransaction	Task<IActionResult>	Public	Updates an existing transaction with the provided details.

3.2.1.2 BaseController class

No	Method	Data Type	Visibility	Description
1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.2.1.3 UpdateTransactionModel class

No	Attribute	Data Type	Visibility	Description
1	Id	Guid	Public	The unique identifier for the transaction.

2	Amount	decimal?	Public	The amount of the transaction.
3	SubcategoryId	Guid	Public	The unique identifier for the subcategory related to the transaction.
4	TransactionDate	DateTime	Public	The date and time when the transaction occurred.
5	Description	string?	Public	A description of the transaction.
6	Images	List<string>?	Public	A list of image URLs related to the transaction.

3.2.1.4 ITransactionService class

No	Method	Data Type	Visibility	Description
1	UpdateTransactionAsync	Task<BaseResultModel>	Public	Asynchronously updates the transaction with the provided details.

3.2.1.5 TransactionService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.
3	_transactionNotificationService	ITransactionNotificationService	Private	Sends transaction-related notifications.
4	_claimsService	IClaimsService	Private	Service for handling user claims extracted from the token.

No	Method	Data Type	Visibility	Description
1	UpdateTransactionAsync	Task<BaseResultModel>	Public	Asynchronously updates the transaction with the provided details.

2	CheckAndNotifyCategorySpendingLimit	Task	Private	Checks the spending limit for a category and notifies if exceeded.
3	GetCurrentUserAsync	Task<User>	Public	Asynchronously retrieves the current logged-in user's information.
4	UpdateFinancialGoalProgress	Task	Private	Updates the progress of the user's financial goal.
5	ValidateSubcategoryInCurrentSpendingModel	Task	Private	Validates if the subcategory is within the current spending model.

3.2.1.6 ClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Email extracted from current user claims.

3.2.1.7 IClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Gets the email of the current user.

3.2.1.8 ITransactionNotificationService class

No	Method	Data Type	Visibility	Description
1	NotifyBudgetExceededAsync	Task	Public	Notifies user when their budget limit is exceeded.
2	NotifyGoalAchievedAsync	Task	Public	Notifies user when a financial goal is achieved.
3	NotifyGoalProgressTrackingAsync	Task	Public	Notifies user about the progress of their financial goal.

3.2.1.9 TransactionNotificationService class

No	Attribute	Data Type	Visibility	Description
1	_notificationService	INotificationService	Private	Service for sending notifications to the user.

No	Method	Data Type	Visibility	Description
1	NotifyBudgetExceededAsync	Task	Public	Notifies user when their budget limit is exceeded.
2	NotifyGoalAchievedAsync	Task	Public	Notifies user when a financial goal is achieved.
3	NotifyGoalProgressTrackingAsync	Task	Public	Notifies user about the progress of their financial goal.

3.2.1.10 INotificationService class

No	Method	Data Type	Visibility	Description
1	AddNotificationByUserId	Task<BaseResultModel>	Public	Adds a notification for a user by their ID.

3.2.1.11 NotificationService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.

No	Method	Data Type	Visibility	Description
1	NotifyBudgetExceededAsync	Task	Public	Notifies user when their budget limit is exceeded.
2	PushMessageFirebase	Task<bool>	Public	Sends a push notification to the user via Firebase.

3.2.1.12 FirebaseLibrary class

No	Method	Data Type	Visibility	Description
1	SendMessageFireBase	Task<string>	Public	Sends a Firebase message with the specified title, body, and token.

3.2.1.13 IUnitOfWork class

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.2.1.14 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	TransactionsRepository	ITransactionRepository	Public	Repository for managing transaction data.
2	UsersRepository	IUsersRepository	Public	Repository for managing user data.
3	SubcategoryRepository	ISubcategoryRepository	Public	Repository for managing subcategory data.
4	CategorySubcategoryRepository	ICategorySubcategoryRepository	Public	Repository for managing category-subcategory relationships.
5	UserSpendingModelRepository	IUserSpendingModelRepository	Public	Repository for managing user spending models.
6	ImageRepository	IImageRepository	Public	Repository for managing image data.
7	FinancialGoalRepository	IFinancialGoalRepository	Public	Repository for managing financial goal data.
8	SpendingModelCategoryRepository	ISpendingModelCategoryRepository	Public	Repository for managing spending model-category relationships.
9	NotificationRepository	INotificationRepository	Public	Repository for managing notification data.

No	Method	Data Type	Visibility	Description

1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.2.1.15 IImageRepository class

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.
2	GetImagesByEntityAsync	Task<List<Image>>	Public	Asynchronously retrieves a list of images associated with a specific entity.

3.2.1.16 IUserSpendingModelRepository class

No	Method	Data Type	Visibility	Description
1	GetCurrentSpendingModelByUserId	Task<UserSpendingModel?>	Public	Asynchronously retrieves the current spending model for a user by their ID.

3.2.1.17 ISpendingModelCategoryRepository class

No	Method	Data Type	Visibility	Description
1	GetByModelAndCategory	Task<SpendingModelCategory?>	Public	Asynchronously retrieves the spending model category by its spending model and category IDs.

3.2.1.18 ICategorySubcategoryRepository class

No	Method	Data Type	Visibility	Description
1	GetCategoryBySubcategoryId	Task<Category?>	Public	Asynchronously retrieves the category associated with a subcategory ID.
2	GetSubcategoriesBySpendingModelId	Task<List<Subcategory>>	Public	Asynchronously retrieves a list of subcategories associated with a spending model.

3.2.1.19 ISubcategoryRepository class

No	Method	Data Type	Visibility	Description
1	GetByIdAsync	Task<TEntity>	Public	Asynchronously retrieves an entity by its ID.

3.2.1.20 ITransactionRepository class

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity and returns the added entity.
2	GetTotalIncomeAsync	Task<decimal>	Public	Asynchronously calculates the total income for a user or group within a date range.
3	GetTotalIncomeByCategory	Task<decimal>	Public	Asynchronously calculates the total income for a user and category within a date range.
4	GetTotalExpenseByCategory	Task<decimal>	Public	Asynchronously calculates the total expense for a user and category within a date range.
5	GetByIdAsync	Task<TEntity>	Public	Asynchronously retrieves an entity by its ID.

3.2.1.21 IUsersRepository class

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.
2	GetByIdAsync	Task<TEntity?>	Public	Asynchronously retrieves an entity by its ID, returning null if not found.

3.2.1.22 INotificationRepository class

No	Method	Data Type	Visibility	Description

1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity to the repository and returns the added entity.

3.2.1.23 IFinancialGoalRepository class

No	Method	Data Type	Visibility	Description
1	GetActiveGoalByUserAndSubcategory	Task<FinancialGoal?>	Public	Asynchronously retrieves the active financial goal for a user and subcategory.

3.2.1.24 FinancialGoalRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetActiveGoalByUserAndSubcategory	Task<FinancialGoal?>	Public	Asynchronously retrieves the active financial goal for a user and subcategory.

3.2.1.25 NotificationRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity to the repository and returns the added entity.

3.2.1.26 UsersRepository class

No	Attribute	Data Type	Visibility	Description

1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.
2	GetByIdAsync	Task<TEntity?>	Public	Asynchronously retrieves an entity by its ID, returning null if not found.

3.2.1.27 TransactionRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity and returns the added entity.
2	GetTotalIncomeAsync	Task<decimal>	Public	Asynchronously calculates the total income for a user or group within a date range.
3	GetTotalIncomeByCategory	Task<decimal>	Public	Asynchronously calculates the total income for a user and category within a date range.
4	GetTotalExpenseByCategory	Task<decimal>	Public	Asynchronously calculates the total expense for a user and category within a date range.

5	GetByIdAsync	Task<TEntity?>	Public	Asynchronously retrieves an entity by its ID, returning null if not found.

3.2.1.28 SubcategoryRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByIdAsync	Task<TEntity>	Public	Asynchronously retrieves an entity by its ID.

3.2.1.29 CategorySubcategoryRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetCategoryBySubcategoryId	Task<Category?>	Public	Asynchronously retrieves the category associated with a subcategory ID.
2	GetSubcategoriesBySpendingModelId	Task<List<Subcategory>>	Public	Asynchronously retrieves a list of subcategories associated with a spending model.

3.2.1.30 SpendingModelCategoryRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByModelAndCategory	Task<SpendingModelCategory?>	Public	Asynchronously retrieves the spending model category by its spending model and category IDs.

3.2.1.31 ImageRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.
2	GetImagesByEntityAsync	Task<List<Image>>	Public	Asynchronously retrieves a list of images associated with a specific entity.

3.2.1.32 UserSpendingModelRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetCurrentSpendingModelByUserId	Task<UserSpendingModel?>	Public	Asynchronously retrieves the current spending model for a user by their ID.

3.2.1.33 Image class

No	Attribute	Data Type	Visibility	Description

1	EntityId	Guid	Public	The unique identifier for the entity.
2	EntityName	string	Public	The name of the entity.
3	ImageUrl	string	Public	The URL of the image associated with the entity.

3.2.1.34 UserSpendingModel class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid?	Public	The unique identifier for the user.
2	SpendingModelId	Guid?	Public	The unique identifier for the spending model.
3	PeriodUnit	int?	Public	The unit of the period.
4	PeriodValue	int?	Public	The value of the period.
5	StartDate	DateTime?	Public	The start date of the spending model.
6	EndDate	DateTime?	Public	The end date of the spending model.
7	Status	UserSpending ModelStatus? = ACTIVE	Public	The status of the user spending model, default is ACTIVE.
8	SpendingModel	SpendingMode l?	Public	The spending model associated with the user.
9	User	User?	Public	The user associated with the spending model.

3.2.1.35 SpendingModelCategory class

No	Attribute	Data Type	Visibility	Description
1	SpendingModelId	Guid?	Public	The ID of the associated spending model.
2	CategoryId	Guid?	Public	The ID of the associated category.
3	PercentageAmount	decimal?	Public	The allocated percentage amount for the category.

4	Category	Category	Public	The category linked to the spending model.
5	SpendingModel	SpendingModel	Public	The spending model that includes this category.

3.2.1.36 CategorySubcategory class

No	Attribute	Data Type	Visibility	Description
1	CategoryId	Guid?	Public	The ID of the associated category.
2	SubcategoryId	Guid?	Public	The ID of the associated subcategory.
3	Category	Category?	Public	Navigation property to the category.
4	Subcategory	Subcategory?	Public	Navigation property to the subcategory.

3.2.1.37 Subcategory class

No	Attribute	Data Type	Visibility	Description
1	Name	string?	Public	The name of the subcategory
2	NameUnsign	string?	Public	The unsigned version of the name
3	Description	string?	Public	A short description of the subcategory
4	Code	string?	Public	The code representing the subcategory
5	Icon	string?	Public	Icon associated with the subcategory
6	RecurringTransactions	ICollection<RecurringTransaction>	Public	Related recurring transactions.
7	Transactions	ICollection<Transaction>	Public	Related one-time transactions.
8	CategorySubcategories	ICollection<CategorySubcategory>	Public	Links to categories through relationships.

3.2.1.38 Transaction class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid?	Public	ID of the group fund.

2	UserId	Guid?	Public	ID of the user who made the transaction.
3	Amount	decimal	Public	The amount of the transaction.
4	Type	TransactionType	Public	Type of transaction.
5	SubcategoryId	Guid?	Public	ID of the subcategory linked to this transaction.
6	TransactionDate	DateTime?	Public	The date the transaction occurred.
7	Description	string?	Public	Description of the transaction.
8	ApprovalRequired	bool?	Public	Whether approval is required.
9	RequestCode	string?	Public	Unique request code for approval tracking.
10	Status	TransactionStatus	Public	Current status of the transaction.
11	UserSpendingModelId	Guid?	Public	Related user's spending model ID.
12	InsertType	InsertType	Public	Indicates how the transaction was added.
13	Group	GroupFund?	Public	Navigation property to the related group.
14	Subcategory	Subcategory?	Public	Navigation to the subcategory.
15	TransactionVotes	ICollection<TransactionVote>	Public	Related transaction votes for approval.
16	User	User?	Public	Navigation property to the user

3.2.1.39 User class

No	Attribute	Data Type	Visibility	Description
1	FullName	string?	Public	Full name of the user.
2	NameUnsign	string?	Public	Full name without diacritics
3	Email	string?	Public	Email address of the user.
4	Password	string?	Public	User's password (if not using Google login).

5	Dob	DateTime?	Public	Date of birth.
6	Gender	Gender?	Public	Gender of the user.
7	Address	string?	Public	User’s address.
8	PhoneNumber	string?	Public	User’s phone number.
9	AvatarUrl	string?	Public	URL to the user's avatar image.
10	GoogleId	string?	Public	Google account ID (if signed in with Google).
11	IsVerified	bool?	Public	Whether the user's email is verified.
12	Role	RolesEnum?	Public	Role of the user
13	DeviceToken	string?	Public	Device token for push notifications.
14	Status	CommonsStatus?	Public	Current status of the user.
15	Assets	ICollection<Asset>	Public	List of user's assets.
16	Liabilities	ICollection<Liability>	Public	List of user's liabilities.
17	ChatHistories	ICollection<ChatHistory>	Public	Chat history records.
18	FinancialGoals	ICollection<FinancialGoal>	Public	User’s financial goals.
19	FinancialReports	ICollection<FinancialReport>	Public	Generated financial reports.
20	GroupMembers	ICollection<GroupMember>	Public	User’s membership in groups.
21	Notifications	ICollection<Notification>	Public	Notifications received by the user.
22	RecurringTransactions	ICollection<RecurringTransaction>	Public	Scheduled recurring transactions.
23	Subscriptions	ICollection<Subscription>	Public	User’s active subscriptions.
24	Transactions	ICollection<Transaction>	Public	All transactions linked to the user.
25	UserQuizResults	ICollection<UserQuizResult>	Public	Results from financial quizzes.

26	UserSettings	ICollection<UserSetting>	Public	Custom user settings/preferences.
27	UserSpendingModels	ICollection<UserSpendingModel>	Public	Linked spending models.
28	BankAccounts	ICollection<BankAccount>	Public	Bank accounts linked to the user.

3.2.1.40 Notification class

No	Attribute	Data Type	Visibility	Description
1	IsRead	bool	Public	Indicates whether the notification has been read.
2	Title	string?	Public	Title of the notification.
3	TitleUnsign	string?	Public	Title without diacritics (for search and filtering).
4	Message	string?	Public	Content/message of the notification.
5	Href	string?	Public	Link to redirect when the notification is clicked.
6	EntityId	Guid?	Public	Related entity ID
7	UserId	Guid?	Public	ID of the user who receives this notification.
8	Type	NotificationType?	Public	Type/category of the notification
9	User	User?	Public	Navigation property to the related User entity.

3.2.1.41 FinancialGoal class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid?	Public	ID of the user who owns the financial goal.
2	GroupId	Guid?	Public	Optional ID of the group associated with the goal.
3	SubcategoryId	Guid?	Public	Optional subcategory linked to the goal.

4	Name	string	Public	Name/title of the financial goal.
5	NameUnsign	string	Public	Name without diacritics.
6	TargetAmount	decimal	Public	The total amount to be saved or achieved.
7	CurrentAmount	decimal	Public	Current saved amount toward the goal.
8	StartDate	DateTime	Public	Date the goal was started.
9	Deadline	DateTime	Public	Deadline to reach the financial goal.
10	Status	FinancialGoalStatus	Public	Current status of the goal.
11	ApprovalStatus	ApprovalStatus	Public	Approval status for goals that require validation.
12	IsSaving	bool	Public	Whether the goal is currently in saving mode.
13	Group	GroupFund?	Public	Navigation property to related group, if any.
14	User	User	Public	Navigation property to the associated user.
15	Subcategory	Subcategory?	Public	Navigation property to linked subcategory.

3.2.1.42 NotificationType enum

No	Enum	Visibility	Description
1	SYSTEM	Public	System-generated notification for all users
2	USER	Public	Personal notification targeted to an individual user
3	GROUP	Public	Notification related to group activity
4	GROUP_INVITE	Public	Invitation to join a group
5	FINANCIAL_GOAL_PERSONAL	Public	Notification regarding a personal financial goal
6	FINANCIAL_GOAL_GROUP	Public	Notification related to a group financial goal

3.2.1.43 EntityName enum

No	Enum	Visibility	Description
1	GROUP	Public	Refers to a group entity
2	TRANSACTION	Public	Refers to a transaction entity

3.2.1.44 TransactionType enum

No	Enum	Visibility	Description
1	INCOME	Public	Refers to an income transaction
2	EXPENSE	Public	Refers to an expense transaction

3.2.1.45 ApprovalStatus enum

No	Enum	Visibility	Description
1	PENDING	Public	The entity is awaiting approval.
2	APPROVED	Public	The entity has been approved.
3	REJECTED	Public	The entity has been rejected.

3.2.1.46 FinancialGoalStatus enum

No	Enum	Visibility	Description
1	PENDING	Public	The financial goal is awaiting action or approval.
2	ACTIVE	Public	The financial goal is currently being pursued or worked on.
3	ARCHIVED	Public	The financial goal has been archived and is no longer active.
4	COMPLETED	Public	The financial goal has been successfully achieved.

3.2.1.47 BaseResultModel class

No	Attribute	Data Type	Visibility	Description
1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies the specific error if

				any occurred.
3	Message	string	Public	A message that describes the result or error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.2.2 Sequence Diagram

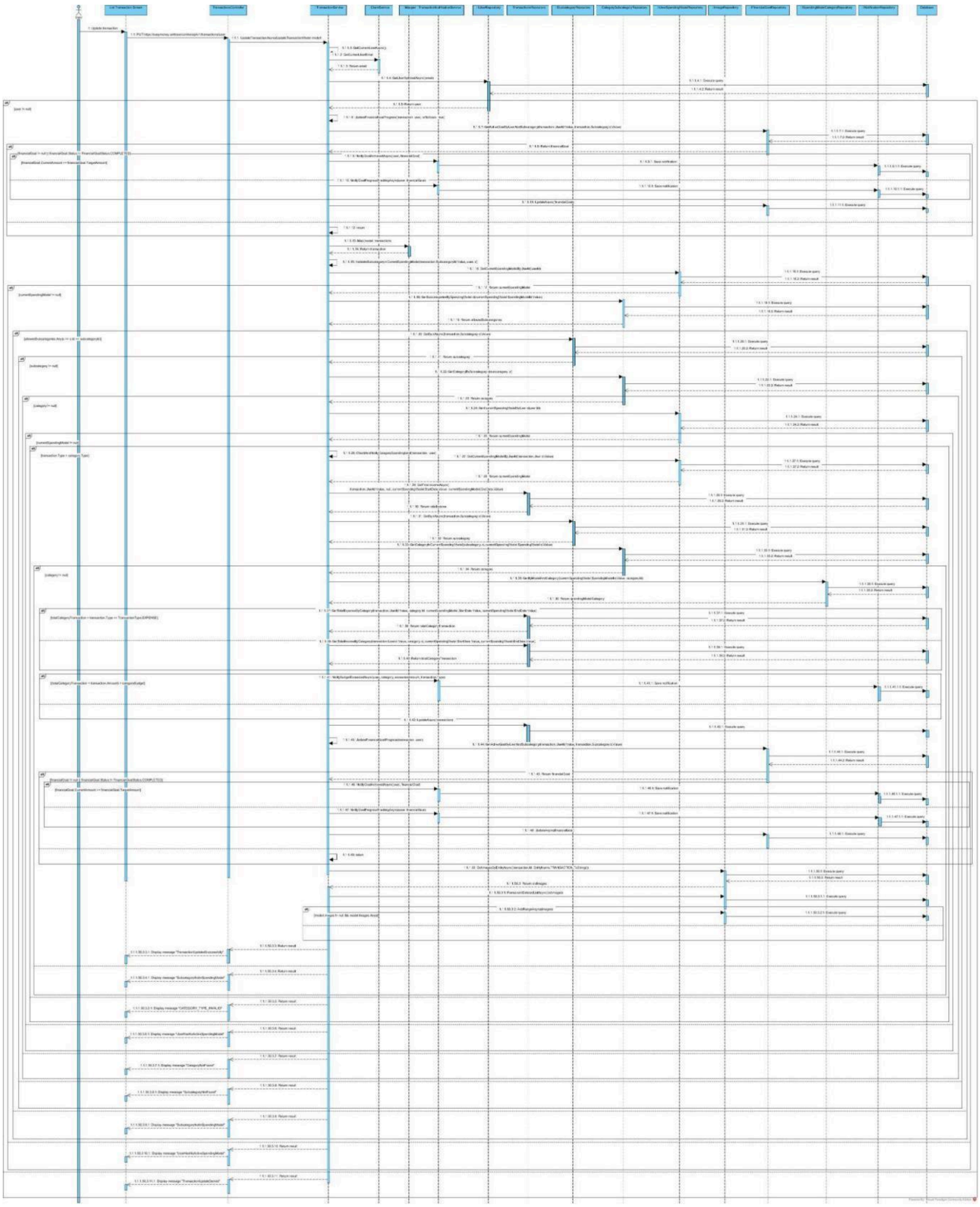


Figure 17 - Sequence Diagram

3.3 Delete Transaction

3.3.1 Class Diagram

Figure 18 - Class Diagram

3.3.1.1 TransactionsController class

No	Attribute	Data Type	Visibility	Description
1	_transactionService	ITransactionService	Private	Service responsible for processing transaction logic.

No	Method	Data Type	Visibility	Description
1	DeleteTransaction	Task<IActionResult>	Public	Asynchronously deletes a transaction by its ID.

3.3.1.2 BaseController class

No	Method	Data Type	Visibility	Description
1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.3.1.3 ITransactionService class

No	Method	Data Type	Visibility	Description
1	DeleteTransactionAsync	Task<BaseResultModel>	Public	Deletes a transaction by ID asynchronously.

3.3.1.4 TransactionService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.
3	_transactionNotificationService	ITransactionNotificationService	Private	Sends transaction-related notifications.
4	_claimsService	IClaimsService	Private	Service for handling user claims extracted from the token.

No	Method	Data Type	Visibility	Description
1	DeleteTransactionAsync	Task<BaseResultModel>	Public	Deletes a transaction by its ID.
2	GetCurrentUserAsync	Task<User>	Private	Retrieves the current authenticated user.
3	UpdateFinancialGoalProgress	Task	Private	Updates or rolls back the progress of a financial goal.

3.3.1.5 ClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Email extracted from current user claims.

3.3.1.6 IClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Gets the email of the current user.

3.3.1.7 ITransactionNotificationService class

No	Method	Data Type	Visibility	Description
1	NotifyGoalAchievedAsync	Task	Public	Notifies user when a financial goal is achieved.
2	NotifyGoalProgressTrackingAsync	Task	Public	Notifies user about the progress of their financial goal.

3.3.1.8 TransactionNotificationService class

No	Attribute	Data Type	Visibility	Description
1	_notificationService	INotificationService	Private	Service for sending notifications to the user.

No	Method	Data Type	Visibility	Description
1	NotifyGoalAchievedAsync	Task	Public	Notifies user when a financial goal is achieved.
2	NotifyGoalProgressTrackingAsync	Task	Public	Notifies user about the progress of their financial goal.

3.3.1.9 INotificationService class

No	Method	Data Type	Visibility	Description
1	AddNotificationByUserId	Task<BaseResultModel>	Public	Adds a notification for a user by their ID.

3.3.1.10 NotificationService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.

No	Method	Data Type	Visibility	Description
1	PushMessageFirebase	Task<bool>	Public	Sends a push notification to the user via Firebase.

3.3.1.11 FirebaseLibrary class

No	Method	Data Type	Visibility	Description
1	SendMessageFireBase	Task<string>	Public	Sends a Firebase message with the specified title, body, and token.

3.3.1.12 IUnitOfWork class

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.3.1.13 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	TransactionsRepository	ITransactionRepository	Public	Repository for managing transaction data.
2	UsersRepository	IUsersRepository	Public	Repository for managing user data.
3	ImageRepository	IImageRepository	Public	Repository for managing image data.
4	FinancialGoalRepository	IFinancialGoalRepository	Public	Repository for managing financial goal data.
5	NotificationRepository	INotificationRepository	Public	Repository for managing notification data.

No	Method	Data Type	Visibility	Description

1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.3.1.14 IImageRepository class

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.
2	GetImagesByEntityAsync	Task<List<Image>>	Public	Asynchronously retrieves a list of images associated with a specific entity.

3.3.1.15 ITransactionRepository class

No	Method	Data Type	Visibility	Description
1	PermanentDeletedAsync	void	Public	Permanently deletes an entity.
2	GetByIdAsync	Task<TEntity>	Public	Asynchronously retrieves an entity by its ID.

3.3.1.16 IUsersRepository class

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.
2	GetByIdAsync	Task<TEntity?>	Public	Asynchronously retrieves an entity by its ID, returning null if not found.

3.3.1.17 INotificationRepository class

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity to the repository and returns the added entity.

3.3.1.18 IFinancialGoalRepository class

No	Method	Data Type	Visibility	Description

1	GetActiveGoalByUserAndSubcategory	Task<FinancialGoal?>	Public	Asynchronously retrieves the active financial goal for a user and subcategory.

3.3.1.19 UsersRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.
2	GetByIdAsync	Task<TEntity?>	Public	Asynchronously retrieves an entity by its ID, returning null if not found.

3.3.1.20 TransactionRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	PermanentDeletedAsync	void	Public	Permanently deletes an entity.
2	GetByIdAsync	Task<TEntity?>	Public	Asynchronously retrieves an entity by its ID, returning null if not found.

3.3.1.21 FinancialGoalRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the

				database.

No	Method	Data Type	Visibility	Description
1	GetActiveGoalByUserAndSubcategory	Task<FinancialGoal?>	Public	Asynchronously retrieves the active financial goal for a user and subcategory.

3.3.1.22 NotificationRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity to the repository and returns the added entity.

3.3.1.23 ImageRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.
2	GetImagesByEntityAsync	Task<List<Image>>	Public	Asynchronously retrieves a list of images associated with a specific entity.

3.3.1.24 Image class

No	Attribute	Data Type	Visibility	Description
1	EntityId	Guid	Public	The unique identifier for the entity.
2	EntityName	string	Public	The name of the entity.
3	ImageUrl	string	Public	The URL of the image associated with the entity.

3.3.1.25 Transaction class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid?	Public	ID of the group fund.
2	UserId	Guid?	Public	ID of the user who made the transaction.
3	Amount	decimal	Public	The amount of the transaction.
4	Type	TransactionType	Public	Type of transaction.
5	SubcategoryId	Guid?	Public	ID of the subcategory linked to this transaction.
6	TransactionDate	DateTime?	Public	The date the transaction occurred.
7	Description	string?	Public	Description of the transaction.
8	ApprovalRequired	bool?	Public	Whether approval is required.
9	RequestCode	string?	Public	Unique request code for approval tracking.
10	Status	TransactionStatus	Public	Current status of the transaction.
11	UserSpendingModelId	Guid?	Public	Related user's spending model ID.
12	InsertType	InsertType	Public	Indicates how the transaction was added.
13	Group	GroupFund?	Public	Navigation property to the related group.
14	Subcategory	Subcategory?	Public	Navigation to the subcategory.

15	TransactionVotes	ICollection<TransactionVote>	Public	Related transaction votes for approval.
16	User	User?	Public	Navigation property to the user

3.3.1.26 User class

No	Attribute	Data Type	Visibility	Description
1	FullName	string?	Public	Full name of the user.
2	NameUnsign	string?	Public	Full name without diacritics
3	Email	string?	Public	Email address of the user.
4	Password	string?	Public	User's password (if not using Google login).
5	Dob	DateTime?	Public	Date of birth.
6	Gender	Gender?	Public	Gender of the user.
7	Address	string?	Public	User’s address.
8	PhoneNumber	string?	Public	User’s phone number.
9	AvatarUrl	string?	Public	URL to the user's avatar image.
10	GoogleId	string?	Public	Google account ID (if signed in with Google).
11	IsVerified	bool?	Public	Whether the user's email is verified.
12	Role	RolesEnum?	Public	Role of the user
13	DeviceToken	string?	Public	Device token for push notifications.
14	Status	CommonsStatus?	Public	Current status of the user.
15	Assets	ICollection<Asset>	Public	List of user's assets.
16	Liabilities	ICollection<Liability>	Public	List of user's liabilities.
17	ChatHistories	ICollection<ChatHistory>	Public	Chat history records.

18	FinancialGoals	ICollection<FinancialGoal>	Public	User's financial goals.
19	FinancialReports	ICollection<FinancialReport>	Public	Generated financial reports.
20	GroupMembers	ICollection<GroupMember>	Public	User's membership in groups.
21	Notifications	ICollection<Notification>	Public	Notifications received by the user.
22	RecurringTransactions	ICollection<RecurringTransaction>	Public	Scheduled recurring transactions.
23	Subscriptions	ICollection<Subscription>	Public	User's active subscriptions.
24	Transactions	ICollection<Transaction>	Public	All transactions linked to the user.
25	UserQuizResults	ICollection<UserQuizResult>	Public	Results from financial quizzes.
26	UserSettings	ICollection<UserSetting>	Public	Custom user settings/preferences.
27	UserSpendingModels	ICollection<UserSpendingModel>	Public	Linked spending models.
28	BankAccounts	ICollection<BankAccount>	Public	Bank accounts linked to the user.

3.3.1.27 Notification class

No	Attribute	Data Type	Visibility	Description
1	IsRead	bool	Public	Indicates whether the notification has been read.
2	Title	string?	Public	Title of the notification.
3	TitleUnsign	string?	Public	Title without diacritics (for search and filtering).
4	Message	string?	Public	Content/message of the notification.
5	Href	string?	Public	Link to redirect when the notification is clicked.
6	EntityId	Guid?	Public	Related entity ID

7	UserId	Guid?	Public	ID of the user who receives this notification.
8	Type	NotificationType?	Public	Type/category of the notification
9	User	User?	Public	Navigation property to the related User entity.

3.3.1.28 FinancialGoal class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid?	Public	ID of the user who owns the financial goal.
2	GroupId	Guid?	Public	Optional ID of the group associated with the goal.
3	SubcategoryId	Guid?	Public	Optional subcategory linked to the goal.
4	Name	string	Public	Name/title of the financial goal.
5	NameUnsign	string	Public	Name without diacritics.
6	TargetAmount	decimal	Public	The total amount to be saved or achieved.
7	CurrentAmount	decimal	Public	Current saved amount toward the goal.
8	StartDate	DateTime	Public	Date the goal was started.
9	Deadline	DateTime	Public	Deadline to reach the financial goal.
10	Status	FinancialGoalStatus	Public	Current status of the goal.
11	ApprovalStatus	ApprovalStatus	Public	Approval status for goals that require validation.
12	IsSaving	bool	Public	Whether the goal is currently in saving mode.
13	Group	GroupFund?	Public	Navigation property to related group, if any.
14	User	User	Public	Navigation property to the associated user.

15	Subcategory	Subcategory?	Public	Navigation property to linked subcategory.

3.3.1.29 EntityName enum

No	Enum	Visibility	Description
1	GROUP	Public	Refers to a group entity
2	TRANSACTION	Public	Refers to a transaction entity

3.3.1.30 ApprovalStatus enum

No	Enum	Visibility	Description
1	PENDING	Public	The entity is awaiting approval.
2	APPROVED	Public	The entity has been approved.
3	REJECTED	Public	The entity has been rejected.

3.3.1.31 FinancialGoalStatus enum

No	Enum	Visibility	Description
1	PENDING	Public	The financial goal is awaiting action or approval.
2	ACTIVE	Public	The financial goal is currently being pursued or worked on.
3	ARCHIVED	Public	The financial goal has been archived and is no longer active.
4	COMPLETED	Public	The financial goal has been successfully achieved.

3.3.1.32 NotificationType enum

No	Enum	Visibility	Description
1	SYSTEM	Public	System-generated notification for all users
2	USER	Public	Personal notification targeted to an individual user
3	GROUP	Public	Notification related to group activity
4	GROUP_INVITE	Public	Invitation to join a group

5	FINANCIAL_GOAL_PERSONAL	Public	Notification regarding a personal financial goal
6	FINANCIAL_GOAL_GROUP	Public	Notification related to a group financial goal

3.3.1.33 BaseResultModel class

No	Attribute	Data Type	Visibility	Description
1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies the specific error if any occurred.
3	Message	string	Public	A message that describes the result or error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.3.2 Sequence Diagram

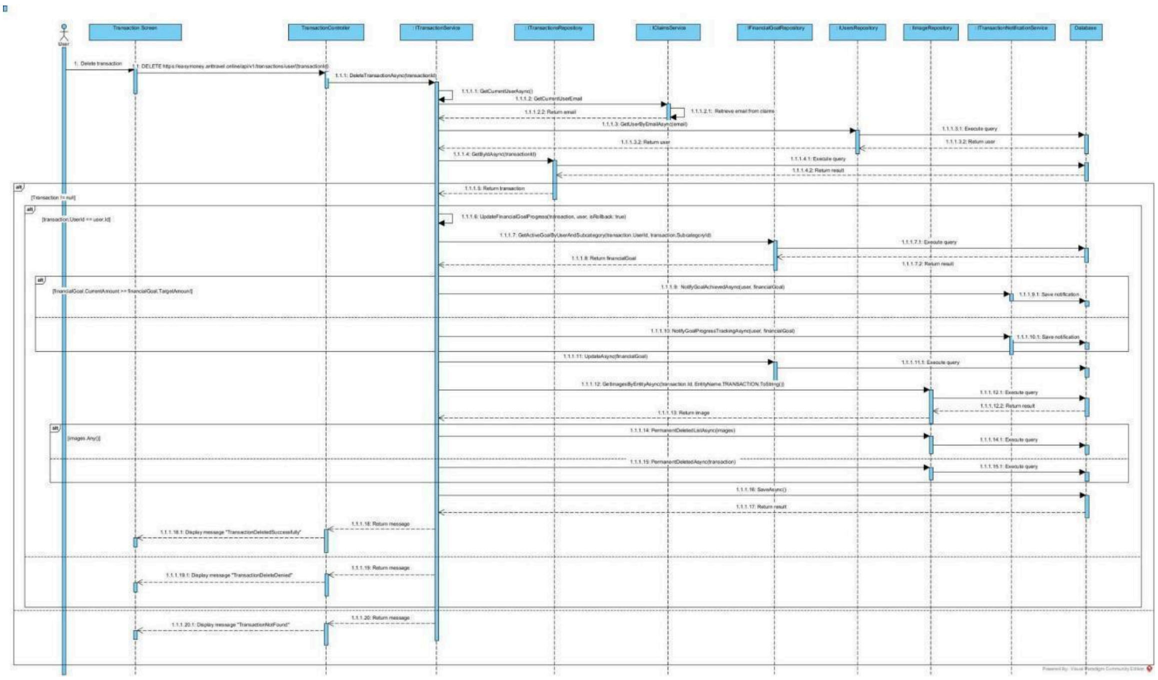


Figure 19 - Sequence Diagram

3.4 Raising fund request

3.4.1 Class Diagram

Figure 20 - Class Diagram

3.4.1.1 GroupController class

No	Attribute	Data Type	Visibility	Description
1	_groupFundsService	IGroupFundsService	Private	Service responsible for processing group fund logic.

No	Method	Data Type	Visibility	Description
1	CreateFundraisingRequest	Task<IActionResult>	Public	Handles the creation of a fundraising request from the given model.

3.4.1.2 BaseController class

No	Method	Data Type	Visibility	Description

1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.4.1.3 CreateFundraisingModel class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid	Public	Unique identifier of the associated group transaction.
2	Amount	decimal	Public	Amount of money involved in the group transaction.
3	Description	string?	Public	Optional details about the group transaction purpose.

3.4.1.4 IGroupFundsService class

No	Method	Data Type	Visibility	Description
1	CreateFundraisingRequest	Task<BaseResultModel>	Public	Creates a new fundraising request with specified group and amount.

3.4.1.5 GroupFundsService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.
3	_transactionNotificationService	ITransactionNotificationService	Private	Sends transaction-related notifications.
4	_claimsService	IClaimsService	Private	Service for handling user claims extracted from the token.

No	Method	Data Type	Visibility	Description
1	CreateFundraisingRequest	Task<BaseResultModel>	Public	Creates a new fundraising request

				with specified group and amount.

3.4.1.6 ClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Email extracted from current user claims.

3.4.1.7 IClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Gets the email of the current user.

3.4.1.8 ITransactionNotificationService class

No	Method	Data Type	Visibility	Description
1	NotifyTransactionApprovalRequestAsync	Task	Public	Sends a notification for transaction approval request in group.
2	NotifyTransactionCreatedAsync	Task	Public	Sends notification when a group transaction is successfully created.

3.4.1.9 TransactionNotificationService class

No	Attribute	Data Type	Visibility	Description
1	_notificationService	INotificationService	Private	Service for sending notifications to the user.

No	Method	Data Type	Visibility	Description
1	NotifyTransactionApprovalRequestAsync	Task	Public	Sends a notification for transaction approval request in group.
2	NotifyTransactionCreatedAsync	Task	Public	Sends notification when a group transaction is successfully created.

3.4.1.10 INotificationService class

No	Method	Data Type	Visibility	Description
1	AddNotificationByUserId	Task<BaseResultModel>	Public	Adds a notification for a user by their ID.

3.4.1.11 NotificationService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.

No	Method	Data Type	Visibility	Description
1	NotifyBudgetExceededAsync	Task	Public	Notifies user when their budget limit is exceeded.
2	PushMessageFirebase	Task<bool>	Public	Sends a push notification to the user via Firebase.

3.4.1.12 FirebaseLibrary class

No	Method	Data Type	Visibility	Description
1	SendMessageFireBase	Task<string>	Public	Sends a Firebase message with the specified title, body, and token.

3.4.1.13 IUnitOfWork class

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.4.1.14 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	TransactionsRepository	ITransactionRepository	Public	Repository for managing transaction

				data.
2	UsersRepository	IUsersRepository	Public	Repository for managing user data.
3	GroupFundRepository	IGroupFundRepository	Public	Repository for managing group fund data.
4	BankAccountRepository	IBankAccount Repository	Public	Repository for managing bank account relationships.
5	ImageRepository	IImageRepository	Public	Repository for managing image data.
6	FinancialGoalRepository	IFinancialGoal Repository	Public	Repository for managing financial goal data.
7	NotificationRepository	INotificationRepository	Public	Repository for managing notification data.

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.4.1.15 IImageRepository class

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.

3.4.1.16 ITransactionRepository class

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.

3.4.1.17 IUsersRepository class

No	Method	Data Type	Visibility	Description

1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.
2	GetByIdAsync	Task<TEntity?>	Public	Asynchronously retrieves an entity by its ID, returning null if not found.

3.4.1.18 INotificationRepository class

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity to the repository and returns the added entity.

3.4.1.19 IFinancialGoalRepository class

No	Method	Data Type	Visibility	Description
1	GetActiveGoalByUserAndSubcategory	Task<FinancialGoal?>	Public	Asynchronously retrieves the active financial goal for a user and subcategory.
2	GetActiveGoalByGroupId	Task<FinancialGoal?>	Public	Retrieves active financial goal associated with a specific group.

3.4.1.20 IBankAccountRepository class

No	Method	Data Type	Visibility	Description
1	GetByIdAsync	Task<TEntity?>	Public	Retrieves entity from the database using the specified ID.

3.4.1.21 IGroupFundRepository class

No	Method	Data Type	Visibility	Description
1	GetByIdIncludeAsync	Task<TEntity?>	Public	Retrieves entity by ID including related data with optional filter.
2	GetByIdAsync	Task<TEntity?>	Public	Retrieves entity from the database using the specified ID.

3.4.1.22 ImageRepository class

No	Attribute	Data Type	Visibility	Description

1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.

3.4.1.23 TransactionRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.

3.4.1.24 UsersRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.
2	GetByIdAsync	Task<TEntity?>	Public	Asynchronously retrieves an entity by its ID, returning null if not found.

3.4.1.25 NotificationRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity to the repository and returns the added entity.

3.4.1.26 FinancialGoalRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetActiveGoalByUserAndSubcategory	Task<FinancialGoal?>	Public	Asynchronously retrieves the active financial goal for a user and subcategory.
2	GetActiveGoalByGroupId	Task<FinancialGoal?>	Public	Retrieves active financial goal associated with a specific group.

3.4.1.27 BankAccountRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description

1	GetByIdAsync	Task<TEntity?>	Public	Retrieves entity from the database using the specified ID.

3.4.1.28 GroupFundRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByIdIncludeAsync	Task<TEntity?>	Public	Retrieves entity by ID including related data with optional filter.
2	GetByIdAsync	Task<TEntity?>	Public	Retrieves entity from the database using the specified ID.

3.4.1.29 Image class

No	Attribute	Data Type	Visibility	Description
1	EntityId	Guid	Public	The unique identifier for the entity.
2	EntityName	string	Public	The name of the entity.
3	ImageUrl	string	Public	The URL of the image associated with the entity.

3.4.1.30 Transaction class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid?	Public	ID of the group fund.
2	UserId	Guid?	Public	ID of the user who made the transaction.
3	Amount	decimal	Public	The amount of the transaction.
4	Type	TransactionType	Public	Type of transaction.
5	SubcategoryId	Guid?	Public	ID of the subcategory linked to this transaction.

6	TransactionDate	DateTime?	Public	The date the transaction occurred.
7	Description	string?	Public	Description of the transaction.
8	ApprovalRequired	bool?	Public	Whether approval is required.
9	RequestCode	string?	Public	Unique request code for approval tracking.
10	Status	TransactionStatus	Public	Current status of the transaction.
11	UserSpendingModelId	Guid?	Public	Related user's spending model ID.
12	InsertType	InsertType	Public	Indicates how the transaction was added.
13	Group	GroupFund?	Public	Navigation property to the related group.
14	Subcategory	Subcategory?	Public	Navigation to the subcategory.
15	TransactionVotes	ICollection<TransactionVote>	Public	Related transaction votes for approval.
16	User	User?	Public	Navigation property to the user

3.4.1.31 User class

No	Attribute	Data Type	Visibility	Description
1	FullName	string?	Public	Full name of the user.
2	NameUnsign	string?	Public	Full name without diacritics
3	Email	string?	Public	Email address of the user.
4	Password	string?	Public	User's password (if not using Google login).
5	Dob	DateTime?	Public	Date of birth.
6	Gender	Gender?	Public	Gender of the user.
7	Address	string?	Public	User’s address.
8	PhoneNumber	string?	Public	User’s phone number.
9	AvatarUrl	string?	Public	URL to the user's

				avatar image.
10	GoogleId	string?	Public	Google account ID (if signed in with Google).
11	IsVerified	bool?	Public	Whether the user's email is verified.
12	Role	RolesEnum?	Public	Role of the user
13	DeviceToken	string?	Public	Device token for push notifications.
14	Status	CommonsStatus?	Public	Current status of the user.
15	Assets	ICollection<Asset>	Public	List of user's assets.
16	Liabilities	ICollection<Liability>	Public	List of user's liabilities.
17	ChatHistories	ICollection<ChatHistory>	Public	Chat history records.
18	FinancialGoals	ICollection<FinancialGoal>	Public	User’s financial goals.
19	FinancialReports	ICollection<FinancialReport>	Public	Generated financial reports.
20	GroupMembers	ICollection<GroupMember>	Public	User’s membership in groups.
21	Notifications	ICollection<Notification>	Public	Notifications received by the user.
22	RecurringTransactions	ICollection<RecurringTransaction>	Public	Scheduled recurring transactions.
23	Subscriptions	ICollection<Subscription>	Public	User’s active subscriptions.
24	Transactions	ICollection<Transaction>	Public	All transactions linked to the user.
25	UserQuizResults	ICollection<UserQuizResult>	Public	Results from financial quizzes.
26	UserSettings	ICollection<UserSetting>	Public	Custom user settings/preferences.
27	UserSpendingModels	ICollection<UserSpendingModel>	Public	Linked spending models.
28	BankAccounts	ICollection<BankAccount>	Public	Bank accounts linked to the user.

3.4.1.32 Notification class

No	Attribute	Data Type	Visibility	Description
1	IsRead	bool	Public	Indicates whether the notification has been read.
2	Title	string?	Public	Title of the notification.
3	TitleUnsign	string?	Public	Title without diacritics (for search and filtering).
4	Message	string?	Public	Content/message of the notification.
5	Href	string?	Public	Link to redirect when the notification is clicked.
6	EntityId	Guid?	Public	Related entity ID
7	UserId	Guid?	Public	ID of the user who receives this notification.
8	Type	NotificationType?	Public	Type/category of the notification
9	User	User?	Public	Navigation property to the related User entity.

3.4.1.33 FinancialGoal class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid?	Public	ID of the user who owns the financial goal.
2	GroupId	Guid?	Public	Optional ID of the group associated with the goal.
3	SubcategoryId	Guid?	Public	Optional subcategory linked to the goal.
4	Name	string	Public	Name/title of the financial goal.
5	NameUnsign	string	Public	Name without diacritics.
6	TargetAmount	decimal	Public	The total amount to be saved or achieved.

7	CurrentAmount	decimal	Public	Current saved amount toward the goal.
8	StartDate	DateTime	Public	Date the goal was started.
9	Deadline	DateTime	Public	Deadline to reach the financial goal.
10	Status	FinancialGoalStatus	Public	Current status of the goal.
11	ApprovalStatus	ApprovalStatus	Public	Approval status for goals that require validation.
12	IsSaving	bool	Public	Whether the goal is currently in saving mode.
13	Group	GroupFund?	Public	Navigation property to related group, if any.
14	User	User	Public	Navigation property to the associated user.
15	Subcategory	Subcategory?	Public	Navigation property to linked subcategory.

3.4.1.34 BankAccount class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid	Public	Represents the identifier of the user who owns the bank account.
2	AccountNumber	string	Public	The account number assigned by the user's bank.
3	BankName	string	Public	The full name of the bank holding the account.
4	BankShortName	string	Public	The abbreviated or short name of the bank.
5	AccountHolderName	string	Public	The name of the person who owns the bank account.
6	Status	CommonsStatus?	Public	Indicates the current status of the bank account.

7	WebhookSecretKey	string?	Public	Secret key used to authenticate webhook calls.
8	WebhookUrl	string?	Public	The URL where webhook events should be sent.
9	User	User?	Public	Navigation property referencing the associated user.

3.4.1.35 GroupFund class

No	Attribute	Data Type	Visibility	Description
1	Name	string	Public	The official name of the group fund.
2	NameUnsign	string	Public	Unaccented version of the group fund name.
3	Description	string	Public	Textual information describing the group fund.
4	CurrentBalance	decimal	Public	Current monetary balance available in the fund.
5	Status	GroupStatus?	Public	The status representing the group's activity state.
6	Visibility	VisibilityEnum?	Public	Defines the access level visibility of the group.
7	AccountBankId	Guid?	Public	Identifier of the linked bank account if any.
8	FinancialGoals	ICollection<FinancialGoal>	Public	List of financial goals related to the group.
9	FinancialReports	ICollection<FinancialReport>	Public	Collection of financial reports for the group.
10	GroupFundLogs	ICollection<GroupFundLog>	Public	Log entries for actions within the group fund.
11	GroupMembers	ICollection<GroupMember>	Public	Members who belong to the group fund.
12	Transactions	ICollection<Transaction>	Public	All transactions performed under the group.

3.4.1.36 EntityName enum

No	Enum	Visibility	Description
1	GROUP	Public	Refers to a group entity
2	TRANSACTION	Public	Refers to a transaction entity

3.4.1.37 NotificationType enum

No	Enum	Visibility	Description
1	SYSTEM	Public	System-generated notification for all users
2	USER	Public	Personal notification targeted to an individual user
3	GROUP	Public	Notification related to group activity
4	GROUP_INVITE	Public	Invitation to join a group
5	FINANCIAL_GOAL_PERSONAL	Public	Notification regarding a personal financial goal
6	FINANCIAL_GOAL_GROUP	Public	Notification related to a group financial goal

3.4.1.38 TransactionType enum

No	Enum	Visibility	Description
1	INCOME	Public	Refers to an income transaction
2	EXPENSE	Public	Refers to an expense transaction

3.4.1.39 RoleGroup enum

No	Enum	Visibility	Description
1	LEADER	Public	The user who created and leads the group.
2	MOD	Public	A moderator who can manage group activities.
3	MEMBER	Public	A regular member with standard group access.

3.4.1.40 TransactionStatus enum

No	Enum	Visibility	Description
1	PENDING	Public	The transaction is waiting for approval or review.
2	APPROVED	Public	The transaction has been reviewed and approved.
3	REJECTED	Public	The transaction has been reviewed and rejected.

3.4.1.41 ITransactionService class

No	Method	Data Type	Visibility	Description
1	CreateGroupTransactionAsync	Task<BaseResultModel>	Public	Creates a new transaction in a group context using user email.

3.4.1.42 TransactionService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.
3	_transactionNotificationService	ITransactionNotificationService	Private	Sends transaction-related notifications.

No	Method	Data Type	Visibility	Description
1	CreateGroupTransactionAsync	Task<BaseResultModel>	Public	Creates a new transaction in a group context using user email.

3.4.1.43 BaseResultModel class

No	Attribute	Data Type	Visibility	Description
1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies the specific error if any occurred.

3	Message	string	Public	A message that describes the result or error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.4.1.44 GroupMemberStatus enum

No	Enum	Visibility	Description
1	ACTIVE	Public	Public The group member is currently active and participating in the group.
2	PENDING	Public	Public The group member has been invited or requested to join but has not yet been approved
3	INACTIVE	Public	The group member is no longer active or has been removed from the group.

3.4.1.45 CreateGroupTransactionModel class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid	Public	Unique identifier for the group associated with the transaction.
2	Amount	decimal	Public	Amount of money involved in the transaction.
3	Type	TransactionType	Public	Type of transaction (Income/Expense).
4	TransactionDate	DateTime	Public	Date when the transaction occurred.
5	Description	string?	Public	A description of the transaction.
6	Images	List<string>?	Public	List of image URLs related to the transaction.
7	RequireVote	bool	Public	Whether the transaction requires a vote for approval.
8	InsertType	InsertType	Public	Type of insertion for the transaction.

9	AccountBankNumber	string?	Public	Bank account number related to the transaction.
10	AccountBankName	string?	Public	Bank name where the transaction was made.
11	BankTransactionId	string?	Public	Bank-specific transaction identifier.
12	BankTransactionDate	DateTime?	Public	Date when the bank transaction was completed.

3.4.2 Sequence Diagram

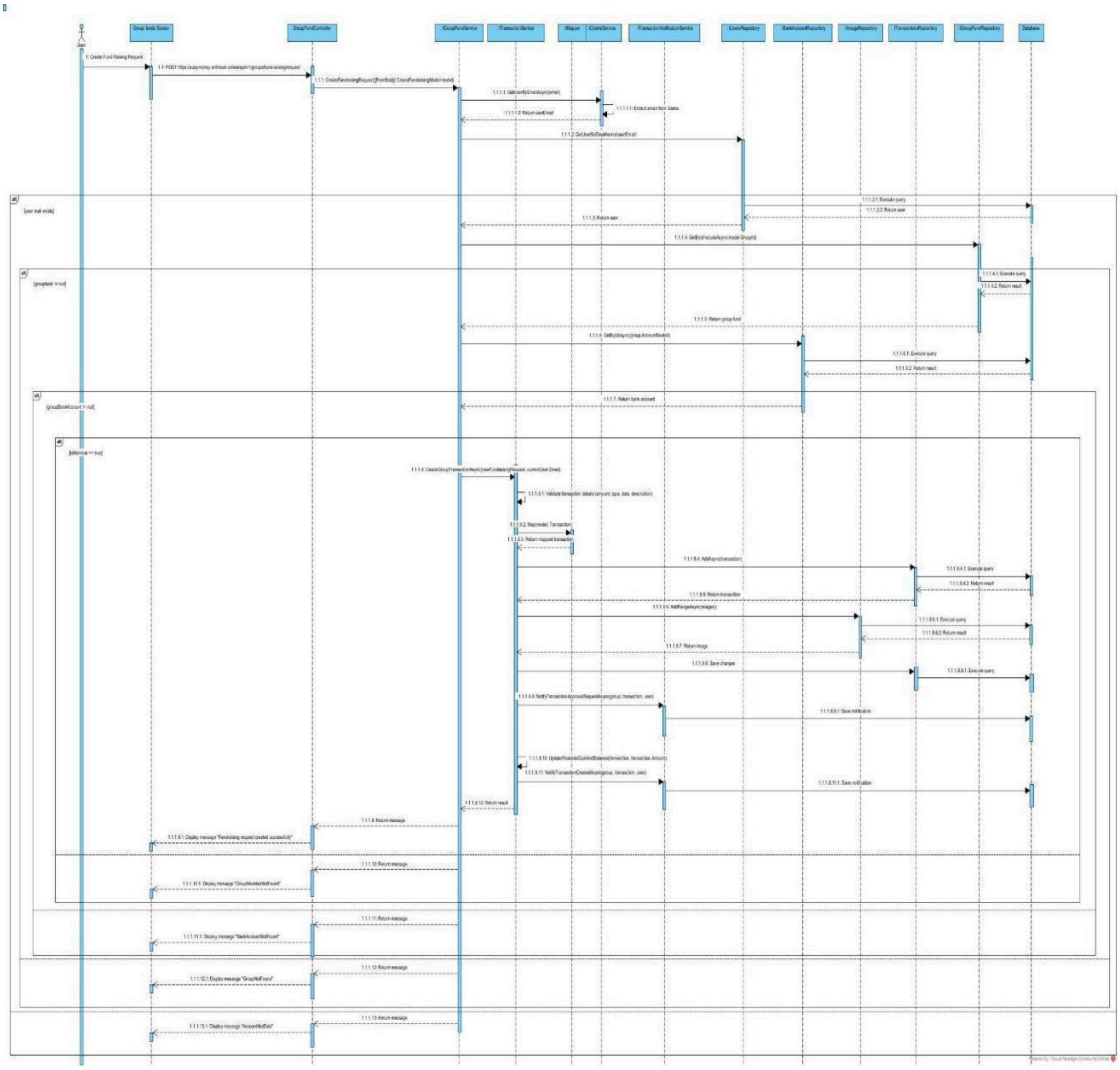


Figure 21 - Sequence Diagram

3.5 Request a withdrawal

3.5.1 Class Diagram

Figure 22 - Class Diagram

3.5.1.1 GroupController class

No	Attribute	Data Type	Visibility	Description
1	_groupFundsService	IGroupFundsService	Private	Service responsible for processing group fund logic.

No	Method	Data Type	Visibility	Description
1	CreateFundWithdrawalRequest	Task<IActionResult>	Public	Handles the creation of a fund withdrawal request.

3.5.1.2 BaseController class

No	Method	Data Type	Visibility	Description
1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.5.1.3 CreateFundWithdrawalModel class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid	Public	Unique identifier of the associated group transaction.
2	Amount	decimal	Public	Amount of money involved in the group transaction.
3	Description	string?	Public	Optional details about the group transaction purpose.
4	Images	List<string>	Public	A list of image URLs related to the withdrawal.

3.5.1.4 IGroupFundsService class

No	Method	Data Type	Visibility	Description
1	CreateFundWithdrawalRequest	Task<BaseResultModel>	Public	Handles the creation of a fund withdrawal request.

3.5.1.5 GroupFundsService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.
3	_transactionNotificationService	ITransactionNotificationService	Private	Sends transaction-related notifications.
4	_claimsService	IClaimsService	Private	Service for handling user claims extracted from the token.

No	Method	Data Type	Visibility	Description
1	CreateFundWithdrawalRequest	Task<BaseResultModel>	Public	Handles the creation of a fund withdrawal request.

3.5.1.6 ClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Email extracted from current user claims.

3.5.1.7 IClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Gets the email of the current user.

3.5.1.8 ITransactionNotificationService class

No	Method	Data Type	Visibility	Description
1	NotifyTransactionApprovalRequestAsync	Task	Public	Sends a notification for transaction approval request in group.
2	NotifyTransactionCreatedAsync	Task	Public	Sends notification when a group transaction is successfully created.

3.5.1.9 TransactionNotificationService class

No	Attribute	Data Type	Visibility	Description
1	_notificationService	INotificationService	Private	Service for sending notifications to the user.

No	Method	Data Type	Visibility	Description
1	NotifyTransactionApprovalRequestAsync	Task	Public	Sends a notification for transaction approval request in group.
2	NotifyTransactionCreatedAsync	Task	Public	Sends notification when a group transaction is successfully created.

3.5.1.10 INotificationService class

No	Method	Data Type	Visibility	Description

1	AddNotificationByUserId	Task<BaseResultModel>	Public	Adds a notification for a user by their ID.

3.5.1.11 NotificationService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.

No	Method	Data Type	Visibility	Description
1	NotifyBudgetExceededAsync	Task	Public	Notifies user when their budget limit is exceeded.
2	PushMessageFirebase	Task<bool>	Public	Sends a push notification to the user via Firebase.

3.5.1.12 FirebaseLibrary class

No	Method	Data Type	Visibility	Description
1	SendMessageFireBase	Task<string>	Public	Sends a Firebase message with the specified title, body, and token.

3.5.1.13 IUnitOfWork class

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.5.1.14 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	TransactionsRepository	ITransactionRepository	Public	Repository for managing transaction data.
2	UsersRepository	IUsersRepository	Public	Repository for managing user data.

3	GroupFundRepository	IGroupFundRepository	Public	Repository for managing group fund data.
4	BankAccountRepository	IBankAccountRepository	Public	Repository for managing bank account relationships.
5	ImageRepository	IImageRepository	Public	Repository for managing image data.
6	FinancialGoalRepository	IFinancialGoalRepository	Public	Repository for managing financial goal data.
7	NotificationRepository	INotificationRepository	Public	Repository for managing notification data.

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.5.1.15 IImageRepository class

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.

3.5.1.16 ITransactionRepository class

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.

3.5.1.17 IUsersRepository class

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.

2	GetByIdAsync	Task<TEntity?>	Public	Asynchronously retrieves an entity by its ID, returning null if not found.

3.5.1.18 INotificationRepository class

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity to the repository and returns the added entity.

3.5.1.19 IFinancialGoalRepository class

No	Method	Data Type	Visibility	Description
1	GetActiveGoalByUserAndSubcategory	Task<FinancialGoal?>	Public	Asynchronously retrieves the active financial goal for a user and subcategory.
2	GetActiveGoalByGroupId	Task<FinancialGoal?>	Public	Retrieves active financial goal associated with a specific group.

3.5.1.20 IBankAccountRepository class

No	Method	Data Type	Visibility	Description
1	GetByIdAsync	Task<TEntity?>	Public	Retrieves entity from the database using the specified ID.

3.5.1.21 IGroupFundRepository class

No	Method	Data Type	Visibility	Description
1	GetByIdIncludeAsync	Task<TEntity?>	Public	Retrieves entity by ID including related data with optional filter.
2	GetByIdAsync	Task<TEntity?>	Public	Retrieves entity from the database using the specified ID.

3.5.1.22 ImageRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.

3.5.1.23 TransactionRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Asynchronously adds a range of entities to the repository.

3.5.1.24 UsersRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.
2	GetByIdAsync	Task<TEntity?>	Public	Asynchronously retrieves an entity by its ID, returning null if not found.

3.5.1.25 NotificationRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity to the repository and returns the added entity.

3.5.1.26 FinancialGoalRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetActiveGoalByUserAndSubcategory	Task<FinancialGoal?>	Public	Asynchronously retrieves the active financial goal for a user and subcategory.
2	GetActiveGoalByGroupId	Task<FinancialGoal?>	Public	Retrieves active financial goal associated with a specific group.

3.5.1.27 BankAccountRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByIdAsync	Task<TEntity?>	Public	Retrieves entity from the database using the specified ID.

3.5.1.28 GroupFundRepository class

No	Attribute	Data Type	Visibility	Description

1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByIdIncludeAsync	Task<TEntity?>	Public	Retrieves entity by ID including related data with optional filter.
2	GetByIdAsync	Task<TEntity?>	Public	Retrieves entity from the database using the specified ID.

3.5.1.29 Image class

No	Attribute	Data Type	Visibility	Description
1	EntityId	Guid	Public	The unique identifier for the entity.
2	EntityName	string	Public	The name of the entity.
3	ImageUrl	string	Public	The URL of the image associated with the entity.

3.5.1.30 Transaction class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid?	Public	ID of the group fund.
2	UserId	Guid?	Public	ID of the user who made the transaction.
3	Amount	decimal	Public	The amount of the transaction.
4	Type	TransactionType	Public	Type of transaction.
5	SubcategoryId	Guid?	Public	ID of the subcategory linked to this transaction.
6	TransactionDate	DateTime?	Public	The date the transaction occurred.
7	Description	string?	Public	Description of the transaction.
8	ApprovalRequired	bool?	Public	Whether approval is

				required.
9	RequestCode	string?	Public	Unique request code for approval tracking.
10	Status	TransactionStatus	Public	Current status of the transaction.
11	UserSpendingModelId	Guid?	Public	Related user's spending model ID.
12	InsertType	InsertType	Public	Indicates how the transaction was added.
13	Group	GroupFund?	Public	Navigation property to the related group.
14	Subcategory	Subcategory?	Public	Navigation to the subcategory.
15	TransactionVotes	ICollection<TransactionVote>	Public	Related transaction votes for approval.
16	User	User?	Public	Navigation property to the user

3.5.1.31 User class

No	Attribute	Data Type	Visibility	Description
1	FullName	string?	Public	Full name of the user.
2	NameUnsign	string?	Public	Full name without diacritics
3	Email	string?	Public	Email address of the user.
4	Password	string?	Public	User's password (if not using Google login).
5	Dob	DateTime?	Public	Date of birth.
6	Gender	Gender?	Public	Gender of the user.
7	Address	string?	Public	User’s address.
8	PhoneNumber	string?	Public	User’s phone number.
9	AvatarUrl	string?	Public	URL to the user's avatar image.
10	GoogleId	string?	Public	Google account ID (if signed in with Google).
11	IsVerified	bool?	Public	Whether the user's

				email is verified.
12	Role	RolesEnum?	Public	Role of the user
13	DeviceToken	string?	Public	Device token for push notifications.
14	Status	CommonsStatus?	Public	Current status of the user.
15	Assets	ICollection<Asset>	Public	List of user's assets.
16	Liabilities	ICollection<Liability>	Public	List of user's liabilities.
17	ChatHistories	ICollection<ChatHistory>	Public	Chat history records.
18	FinancialGoals	ICollection<FinancialGoal>	Public	User’s financial goals.
19	FinancialReports	ICollection<FinancialReport>	Public	Generated financial reports.
20	GroupMembers	ICollection<GroupMember>	Public	User’s membership in groups.
21	Notifications	ICollection<Notification>	Public	Notifications received by the user.
22	RecurringTransactions	ICollection<RecurringTransaction>	Public	Scheduled recurring transactions.
23	Subscriptions	ICollection<Subscription>	Public	User’s active subscriptions.
24	Transactions	ICollection<Transaction>	Public	All transactions linked to the user.
25	UserQuizResults	ICollection<UserQuizResult>	Public	Results from financial quizzes.
26	UserSettings	ICollection<UserSetting>	Public	Custom user settings/preferences.
27	UserSpendingModels	ICollection<UserSpendingModel>	Public	Linked spending models.
28	BankAccounts	ICollection<BankAccount>	Public	Bank accounts linked to the user.

3.5.1.32 Notification class

No	Attribute	Data Type	Visibility	Description
1	IsRead	bool	Public	Indicates whether the notification has been read.

2	Title	string?	Public	Title of the notification.
3	TitleUnsign	string?	Public	Title without diacritics (for search and filtering).
4	Message	string?	Public	Content/message of the notification.
5	Href	string?	Public	Link to redirect when the notification is clicked.
6	EntityId	Guid?	Public	Related entity ID
7	UserId	Guid?	Public	ID of the user who receives this notification.
8	Type	NotificationType?	Public	Type/category of the notification
9	User	User?	Public	Navigation property to the related User entity.

3.5.1.33 FinancialGoal class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid?	Public	ID of the user who owns the financial goal.
2	GroupId	Guid?	Public	Optional ID of the group associated with the goal.
3	SubcategoryId	Guid?	Public	Optional subcategory linked to the goal.
4	Name	string	Public	Name/title of the financial goal.
5	NameUnsign	string	Public	Name without diacritics.
6	TargetAmount	decimal	Public	The total amount to be saved or achieved.
7	CurrentAmount	decimal	Public	Current saved amount toward the goal.
8	StartDate	DateTime	Public	Date the goal was started.
9	Deadline	DateTime	Public	Deadline to reach the financial goal.

10	Status	FinancialGoalStatus	Public	Current status of the goal.
11	ApprovalStatus	ApprovalStatus	Public	Approval status for goals that require validation.
12	IsSaving	bool	Public	Whether the goal is currently in saving mode.
13	Group	GroupFund?	Public	Navigation property to related group, if any.
14	User	User	Public	Navigation property to the associated user.
15	Subcategory	Subcategory?	Public	Navigation property to linked subcategory.

3.5.1.34 BankAccount class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid	Public	Represents the identifier of the user who owns the bank account.
2	AccountNumber	string	Public	The account number assigned by the user's bank.
3	BankName	string	Public	The full name of the bank holding the account.
4	BankShortName	string	Public	The abbreviated or short name of the bank.
5	AccountHolderName	string	Public	The name of the person who owns the bank account.
6	Status	CommonsStatus?	Public	Indicates the current status of the bank account.
7	WebhookSecretKey	string?	Public	Secret key used to authenticate webhook calls.
8	WebhookUrl	string?	Public	The URL where webhook events should be sent.

9	User	User?	Public	Navigation property referencing the associated user.

3.5.1.35 GroupFund class

No	Attribute	Data Type	Visibility	Description
1	Name	string	Public	The official name of the group fund.
2	NameUnsign	string	Public	Unaccented version of the group fund name.
3	Description	string	Public	Textual information describing the group fund.
4	CurrentBalance	decimal	Public	Current monetary balance available in the fund.
5	Status	GroupStatus?	Public	The status representing the group's activity state.
6	Visibility	VisibilityEnum ?	Public	Defines the access level visibility of the group.
7	AccountBankId	Guid?	Public	Identifier of the linked bank account if any.
8	FinancialGoals	ICollection<FinancialGoal>	Public	List of financial goals related to the group.
9	FinancialReports	ICollection<FinancialReport>	Public	Collection of financial reports for the group.
10	GroupFundLogs	ICollection<GroupFundLog>	Public	Log entries for actions within the group fund.
11	GroupMembers	ICollection<GroupMember>	Public	Members who belong to the group fund.
12	Transactions	ICollection<Transaction>	Public	All transactions performed under the group.

3.5.1.36 EntityName enum

No	Enum	Visibility	Description
1	GROUP	Public	Refers to a group entity
2	TRANSACTION	Public	Refers to a transaction entity

3.5.1.37 NotificationType enum

No	Enum	Visibility	Description
1	SYSTEM	Public	System-generated notification for all users
2	USER	Public	Personal notification targeted to an individual user
3	GROUP	Public	Notification related to group activity
4	GROUP_INVITE	Public	Invitation to join a group
5	FINANCIAL_GOAL_PERSONAL	Public	Notification regarding a personal financial goal
6	FINANCIAL_GOAL_GROUP	Public	Notification related to a group financial goal

3.5.1.38 TransactionType enum

No	Enum	Visibility	Description
1	INCOME	Public	Refers to an income transaction
2	EXPENSE	Public	Refers to an expense transaction

3.5.1.39 RoleGroup enum

No	Enum	Visibility	Description
1	LEADER	Public	The user who created and leads the group.
2	MOD	Public	A moderator who can manage group activities.
3	MEMBER	Public	A regular member with standard group access.

3.5.1.40 TransactionStatus enum

No	Enum	Visibility	Description
1	PENDING	Public	The transaction is waiting for approval or review.
2	APPROVED	Public	The transaction has been reviewed and approved.
3	REJECTED	Public	The transaction has been reviewed and rejected.

3.5.1.41 ITransactionService class

No	Method	Data Type	Visibility	Description
1	CreateGroupTransactionAsync	Task<BaseResultModel>	Public	Creates a new transaction in a group context using user email.

3.5.1.42 TransactionService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.
3	_transactionNotificationService	ITransactionNotificationService	Private	Sends transaction-related notifications.

No	Method	Data Type	Visibility	Description
1	CreateGroupTransactionAsync	Task<BaseResultModel>	Public	Creates a new transaction in a group context using user email.

3.5.1.43 BaseResultModel class

No	Attribute	Data Type	Visibility	Description
1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies the specific error if any occurred.
3	Message	string	Public	A message that describes the result or error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.5.1.44 GroupMemberStatus enum

No	Enum	Visibility	Description

1	ACTIVE	Public	Public The group member is currently active and participating in the group.
2	PENDING	Public	Public The group member has been invited or requested to join but has not yet been approved
3	INACTIVE	Public	The group member is no longer active or has been removed from the group.

3.5.1.45 CreateGroupTransactionModel class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid	Public	Unique identifier for the group associated with the transaction.
2	Amount	decimal	Public	Amount of money involved in the transaction.
3	Type	TransactionType	Public	Type of transaction (Income/Expense).
4	TransactionDate	DateTime	Public	Date when the transaction occurred.
5	Description	string?	Public	A description of the transaction.
6	Images	List<string>?	Public	List of image URLs related to the transaction.
7	RequireVote	bool	Public	Whether the transaction requires a vote for approval.
8	InsertType	InsertType	Public	Type of insertion for the transaction.
9	AccountBankNumber	string?	Public	Bank account number related to the transaction.
10	AccountBankName	string?	Public	Bank name where the transaction was made.
11	BankTransactionId	string?	Public	Bank-specific transaction identifier.
12	BankTransactionDate	DateTime?	Public	Date when the bank transaction was completed.

3.5.2 Sequence Diagram

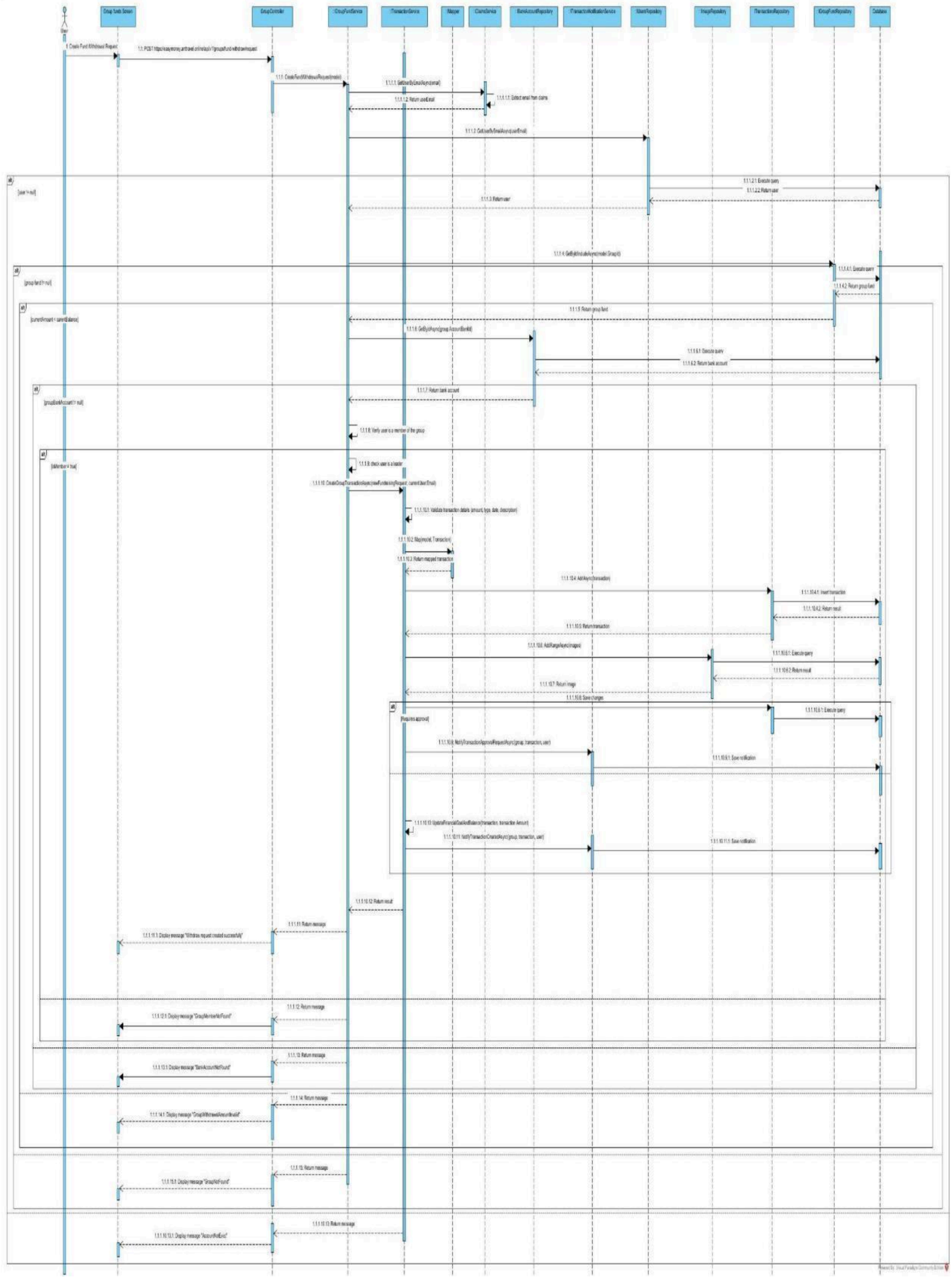


Figure 23 - Sequence Diagram

3.6 Create recurring transaction

3.6.1 Class Diagram

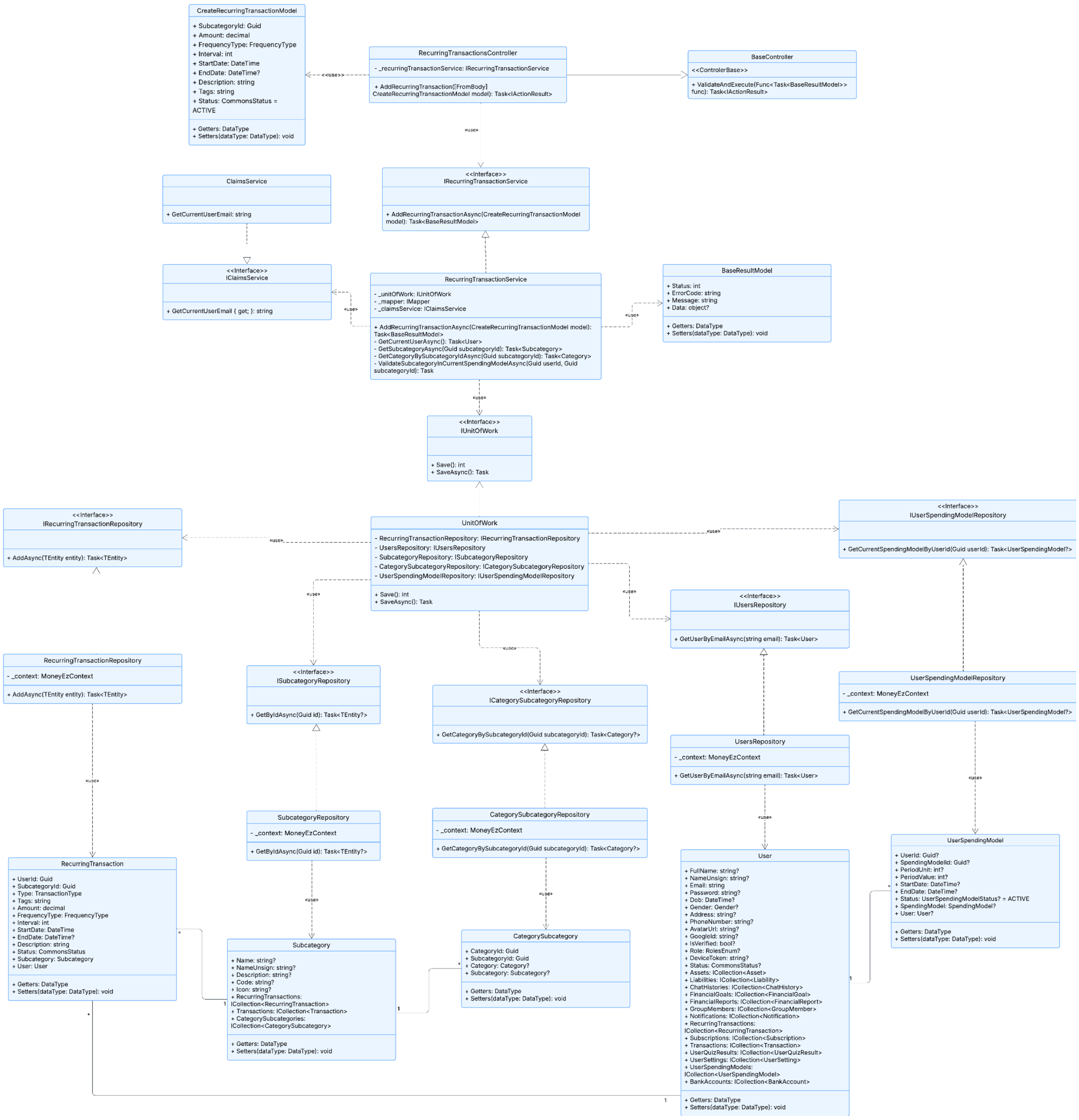


Figure 24 - Class Diagram

3.6.1.1 RecurringTransactionsController class

No	Attribute	Data Type	Visibility	Description
1	_recurringTransactionService	IRecurringTransactionService	Private	Service responsible for processing recurring transaction logic.

No	Method	Data Type	Visibility	Description

1	AddRecurringTransaction	Task<IActionResult>	Public	Handles the addition of a recurring transaction.

3.6.1.2 BaseController class

No	Method	Data Type	Visibility	Description
1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.6.1.3 CreateRecurringTransactionModel class

No	Attribute	Data Type	Visibility	Description
1	SubcategoryId	Guid	Public	The ID of the subcategory for the transaction.
2	Amount	decimal	Public	The amount of the recurring transaction.
3	FrequencyType	FrequencyType	Public	The frequency type.
4	Interval	int	Public	The interval at which the transaction occurs.
5	StartDate	DateTime	Public	The start date for the recurring transaction.
6	EndDate	DateTime?	Public	The end date for the recurring transaction.
7	Description	string	Public	The description of the transaction.
8	Tags	string	Public	Tags associated with the transaction for categorization.
9	Status	CommonsStatus	Public	The status of the transaction.

3.6.1.4 IRecurringTransactionService class

No	Method	Data Type	Visibility	Description
1	AddRecurringTransactionAsync	Task<BaseResultModel>	Public	The model containing data for the recurring transaction to be added.

3.6.1.5 RecurringTransactionService class

No	Attribute	Data Type	Visibility	Description

1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.
3	_claimsService	IClaimsService	Private	Service for handling user claims extracted from the token.

No	Method	Data Type	Visibility	Description
1	AddRecurringTransactionAsync	Task<BaseResultModel>	Public	The model containing data for the recurring transaction to be added.

3.6.1.6 ClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Email extracted from current user claims.

3.6.1.7 IClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Gets the email of the current user.

3.6.1.8 BaseResultModel class

No	Attribute	Data Type	Visibility	Description
1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies the specific error if any occurred.
3	Message	string	Public	A message that describes the result or error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.6.1.9 IUnitOfWork class

No	Method	Data Type	Visibility	Description

1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.6.1.10 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	RecurringTransactionRepository	IRecurringTransactionRepository	Public	Repository for managing recurring transaction data.
2	UsersRepository	IUsersRepository	Public	Repository for managing user data.
3	SubcategoryRepository	ISubcategoryRepository	Public	Repository for managing subcategory data.
4	CategorySubcategoryRepository	ICategorySubcategoryRepository	Public	Repository for managing category subcategory relationships.
5	UserSpendingModelRepository	IUserSpendingModelRepository	Public	Repository for managing user spending model data.

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.6.1.11 IUsersRepository class

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.

3.6.1.12 ICategorySubcategoryRepository class

No	Method	Data Type	Visibility	Description

1	GetCategoryBySubcategoryId	Task<Category?>	Public	Retrieves the category associated with the specified subcategory identifier.

3.6.1.13 ISubcategoryRepository class

No	Method	Data Type	Visibility	Description
1	GetByIdAsync	Task<TEntity?>	Public	Retrieves an entity by its unique identifier asynchronously.

3.6.1.14 IRecurringTransactionRepository class

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity?>	Public	Asynchronously adds a new entity to the data source and returns it.

3.6.1.15 IUserSpendingModelRepository class

No	Method	Data Type	Visibility	Description
1	GetCurrentSpendingModelByUserId	Task<UserSpendingModel?>	Public	Retrieves the current active spending model for the given user identifier.

3.6.1.16 UsersRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.

3.6.1.17 CategorySubcategoryRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the

				database.

No	Method	Data Type	Visibility	Description
1	GetCategoryBySubcategoryId	Task<Category?>	Public	Retrieves the category associated with the specified subcategory identifier.

3.6.1.18 SubcategoryRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByIdAsync	Task<TEntity?>	Public	Retrieves an entity by its unique identifier asynchronously.

3.6.1.19 RecurringTransactionRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity?>	Public	Asynchronously adds a new entity to the data source and returns it.

3.6.1.20 UserSpendingModelRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetCurrentSpendingModelByUserId	Task<UserSpendingModel?>	Public	Retrieves the current active spending model for the given user identifier.

3.6.1.21 User class

No	Attribute	Data Type	Visibility	Description
1	FullName	string?	Public	Full name of the user.
2	NameUnsign	string?	Public	Full name without diacritics
3	Email	string?	Public	Email address of the user.
4	Password	string?	Public	User's password (if not using Google login).
5	Dob	DateTime?	Public	Date of birth.
6	Gender	Gender?	Public	Gender of the user.
7	Address	string?	Public	User’s address.
8	PhoneNumber	string?	Public	User’s phone number.
9	AvatarUrl	string?	Public	URL to the user's avatar image.
10	GoogleId	string?	Public	Google account ID (if signed in with Google).
11	IsVerified	bool?	Public	Whether the user's email is verified.
12	Role	RolesEnum?	Public	Role of the user
13	DeviceToken	string?	Public	Device token for push notifications.
14	Status	CommonsStatus?	Public	Current status of the user.
15	Assets	ICollection<Asset>	Public	List of user's assets.

16	Liabilities	ICollection<Liability>	Public	List of user's liabilities.
17	ChatHistories	ICollection<ChatHistory>	Public	Chat history records.
18	FinancialGoals	ICollection<FinancialGoal>	Public	User’s financial goals.
19	FinancialReports	ICollection<FinancialReport>	Public	Generated financial reports.
20	GroupMembers	ICollection<GroupMember>	Public	User’s membership in groups.
21	Notifications	ICollection<Notification>	Public	Notifications received by the user.
22	RecurringTransactions	ICollection<RecurringTransaction>	Public	Scheduled recurring transactions.
23	Subscriptions	ICollection<Subscription>	Public	User’s active subscriptions.
24	Transactions	ICollection<Transaction>	Public	All transactions linked to the user.
25	UserQuizResults	ICollection<UserQuizResult>	Public	Results from financial quizzes.
26	UserSettings	ICollection<UserSetting>	Public	Custom user settings/preferences.
27	UserSpendingModels	ICollection<UserSpendingModel>	Public	Linked spending models.
28	BankAccounts	ICollection<BankAccount>	Public	Bank accounts linked to the user.

3.6.1.22 UserSpendingModel class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid?	Public	The ID of the user.
2	SpendingModelId	Guid?	Public	The ID of the spending model.
3	PeriodUnit	int?	Public	The unit of the time period.
4	PeriodValue	int?	Public	The value of the time period.
5	StartDate	DateTime?	Public	The start date.
6	EndDate	DateTime?	Public	The end date.

7	Status	UserSpendingModelStatus?	Public	The status of the spending model.
8	SpendingModel	SpendingModel?	Public	The spending model linked to the user.
9	User	User?	Public	The user linked to this model.

3.6.1.23 CategorySubcategory class

No	Attribute	Data Type	Visibility	Description
1	CategoryId	Guid	Public	The ID of the category.
2	SubcategoryId	Guid	Public	The ID of the subcategory.
3	Category	Category?	Public	The category object associated with this model.
4	Subcategory	Subcategory?	Public	The subcategory object associated with this model.

3.6.1.24 Subcategory class

No	Attribute	Data Type	Visibility	Description
1	Name	string?	Public	The name of the model
2	NameUnsign	string?	Public	The unsign version of the name
3	Description	string?	Public	A description of the model.
4	Code	string?	Public	The code associated with the model.
5	Icon	string?	Public	The icon associated with the model.
6	RecurringTransactions	ICollection<RecurringTransaction>	Public	A collection of recurring transactions linked to the model.
7	Transactions	ICollection<Transaction>	Public	A collection of transactions linked to the model.
8	CategorySubcategories	ICollection<CategorySubcategory>	Public	A collection of category subcategories linked to the model.

3.6.1.25 RecurringTransaction class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid	Public	The ID of the user.
2	SubcategoryId	Guid	Public	The ID of the subcategory.
3	Type	TransactionTyp e	Public	The type of the transaction.
4	Tags	string	Public	The tags associated with the transaction.
5	Amount	decimal	Public	The amount of the transaction.
6	FrequencyType	FrequencyType	Public	The type of frequency.
7	Interval	int	Public	The interval for the transaction.
8	StartDate	DateTime	Public	The start date of the transaction.
9	EndDate	DateTime?	Public	The end date of the transaction.
10	Description	string	Public	A description of the transaction.
11	Status	CommonsStatu s	Public	The status of the transaction.
12	Subcategory	Subcategory	Public	The subcategory linked to the transaction.
13	User	User	Public	The user linked to the transaction.

3.6.2 Sequence Diagram

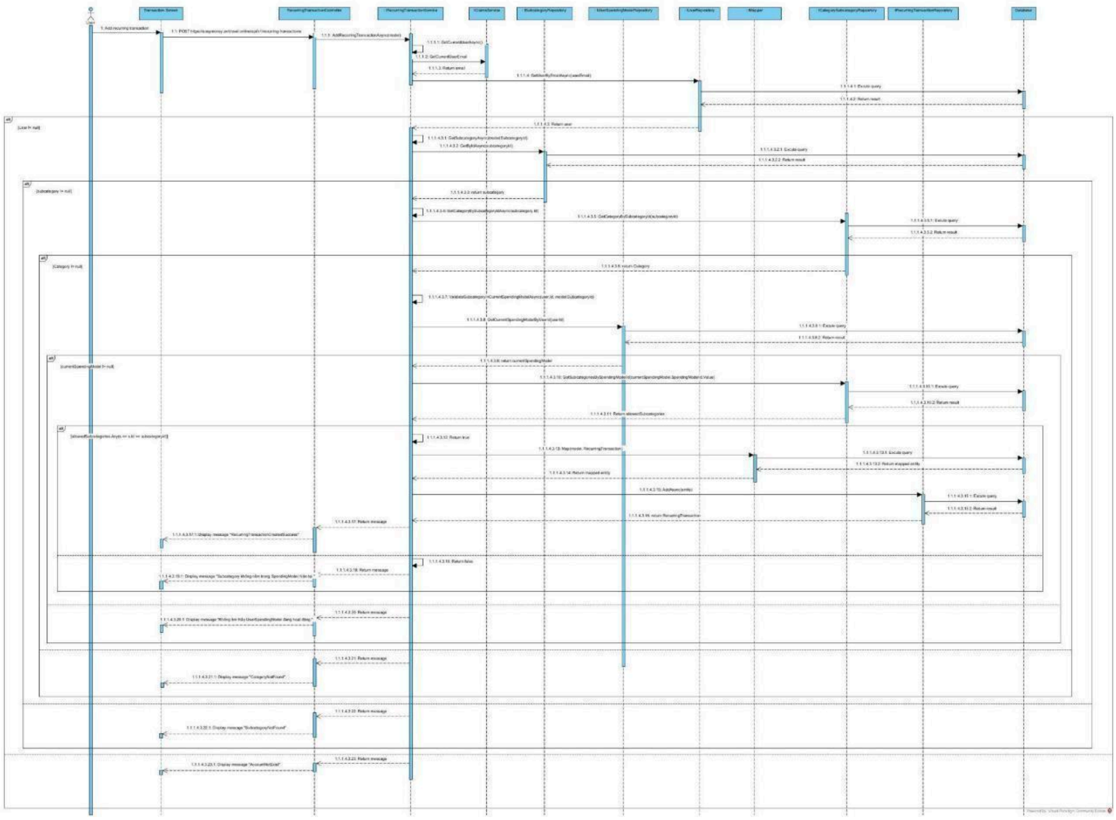


Figure 25 - Sequence Diagram

3.7 Create spending model

3.7.1 Class Diagram

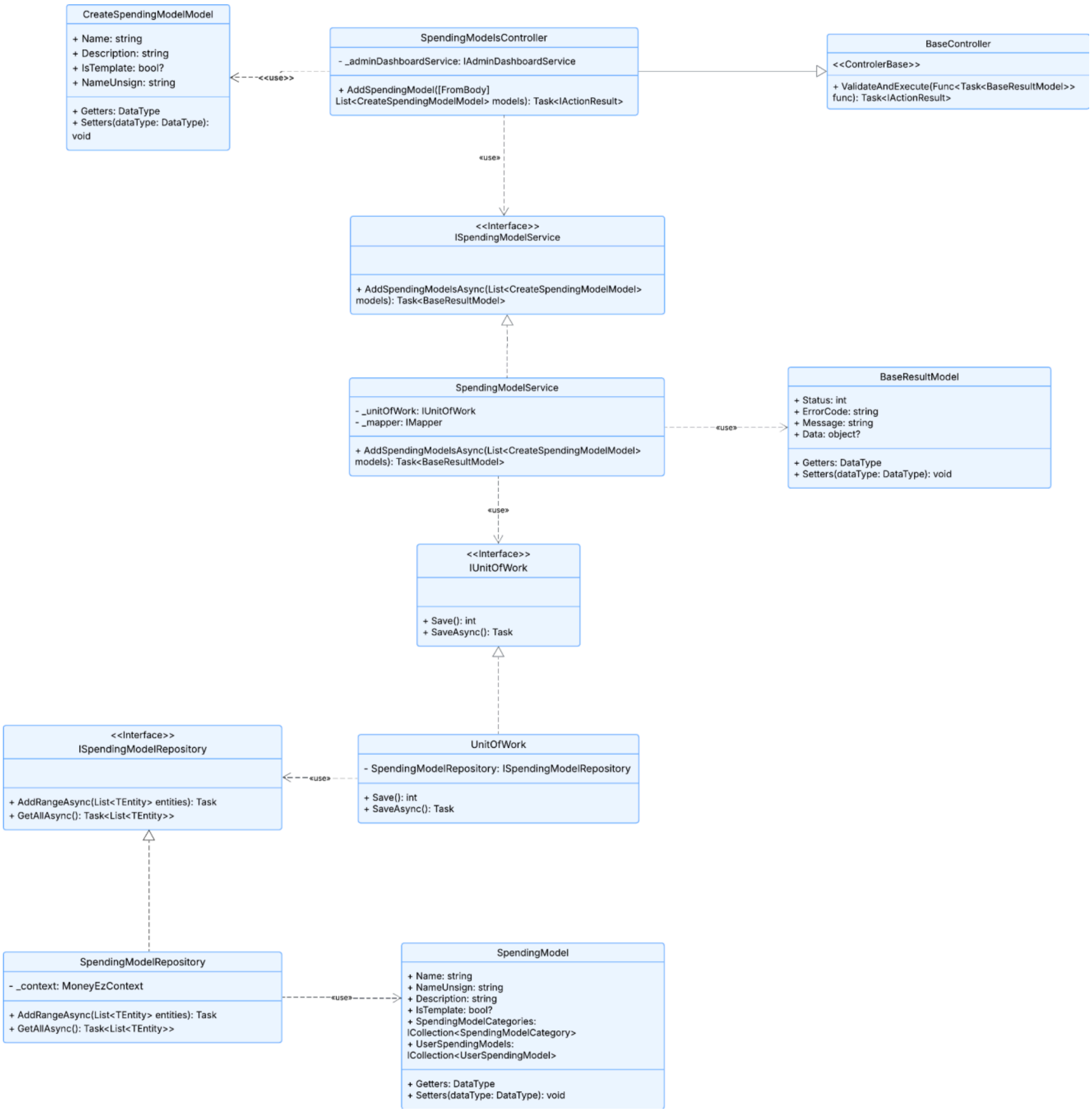


Figure 26 - Class Diagram

3.7.1.1 SpendingModelsController class

No	Attribute	Data Type	Visibility	Description
1	_adminDashboardService	IAdminDashboardService	Private	Service responsible for processing spending model logic.

No	Method	Data Type	Visibility	Description

1	AddSpendingModel	Task<IActionResult>	Public	Adds a list of spending models.

3.7.1.2 BaseController class

No	Method	Data Type	Visibility	Description
1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.7.1.3 CreateRecurringTransactionModel class

No	Attribute	Data Type	Visibility	Description
1	Name	string	Public	The name of the model.
2	Description	string	Public	A description of the model.
3	IsTemplate	bool?	Public	Whether it is a template.
4	NameUnsign	string	Public	The unsign version of the name.

3.7.1.4 ISpendingModelService class

No	Method	Data Type	Visibility	Description
1	AddSpendingModelsAsync	Task<BaseResultModel>	Public	Adds a list of spending models

3.7.1.5 SpendingModelService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.

No	Method	Data Type	Visibility	Description
1	AddSpendingModelsAsync	Task<BaseResultModel>	Public	Adds a list of spending models

3.7.1.6 BaseResultModel class

No	Attribute	Data Type	Visibility	Description
1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies the specific error if any occurred.
3	Message	string	Public	A message that describes the result or error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.7.1.7 IUnitOfWork class

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.7.1.8 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	SpendingModelRepository	ISpendingModelRepository	Public	Repository for managing spending model data.

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.7.1.9 ISpendingModelRepository class

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Adds a range of entities.

2	GetAllAsync	Task<List<TEntity>>	Public	Retrieves all entities.

3.7.1.10 SpendingModelRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Adds a range of entities.
2	GetAllAsync	Task<List<TEntity>>	Public	Retrieves all entities.

3.7.1.11 SpendingModel class

No	Attribute	Data Type	Visibility	Description
1	Name	string	Public	The name of the model.
2	NameUnsign	string	Public	The unsign version of the name.
3	Description	string	Public	A description of the model.
4	IsTemplate	bool?	Public	Whether it is a template.
5	SpendingModelCategories	ICollection<SpendingModelCategory>	Public	Collection of spending model categories.
6	UserSpendingModels	ICollection<UserSpendingModel>	Public	Collection of user spending models.

3.7.2 Sequence Diagram

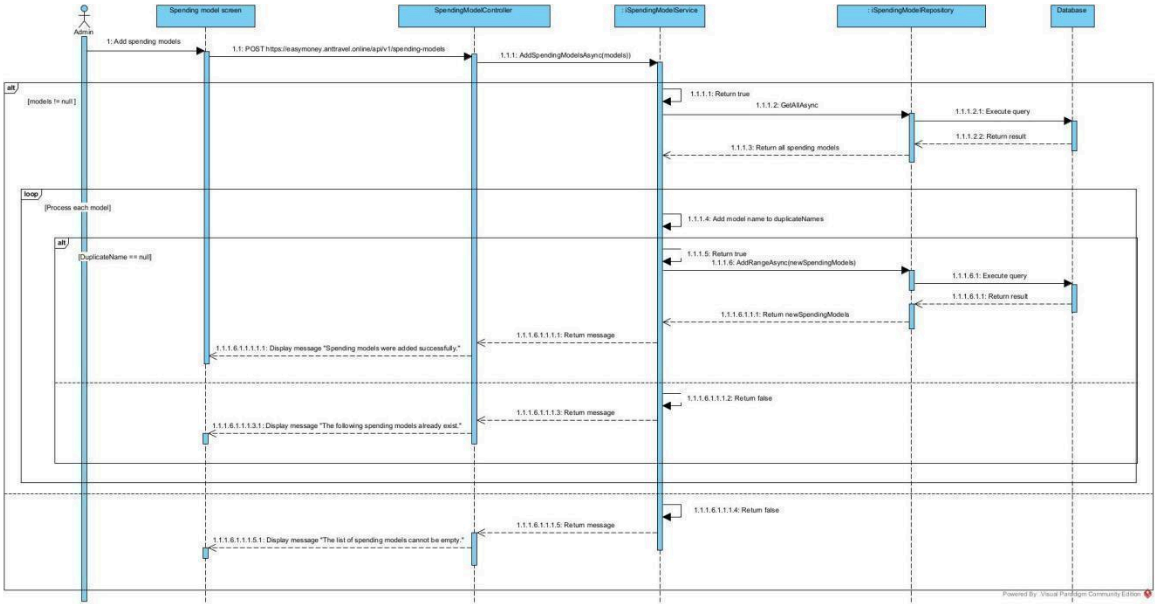


Figure 27 - Sequence Diagram

3.8 Invite member by QR

3.8.1 Class Diagram

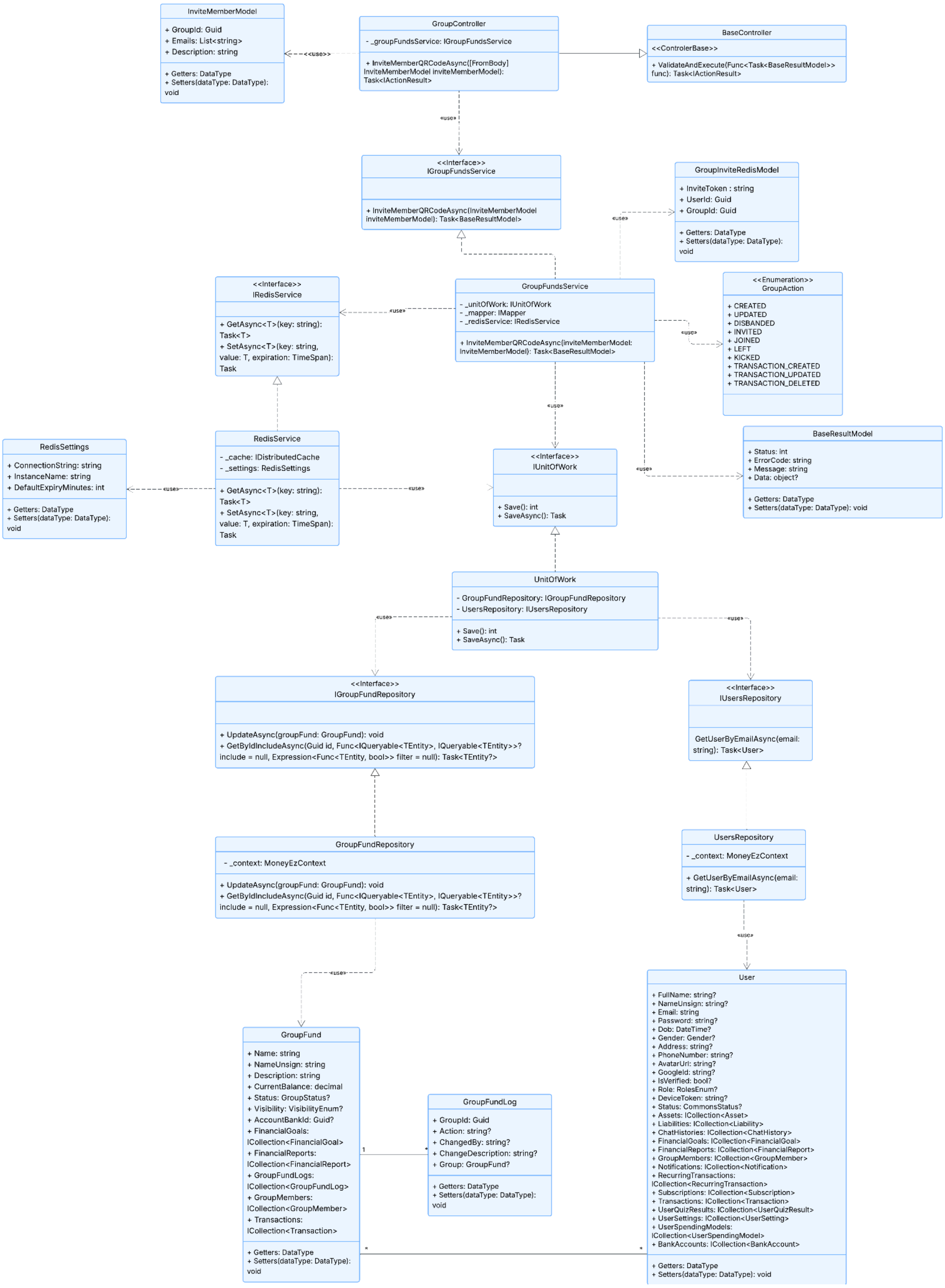


Figure 28 - Class Diagram

3.8.1.1 GroupController class

No	Attribute	Data Type	Visibility	Description
1	_groupFundsService	IGroupFundsService	Private	Service responsible for processing group fund logic.

No	Method	Data Type	Visibility	Description
1	InviteMemberQRCodeAsync	Task<IActionResult>	Public	Invites a member via QR code.

3.8.1.2 BaseController class

No	Method	Data Type	Visibility	Description
1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.8.1.3 InviteMemberModel class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid	Public	The ID of the group.
2	Emails	List<string>	Public	List of member email addresses.
3	Description	string	Public	A description of the request.

3.8.1.4 IGroupFundsService class

No	Method	Data Type	Visibility	Description
1	InviteMemberQRCodeAsync	Task<BaseResultModel>	Public	Invites a member via QR code.

3.8.1.5 GroupFundsService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.
3	_redisService	IRedisService	Private	Manages Redis cache service.

No	Method	Data Type	Visibility	Description
1	InviteMemberQRCodeAsync	Task<BaseResultModel>	Public	Invites a member via QR code.

3.8.1.6 BaseResultModel class

No	Attribute	Data Type	Visibility	Description
1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies the specific error if any occurred.
3	Message	string	Public	A message that describes the result or error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.8.1.7 IUnitOfWork class

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.8.1.8 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	GroupFundRepository	IGroupFundRepository	Public	Repository for managing sgroup fund data.
2	UsersRepository	IUsersRepository	Public	Repository for managing user data.

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.8.1.9 IGroupFundRepository class

No	Method	Data Type	Visibility	Description
1	UpdateAsync	void	Public	Updates a group fund.
2	GetByIdIncludeAsync	Task<TEntity?>	Public	Gets entity by ID with includes.

3.8.1.10 IUsersRepository class

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Gets user by email address.

3.8.1.11 GroupFundRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	UpdateAsync	void	Public	Updates a group fund.
2	GetByIdIncludeAsync	Task<TEntity?>	Public	Gets entity by ID with includes.

3.8.1.12 UsersRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Gets user by email address.

3.8.1.13 IRedisService class

No	Method	Data Type	Visibility	Description

1	GetAsync	Task<T>	Public	Retrieves a value from cache.
2	SetAsync	Task	Public	Stores a value in cache with expiry.

3.8.1.14 RedisService class

No	Attribute	Data Type	Visibility	Description
1	_cache	IDistributedCache	Public	Interface for distributed caching.
2	_settings	RedisSettings	Public	Configuration settings for Redis.

No	Method	Data Type	Visibility	Description
1	GetAsync	Task<T>	Public	Retrieves a value from cache.
2	SetAsync	Task	Public	Stores a value in cache with expiry.

3.8.1.15 RedisSettings class

No	Attribute	Data Type	Visibility	Description
1	ConnectionString	string	Public	Redis connection string.
2	InstanceName	string	Public	Redis instance name.
3	DefaultExpiryMinutes	int	Public	Default expiration time in minutes.

3.8.1.16 User class

No	Attribute	Data Type	Visibility	Description
1	FullName	string?	Public	Full name of the user.
2	NameUnsign	string?	Public	Full name without diacritics
3	Email	string?	Public	Email address of the user.
4	Password	string?	Public	User's password (if not using Google login).
5	Dob	DateTime?	Public	Date of birth.
6	Gender	Gender?	Public	Gender of the user.

7	Address	string?	Public	User’s address.
8	PhoneNumber	string?	Public	User’s phone number.
9	AvatarUrl	string?	Public	URL to the user's avatar image.
10	GoogleId	string?	Public	Google account ID (if signed in with Google).
11	IsVerified	bool?	Public	Whether the user's email is verified.
12	Role	RolesEnum?	Public	Role of the user
13	DeviceToken	string?	Public	Device token for push notifications.
14	Status	CommonsStatus?	Public	Current status of the user.
15	Assets	ICollection<Asset>	Public	List of user's assets.
16	Liabilities	ICollection<Liability>	Public	List of user's liabilities.
17	ChatHistories	ICollection<ChatHistory>	Public	Chat history records.
18	FinancialGoals	ICollection<FinancialGoal>	Public	User’s financial goals.
19	FinancialReports	ICollection<FinancialReport>	Public	Generated financial reports.
20	GroupMembers	ICollection<GroupMember>	Public	User’s membership in groups.
21	Notifications	ICollection<Notification>	Public	Notifications received by the user.
22	RecurringTransactions	ICollection<RecurringTransaction>	Public	Scheduled recurring transactions.
23	Subscriptions	ICollection<Subscription>	Public	User’s active subscriptions.
24	Transactions	ICollection<Transaction>	Public	All transactions linked to the user.
25	UserQuizResults	ICollection<UserQuizResult>	Public	Results from financial quizzes.
26	UserSettings	ICollection<UserSetting>	Public	Custom user settings/preferences.

27	UserSpendingModels	ICollection<UserSpendingModel>	Public	Linked spending models.
28	BankAccounts	ICollection<BankAccount>	Public	Bank accounts linked to the user.

3.8.1.17 GroupFund class

No	Attribute	Data Type	Visibility	Description
1	Name	string	Public	The official name of the group fund.
2	NameUnsign	string	Public	Unaccented version of the group fund name.
3	Description	string	Public	Textual information describing the group fund.
4	CurrentBalance	decimal	Public	Current monetary balance available in the fund.
5	Status	GroupStatus?	Public	The status representing the group's activity state.
6	Visibility	VisibilityEnum?	Public	Defines the access level visibility of the group.
7	AccountBankId	Guid?	Public	Identifier of the linked bank account if any.
8	FinancialGoals	ICollection<FinancialGoal>	Public	List of financial goals related to the group.
9	FinancialReports	ICollection<FinancialReport>	Public	Collection of financial reports for the group.
10	GroupFundLogs	ICollection<GroupFundLog>	Public	Log entries for actions within the group fund.
11	GroupMembers	ICollection<GroupMember>	Public	Members who belong to the group fund.
12	Transactions	ICollection<Transaction>	Public	All transactions performed under the group.

3.8.1.18 GroupFundLog class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid	Public	ID of the related group fund.

2	Action	string?	Public	Performed action name.
3	ChangedBy	string?	Public	User who made the change.
4	ChangeDescription	string?	Public	Description of the change.
5	Group	GroupFund?	Public	Related group fund entity.

3.8.1.19 GroupAction enum

No	Enum	Visibility	Description
1	CREATED	Public	Group has been created.
2	UPDATED	Public	Group information was updated.
3	DISBANDED	Public	Group has been disbanded.
4	INVITED	Public	Member has been invited.
5	JOINED	Public	Member has joined the group.
6	LEFT	Public	Member has left the group.
7	KICKED	Public	Member was removed from the group.
8	TRANSACTION_CREATED	Public	A transaction was created in group.
9	TRANSACTION_UPDATED	Public	A transaction was updated in group.
10	TRANSACTION_DELETED	Public	A transaction was deleted from group.

3.8.1.20 GroupInviteRedisModel class

No	Attribute	Data Type	Visibility	Description
1	InviteToken	string	Public	Token used for joining a group.
2	UserId	Guid	Public	ID of the invited user.
3	GroupId	Guid	Public	ID of the target group.

3.8.2 Sequence Diagram

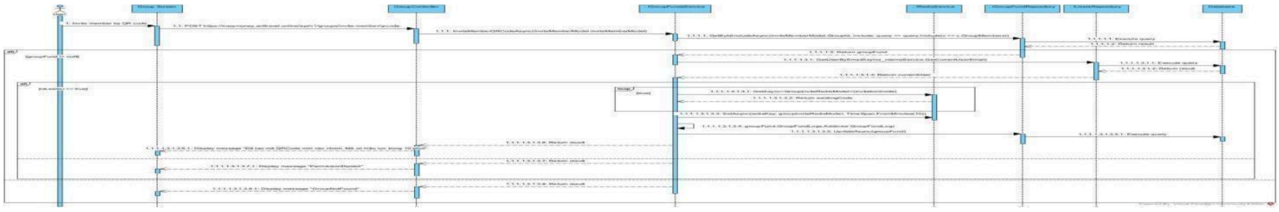


Figure 29 - Sequence Diagram

3.9 Invite member by email

3.9.1 Class Diagram

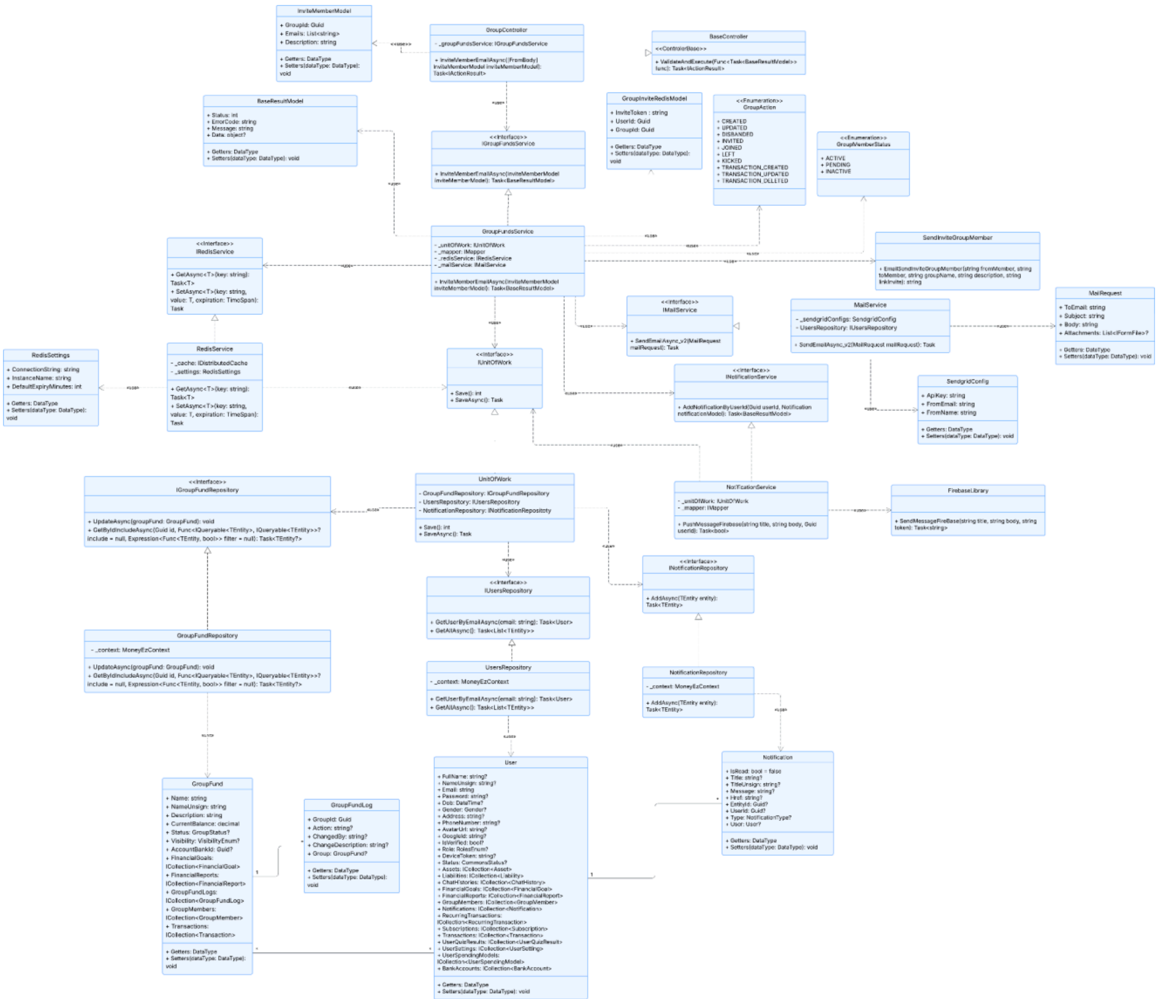


Figure 30 - Class Diagram

3.9.1.1 GroupController class

No	Attribute	Data Type	Visibility	Description
1	_groupFundsService	IGroupFundsService	Private	Service responsible for processing group fund logic.

No	Method	Data Type	Visibility	Description
1	InviteMemberEmailAsync	Task<IActionResult>	Public	Invites a member via email.

3.9.1.2 BaseController class

No	Method	Data Type	Visibility	Description
1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.9.1.3 InviteMemberModel class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid	Public	The ID of the group.
2	Emails	List<string>	Public	List of member email addresses.
3	Description	string	Public	A description of the request.

3.9.1.4 IGroupFundsService class

No	Method	Data Type	Visibility	Description
1	InviteMemberEmailAsync	Task<BaseResultModel>	Public	Invites a member via email.

3.9.1.5 GroupFundsService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.
3	_redisService	IRedisService	Private	Manages Redis cache service.
4	_mailService	IMailService	Private	Sends emails to users

No	Method	Data Type	Visibility	Description
1	InviteMemberEmailAsync	Task<BaseResultModel>	Public	Invites a member via email.

3.9.1.6 BaseResultModel class

No	Attribute	Data Type	Visibility	Description
1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies the specific error if any occurred.
3	Message	string	Public	A message that describes the result or error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.9.1.7 IUnitOfWork class

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.9.1.8 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	GroupFundRepository	IGroupFundRepository	Public	Repository for managing sgroup fund data.
2	UsersRepository	IUsersRepository	Public	Repository for managing user data.
3	NotificationRepository	INotificationRepository	Public	Repository for managing notification data.

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.9.1.9 IRedisService class

No	Method	Data Type	Visibility	Description
1	GetAsync	Task<T>	Public	Retrieves a value from cache.
2	SetAsync	Task	Public	Stores a value in cache with expiry.

3.9.1.10 RedisService class

No	Attribute	Data Type	Visibility	Description
1	_cache	IDistributedCache	Public	Interface for distributed caching.
2	_settings	RedisSettings	Public	Configuration settings for Redis.

No	Method	Data Type	Visibility	Description
1	GetAsync	Task<T>	Public	Retrieves a value from cache.
2	SetAsync	Task	Public	Stores a value in cache with expiry.

3.9.1.11 RedisSettings class

No	Attribute	Data Type	Visibility	Description
1	ConnectionString	string	Public	Redis connection string.
2	InstanceName	string	Public	Redis instance name.
3	DefaultExpiryMinutes	int	Public	Default expiration time in minutes.

3.9.1.12 IGroupFundRepository class

No	Method	Data Type	Visibility	Description
1	UpdateAsync	void	Public	Updates a group fund.
2	GetByIdIncludeAsync	Task<TEntity?>	Public	Gets entity by ID with includes.

3.9.1.13 IUsersRepository class

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Gets user by email address.
2	GetAllAsync	Task<List<TEntity>>	Public	Retrieves all entities

3.9.1.14 INotificationRepository class

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Adds a new entity to store

3.9.1.15 IMailService class

No	Method	Data Type	Visibility	Description
1	SendEmailAsync_v2	Task	Public	Sends an email asynchronously.

3.9.1.16 INotificationService class

No	Method	Data Type	Visibility	Description
1	AddNotificationByUserId	Task<BaseResultModel>	Public	Adds a notification for a specific user.

3.9.1.17 GroupFundRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	UpdateAsync	void	Public	Updates a group fund.
2	GetByIdIncludeAsync	Task<TEntity?>	Public	Gets entity by ID with includes.

3.9.1.18 UsersRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description

1	GetUserByEmailAsync	Task<User>	Public	Gets user by email address.

3.9.1.19 MailService class

No	Attribute	Data Type	Visibility	Description
1	_sendgridConfigs	SendgridConfig	Private	Configuration for SendGrid service.
2	UsersRepository	IUsersRepository	Private	Repository for accessing user data.

No	Method	Data Type	Visibility	Description
1	SendEmailAsync_v2	Task	Public	Sends an email asynchronously.

3.9.1.20 NotificationService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.

No	Method	Data Type	Visibility	Description
1	NotifyBudgetExceededAsync	Task	Public	Notifies user when their budget limit is exceeded.
2	PushMessageFirebase	Task<bool>	Public	Sends a push notification to the user via Firebase.

3.9.1.21 FirebaseLibrary class

No	Method	Data Type	Visibility	Description
1	SendMessageFireBase	Task<string>	Public	Sends a Firebase message with the specified title, body, and token.

3.9.1.22 SendInviteGroupMember class

No	Method	Data Type	Visibility	Description

1	EmailSendInviteGroupMember	string	Public	Generates an email content for group invitation.

3.9.1.23 GroupInviteRedisModel class

No	Attribute	Data Type	Visibility	Description
1	InviteToken	string	Public	Token used for joining a group.
2	UserId	Guid	Public	ID of the invited user.
3	GroupId	Guid	Public	ID of the target group.

3.9.1.24 SendgridConfig class

No	Attribute	Data Type	Visibility	Description
1	ApiKey	string	Public	API key for SendGrid service.
2	FromEmail	string	Public	Sender's email address.
3	FromName	string	Public	Sender's name.

3.9.1.25 MailRequest class

No	Attribute	Data Type	Visibility	Description
1	ToEmail	string	Public	Recipient's email address.
2	Subject	string	Public	Subject of the email.
3	Body	string	Public	Body content of the email.
4	Attachments	List<IFormFile>	Public	List of file attachments.

3.9.1.26 NotificationRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Adds a new entity to store

3.9.1.27 User class

No	Attribute	Data Type	Visibility	Description
1	FullName	string?	Public	Full name of the user.
2	NameUnsign	string?	Public	Full name without diacritics
3	Email	string?	Public	Email address of the user.
4	Password	string?	Public	User's password (if not using Google login).
5	Dob	DateTime?	Public	Date of birth.
6	Gender	Gender?	Public	Gender of the user.
7	Address	string?	Public	User’s address.
8	PhoneNumber	string?	Public	User’s phone number.
9	AvatarUrl	string?	Public	URL to the user's avatar image.
10	GoogleId	string?	Public	Google account ID (if signed in with Google).
11	IsVerified	bool?	Public	Whether the user's email is verified.
12	Role	RolesEnum?	Public	Role of the user
13	DeviceToken	string?	Public	Device token for push notifications.
14	Status	CommonsStatus?	Public	Current status of the user.
15	Assets	ICollection<Asset>	Public	List of user's assets.
16	Liabilities	ICollection<Liability>	Public	List of user's liabilities.
17	ChatHistories	ICollection<ChatHistory>	Public	Chat history records.

18	FinancialGoals	ICollection<FinancialGoal>	Public	User's financial goals.
19	FinancialReports	ICollection<FinancialReport>	Public	Generated financial reports.
20	GroupMembers	ICollection<GroupMember>	Public	User's membership in groups.
21	Notifications	ICollection<Notification>	Public	Notifications received by the user.
22	RecurringTransactions	ICollection<RecurringTransaction>	Public	Scheduled recurring transactions.
23	Subscriptions	ICollection<Subscription>	Public	User's active subscriptions.
24	Transactions	ICollection<Transaction>	Public	All transactions linked to the user.
25	UserQuizResults	ICollection<UserQuizResult>	Public	Results from financial quizzes.
26	UserSettings	ICollection<UserSetting>	Public	Custom user settings/preferences.
27	UserSpendingModels	ICollection<UserSpendingModel>	Public	Linked spending models.
28	BankAccounts	ICollection<BankAccount>	Public	Bank accounts linked to the user.

3.9.1.28 GroupFund class

No	Attribute	Data Type	Visibility	Description
1	Name	string	Public	The official name of the group fund.
2	NameUnsign	string	Public	Unaccented version of the group fund name.
3	Description	string	Public	Textual information describing the group fund.
4	CurrentBalance	decimal	Public	Current monetary balance available in the fund.
5	Status	GroupStatus?	Public	The status representing the group's activity state.
6	Visibility	VisibilityEnum?	Public	Defines the access level visibility of the group.

7	AccountBankId	Guid?	Public	Identifier of the linked bank account if any.
8	FinancialGoals	ICollection<FinancialGoal>	Public	List of financial goals related to the group.
9	FinancialReports	ICollection<FinancialReport>	Public	Collection of financial reports for the group.
10	GroupFundLogs	ICollection<GroupFundLog>	Public	Log entries for actions within the group fund.
11	GroupMembers	ICollection<GroupMember>	Public	Members who belong to the group fund.
12	Transactions	ICollection<Transaction>	Public	All transactions performed under the group.

3.9.1.29 GroupFundLog class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid	Public	ID of the related group fund.
2	Action	string?	Public	Performed action name.
3	ChangedBy	string?	Public	User who made the change.
4	ChangeDescription	string?	Public	Description of the change.
5	Group	GroupFund?	Public	Related group fund entity.

3.9.1.30 Notification class

No	Attribute	Data Type	Visibility	Description
1	IsRead	bool	Public	Indicates whether the notification has been read.
2	Title	string?	Public	Title of the notification.
3	TitleUnsign	string?	Public	Title without diacritics (for search and filtering).
4	Message	string?	Public	Content/message of the notification.
5	Href	string?	Public	Link to redirect when the notification is clicked.

6	EntityId	Guid?	Public	Related entity ID
7	UserId	Guid?	Public	ID of the user who receives this notification.
8	Type	NotificationType?	Public	Type/category of the notification
9	User	User?	Public	Navigation property to the related User entity.

3.9.1.31 GroupAction enum

No	Enum	Visibility	Description
1	CREATED	Public	Group has been created.
2	UPDATED	Public	Group information was updated.
3	DISBANDED	Public	Group has been disbanded.
4	INVITED	Public	Member has been invited.
5	JOINED	Public	Member has joined the group.
6	LEFT	Public	Member has left the group.
7	KICKED	Public	Member was removed from the group.
8	TRANSACTION_CREATED	Public	A transaction was created in group.
9	TRANSACTION_UPDATED	Public	A transaction was updated in group.
10	TRANSACTION_DELETED	Public	A transaction was deleted from group.

3.9.1.32 GroupMemberStatus enum

No	Enum	Visibility	Description
1	ACTIVE	Public	Member is active in the group.
2	PENDING	Public	Member's status is pending.
3	INACTIVE	Public	Member is inactive in the group.

3.9.2 Sequence Diagram

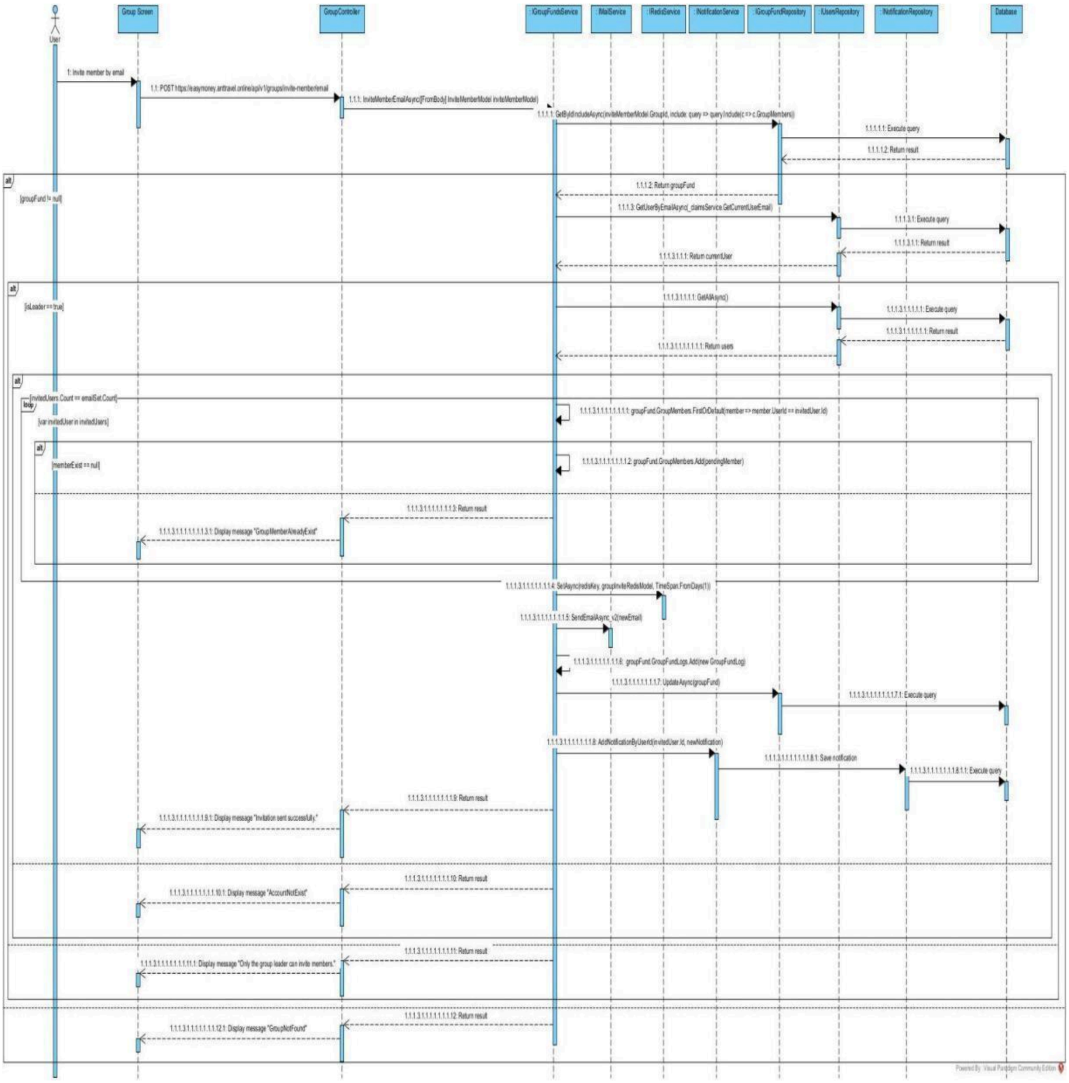


Figure 31 - Sequence Diagram

3.10 View dashboard statistics

3.10.1 Class Diagram

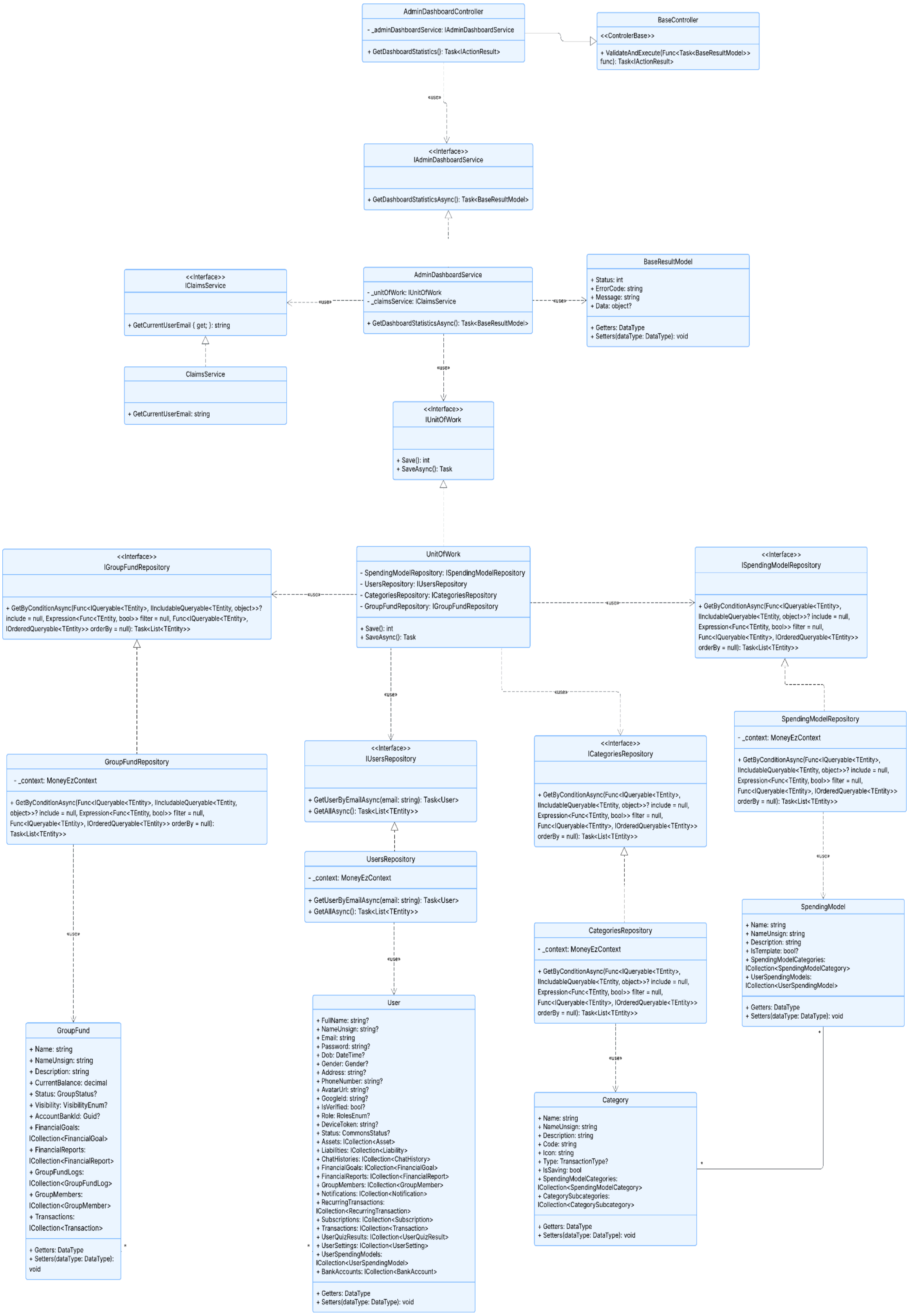


Figure 32 - Class Diagram

3.10.1.1 AdminDashboardController class

No	Attribute	Data Type	Visibility	Description
1	_adminDashboardService	IAdminDashboardService	Private	Service responsible for processing admin dashboard logic.

No	Method	Data Type	Visibility	Description
1	GetDashboardStatistics	Task<IActionResult>	Public	Retrieves summary statistics for dashboard display.

3.10.1.2 BaseController class

No	Method	Data Type	Visibility	Description
1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.10.1.3 IAdminDashboardService class

No	Method	Data Type	Visibility	Description
1	GetDashboardStatisticsAsync	Task<BaseResultModel>	Public	Asynchronously retrieves dashboard statistics.

3.10.1.4 AdminDashboardService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_claimsService	IClaimsService	Private	Service for handling user claims extracted from the token.

No	Method	Data Type	Visibility	Description
1	GetDashboardStatisticsAsync	Task<BaseResultModel>	Public	Asynchronously retrieves dashboard statistics.

3.10.1.5 BaseResultModel class

No	Attribute	Data Type	Visibility	Description
1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies the specific error if any occurred.
3	Message	string	Public	A message that describes the result or error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.10.1.6 IUnitOfWork class

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.10.1.7 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	GroupFundRepository	IGroupFundRepository	Public	Repository for managing sgroup fund data.
2	UsersRepository	IUsersRepository	Public	Repository for managing user data.
3	CategoriesRepository	ICategoriesRepository	Public	Repository for managing category data.
4	SpendingModelRepository	ISpendingModelRepository	Public	Repository for managing spending model data.

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.

2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.10.1.8 ClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Email extracted from current user claims.

3.10.1.9 IClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Gets the email of the current user.

3.10.1.10 IGroupFundRepository class

No	Method	Data Type	Visibility	Description
1	GetByConditionAsync	Task<List<TEntity>>	Public	Retrieves entities based on provided conditions.

3.10.1.11 IUsersRepository class

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Retrieves a user based on their email.
2	GetAllAsync	Task<List<TEntity>>	Public	Retrieves all entities from the database.

3.10.1.12 ICategoriesRepository class

No	Method	Data Type	Visibility	Description
1	GetByConditionAsync	Task<List<TEntity>>	Public	Retrieves entities based on provided conditions.

3.10.1.13 ISpendingModelRepository class

No	Method	Data Type	Visibility	Description
1	GetByConditionAsync	Task<List<TEntity>>	Public	Retrieves entities based on provided conditions.

3.10.1.14 GroupFundRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the

				database.

No	Method	Data Type	Visibility	Description
1	GetByConditionAsync	Task<List<TEntity>>	Public	Retrieves entities based on provided conditions.

3.10.1.15 UsersRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Retrieves a user based on their email.
2	GetAllAsync	Task<List<TEntity>>	Public	Retrieves all entities from the database.

3.10.1.16 CategoriesRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByConditionAsync	Task<List<TEntity>>	Public	Retrieves entities based on provided conditions.

3.10.1.17 SpendingModelRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByConditionAsync	Task<List<TEntity>>	Public	Retrieves entities based on provided conditions.

3.10.1.18 User class

No	Attribute	Data Type	Visibility	Description
1	FullName	string?	Public	Full name of the user.
2	NameUnsign	string?	Public	Full name without diacritics
3	Email	string?	Public	Email address of the user.
4	Password	string?	Public	User's password (if not using Google login).
5	Dob	DateTime?	Public	Date of birth.
6	Gender	Gender?	Public	Gender of the user.
7	Address	string?	Public	User's address.
8	PhoneNumber	string?	Public	User's phone number.
9	AvatarUrl	string?	Public	URL to the user's avatar image.
10	GoogleId	string?	Public	Google account ID (if signed in with Google).
11	IsVerified	bool?	Public	Whether the user's email is verified.
12	Role	RolesEnum?	Public	Role of the user
13	DeviceToken	string?	Public	Device token for push notifications.
14	Status	CommonsStatus?	Public	Current status of the user.
15	Assets	ICollection<Asset>	Public	List of user's assets.
16	Liabilities	ICollection<Liability>	Public	List of user's liabilities.

17	ChatHistories	ICollection<ChatHistory>	Public	Chat history records.
18	FinancialGoals	ICollection<FinancialGoal>	Public	User’s financial goals.
19	FinancialReports	ICollection<FinancialReport>	Public	Generated financial reports.
20	GroupMembers	ICollection<GroupMember>	Public	User’s membership in groups.
21	Notifications	ICollection<Notification>	Public	Notifications received by the user.
22	RecurringTransactions	ICollection<RecurringTransaction>	Public	Scheduled recurring transactions.
23	Subscriptions	ICollection<Subscription>	Public	User’s active subscriptions.
24	Transactions	ICollection<Transaction>	Public	All transactions linked to the user.
25	UserQuizResults	ICollection<UserQuizResult>	Public	Results from financial quizzes.
26	UserSettings	ICollection<UserSetting>	Public	Custom user settings/preferences.
27	UserSpendingModels	ICollection<UserSpendingModel>	Public	Linked spending models.
28	BankAccounts	ICollection<BankAccount>	Public	Bank accounts linked to the user.

3.10.1.19 GroupFund class

No	Attribute	Data Type	Visibility	Description
1	Name	string	Public	The official name of the group fund.
2	NameUnsign	string	Public	Unaccented version of the group fund name.
3	Description	string	Public	Textual information describing the group fund.
4	CurrentBalance	decimal	Public	Current monetary balance available in the fund.
5	Status	GroupStatus?	Public	The status representing the group's activity state.

6	Visibility	VisibilityEnum?	Public	Defines the access level visibility of the group.
7	AccountBankId	Guid?	Public	Identifier of the linked bank account if any.
8	FinancialGoals	ICollection<FinancialGoal>	Public	List of financial goals related to the group.
9	FinancialReports	ICollection<FinancialReport>	Public	Collection of financial reports for the group.
10	GroupFundLogs	ICollection<GroupFundLog>	Public	Log entries for actions within the group fund.
11	GroupMembers	ICollection<GroupMember>	Public	Members who belong to the group fund.
12	Transactions	ICollection<Transaction>	Public	All transactions performed under the group.

3.10.1.20 Category class

No	Attribute	Data Type	Visibility	Description
1	Name	string	Public	Name of the category.
2	NameUnsign	string	Public	Unaccented version of the name.
3	Description	string	Public	Short description of the category.
4	Code	string	Public	Unique code identifier.
5	Icon	string	Public	Icon associated with the category.
6	Type	TransactionType?	Public	Type of transaction.
7	IsSaving	bool	Public	Indicates if category is for saving.
8	SpendingModelCategories	ICollection<SpendingModelCategory>	Public	Related spending model categories.
9	CategorySubcategories	ICollection<CategorySubcategory>	Public	Related subcategories.

3.10.1.21 Category class

No	Attribute	Data Type	Visibility	Description

1	Name	string	Public	Name of the spending model.
2	NameUnsign	string	Public	Unaccented version of the name.
3	Description	string	Public	Short description of the category.
4	IsTemplate	bool?	Public	Indicates if this is a template.
5	SpendingModelCategories	ICollection<SpendingModelCategory>	Public	Related spending model categories.
6	UserSpendingModels	ICollection<UserSpendingModel>	Public	Related user spending models.

3.10.2 Sequence Diagram

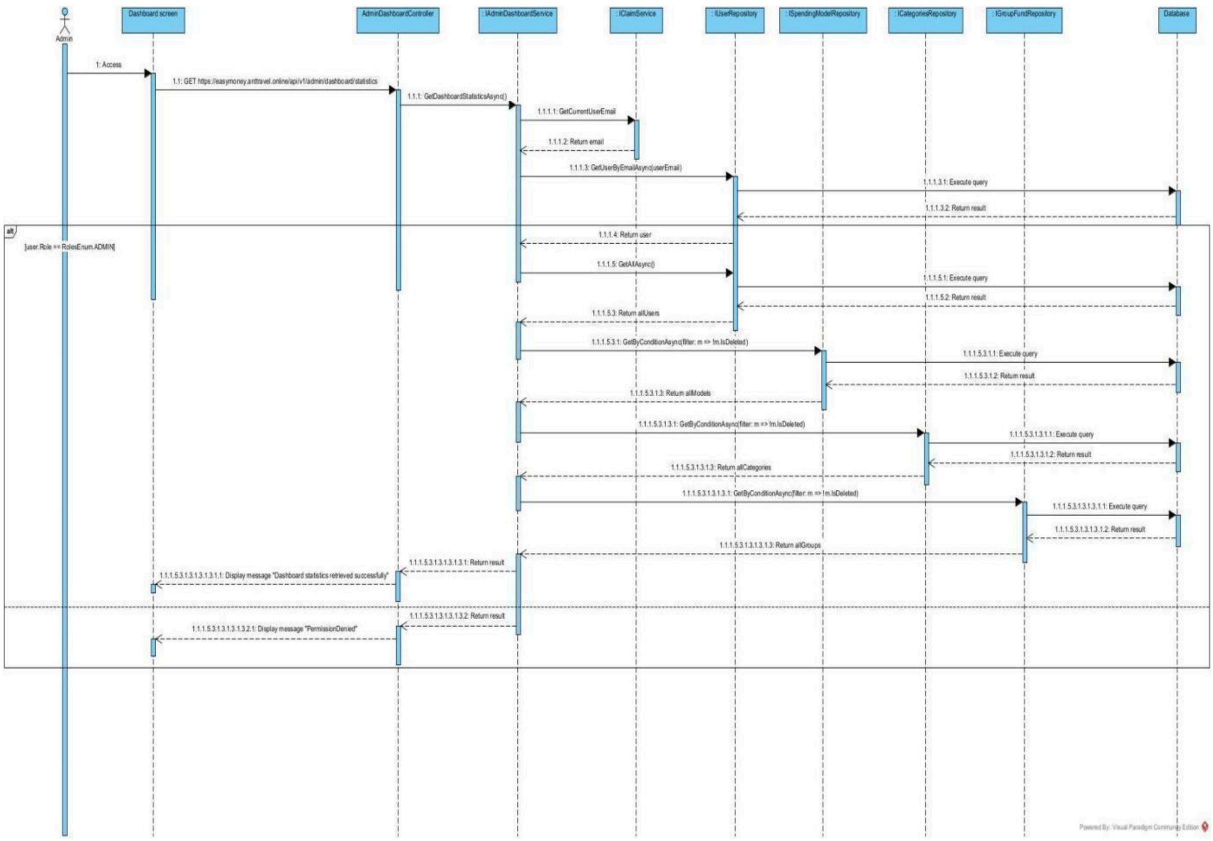


Figure 33 - Sequence Diagram

3.11 Add categories to spending model

3.11.1 Class Diagram

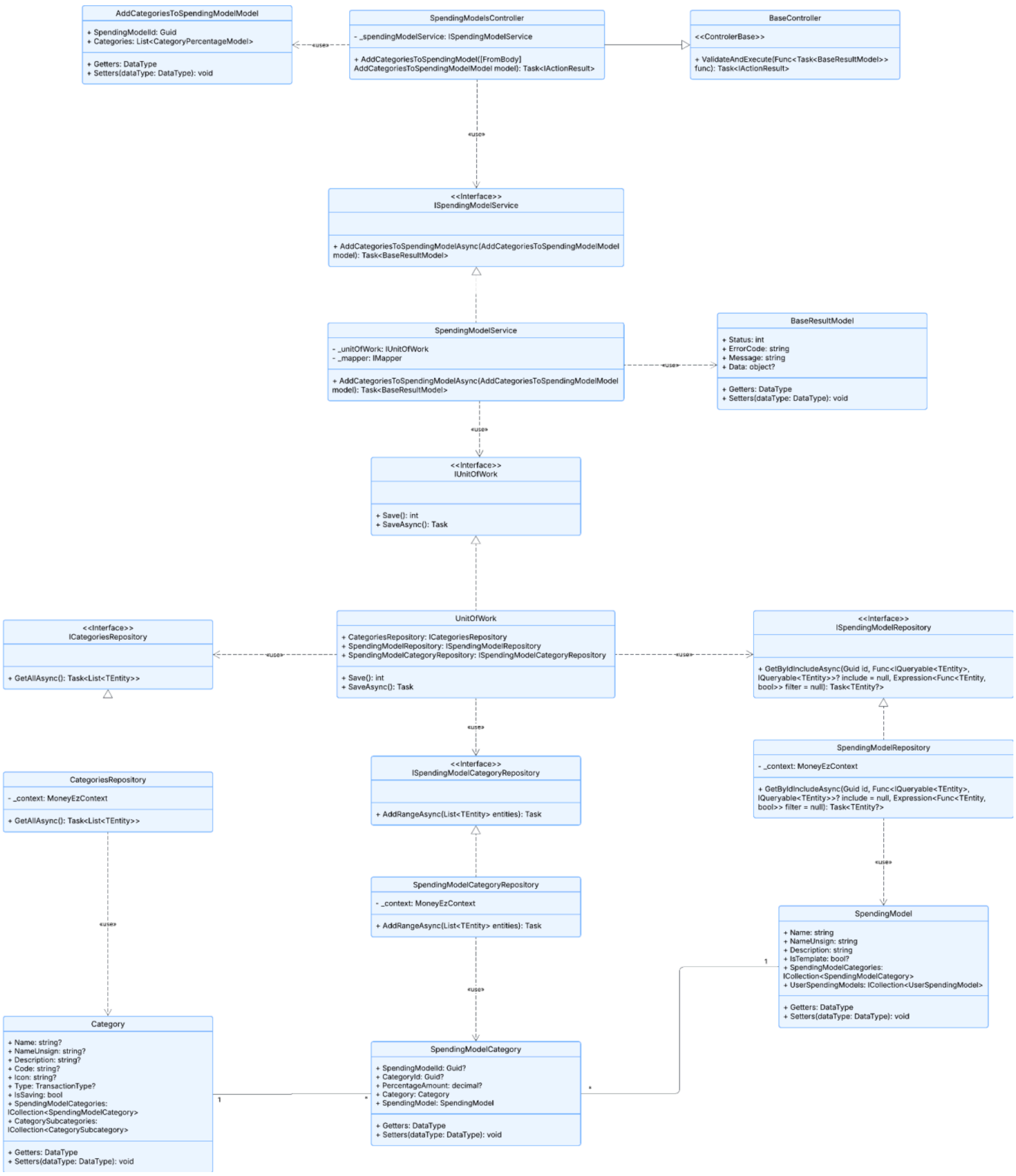


Figure 34 - Class Diagram

3.11.1.1 SpendingModelsController class

No	Attribute	Data Type	Visibility	Description
1	_adminDashboardService	IAdminDashboardService	Private	Service responsible for processing spending model logic.

No	Method	Data Type	Visibility	Description

1	AddCategoriesToSpendingModel	Task<IActionResult>	Public	Adds categories to a spending model.

3.11.1.2 BaseController class

No	Method	Data Type	Visibility	Description
1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.11.1.3 AddCategoriesToSpendingModelModel class

No	Attribute	Data Type	Visibility	Description
1	SpendingModelId	Guid	Public	Identifier of the spending model.
2	Categories	List<CategoryPercentageModel>	Public	List of categories with percentages.

3.11.1.4 ISpendingModelService class

No	Method	Data Type	Visibility	Description
1	AddCategoriesToSpendingModelAsync	Task<BaseResultModel>	Public	Asynchronously adds categories to a spending model.

3.11.1.5 SpendingModelService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.

No	Method	Data Type	Visibility	Description
1	AddCategoriesToSpendingModelAsync	Task<BaseResultModel>	Public	Asynchronously adds categories to a spending model.

3.11.1.6 BaseResultModel class

No	Attribute	Data Type	Visibility	Description
1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies

				the specific error if any occurred.
3	Message	string	Public	A message that describes the result or error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.11.1.7 IUnitOfWork class

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.11.1.8 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	SpendingModelRepository	ISpendingModelRepository	Public	Repository for managing spending model data.
2	CategoriesRepository	ICategoriesRepository	Public	Repository for managing category data.
3	SpendingModelCategoryRepository	ISpendingModelCategoryRepository	Public	Repository for managing spending model category data.

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.11.1.9 ISpendingModelRepository class

No	Method	Data Type	Visibility	Description
1	GetByIdIncludeAsync	Task<TEntity?>	Public	Retrieves entity by ID with includes.

3.11.1.10 ICategoriesRepository class

No	Method	Data Type	Visibility	Description
1	GetAllAsync	Task<List<TEntity>>	Public	Retrieves all entities.

3.11.1.11 ISpendingModelCategoryRepository class

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Adds a range of entities.

3.11.1.12 SpendingModelRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Adds a range of entities.
2	GetAllAsync	Task<List<TEntity>>	Public	Retrieves all entities.

3.11.1.13 CategoriesRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetAllAsync	Task<List<TEntity>>	Public	Retrieves all entities.

3.11.1.14 SpendingModelCategoryRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddRangeAsync	Task	Public	Adds a range of entities.

3.11.1.15 SpendingModel class

No	Attribute	Data Type	Visibility	Description
1	Name	string	Public	The name of the model.
2	NameUnsign	string	Public	The unsign version of the name.
3	Description	string	Public	A description of the model.
4	IsTemplate	bool?	Public	Whether it is a template.
5	SpendingModelCategories	ICollection<SpendingModelCategory>	Public	Collection of spending model categories.
6	UserSpendingModels	ICollection<UserSpendingModel>	Public	Collection of user spending models.

3.11.1.16 Category class

No	Attribute	Data Type	Visibility	Description
1	Name	string	Public	Name of the category.
2	NameUnsign	string	Public	Unaccented version of the name.
3	Description	string	Public	Short description of the category.
4	Code	string	Public	Unique code identifier.
5	Icon	string	Public	Icon associated with the category.
6	Type	TransactionType?	Public	Type of transaction.
7	IsSaving	bool	Public	Indicates if category is for saving.
8	SpendingModelCategories	ICollection<SpendingModelCategory>	Public	Related spending model categories.

9	CategorySubcategories	ICollection<CategorySubcategory>	Public	Related subcategories.

3.11.1.17 SpendingModelCategory class

No	Attribute	Data Type	Visibility	Description
1	SpendingModelId	Guid?	Public	ID of the associated spending model.
2	CategoryId	Guid?	Public	ID of the associated category.
3	PercentageAmount	decimal?	Public	Percentage amount for the category.
4	Category	Category	Public	Associated category object.
5	SpendingModel	SpendingModel	Public	Associated spending model object.

3.11.2 Sequence Diagram

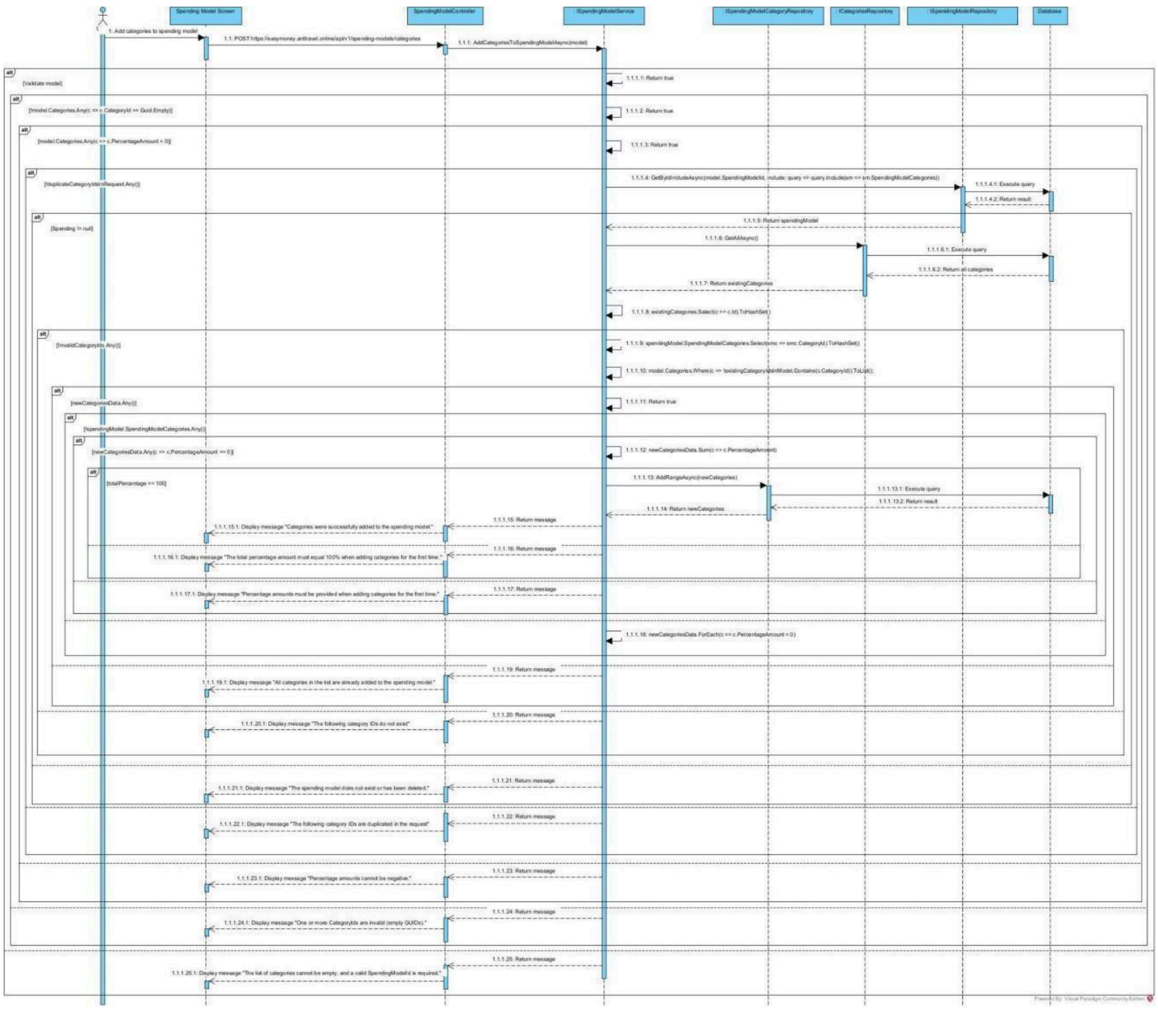


Figure 35 - Sequence Diagram

3.12 Add personal financial goal

3.12.1 Class Diagram

Figure 36 - Class Diagram

3.12.1.1 FinancialGoalsController class

No	Attribute	Data Type	Visibility	Description
1	_transactionService	ITransactionService	Private	Service responsible for processing transaction logic.

No	Method	Data Type	Visibility	Description
1	AddPersonalFinancialGoal	Task<IActionResult>	Public	Add a new personal financial goal.

3.12.1.2 BaseController class

No	Method	Data Type	Visibility	Description
1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.12.1.3 AddPersonalFinancialGoalModel class

No	Attribute	Data Type	Visibility	Description
1	TargetAmount	decimal	Public	Goal target amount.
2	SubcategoryId	Guid	Public	ID of the subcategory.
3	CurrentAmount	decimal	Public	Amount saved so far.
4	Status	FinancialGoalStatus	Public	Status of the financial goal.
5	ApprovalStatus	ApprovalStatus	Public	Approval state of the goal.

3.12.1.4 ISpendingModelService class

No	Method	Data Type	Visibility	Description
1	AddPersonalFinancialGoalAsync	Task<BaseResultModel>	Public	Add a personal financial goal.

3.12.1.5 SpendingModelService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.
3	_transactionNotificationService	ITransactionNotificationService	Private	Sends transaction-related notifications.
4	_claimsService	IClaimsService	Private	Service for handling user claims extracted from the token.

No	Method	Data Type	Visibility	Description
1	Add a personal financial goal	Task<BaseResultModel>	Public	Add a personal financial goal.
2	CalculateMaximumTargetAmountSubcategory	Task<decimal>	Private	Calculate the max target for subcategory.

3.12.1.6 ClaimsService class

No	Method	Data Type	Visibility	Description

1	GetCurrentUserEmail	string	Public	Email extracted from current user claims.

3.12.1.7 IClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Gets the email of the current user.

3.12.1.8 ITransactionNotificationService class

No	Method	Data Type	Visibility	Description
1	NotifyGoalAchievedAsync	Task	Public	Notifies user when a financial goal is achieved.
2	NotifyGoalProgressTrackingAsync	Task	Public	Notifies user about the progress of their financial goal.

3.12.1.9 TransactionNotificationService class

No	Attribute	Data Type	Visibility	Description
1	_notificationService	INotificationService	Private	Service for sending notifications to the user.

No	Method	Data Type	Visibility	Description
1	NotifyGoalAchievedAsync	Task	Public	Notifies user when a financial goal is achieved.
2	NotifyGoalProgressTrackingAsync	Task	Public	Notifies user about the progress of their financial goal.

3.12.1.10 INotificationService class

No	Method	Data Type	Visibility	Description
1	AddNotificationByUserId	Task<BaseResultModel>	Public	Adds a notification for a user by their ID.

3.12.1.11 NotificationService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.

2	_mapper	IMapper	Private	Maps between models and entities.

No	Method	Data Type	Visibility	Description
1	PushMessageFirebase	Task<bool>	Public	Sends a push notification to the user via Firebase.

3.12.1.12 FirebaseLibrary class

No	Method	Data Type	Visibility	Description
1	SendMessageFireBase	Task<string>	Public	Sends a Firebase message with the specified title, body, and token.

3.12.1.13 IUnitOfWork class

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.12.1.14 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	TransactionsRepository	ITransactionRepository	Public	Repository for managing transaction data.
2	UsersRepository	IUsersRepository	Public	Repository for managing user data.
3	SubcategoryRepository	ISubcategoryRepository	Public	Repository for managing subcategory data.
4	CategorySubcategoryRepository	ICategorySubcategoryRepository	Public	Repository for managing category-subcategory relationships.
5	UserSpendingModelRepository	IUserSpendingModelRepository	Public	Repository for managing user spending models.

6	FinancialGoalRepository	IFinancialGoalRepository	Public	Repository for managing financial goal data.
7	SpendingModelCategoryRepository	ISpendingModelCategoryRepository	Public	Repository for managing spending model-category relationships.
8	NotificationRepository	INotificationRepository	Public	Repository for managing notification data.

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.12.1.15 IUserSpendingModelRepository class

No	Method	Data Type	Visibility	Description
1	GetCurrentSpendingModelByUserId	Task<UserSpendingModel?>	Public	Get current spending model of a user.

3.12.1.16 ISpendingModelCategoryRepository class

No	Method	Data Type	Visibility	Description
1	GetByModelAndCategory	Task<SpendingModelCategory?>	Public	Get category by model and category ID.
2	GetByConditionAsync	Task<List<TEntity>>	Public	Get list of entities with conditions.

3.12.1.17 ICategorySubcategoryRepository class

No	Method	Data Type	Visibility	Description
1	GetCategoryBySubcategoryId	Task<Category?>	Public	Asynchronously retrieves the category associated with a subcategory ID.
2	GetByConditionAsync	Task<List<TEntity>>	Public	Retrieves a list of entities with optional

				filtering, including, and ordering.

3.12.1.18 ISubcategoryRepository class

No	Method	Data Type	Visibility	Description
1	GetByIdAsync	Task<TEntity>	Public	Asynchronously retrieves an entity by its ID.

3.12.1.19 ITransactionRepository class

No	Method	Data Type	Visibility	Description
1	GetByConditionAsync	Task<List<TEntity>>	Public	Retrieves entities based on filter, conditions, and order.
2	GetTotalIncomeByUserSpendingModelAsync	Task<decimal>	Public	Retrieves total income for a specific user spending model.

3.12.1.20 IUsersRepository class

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.

3.12.1.21 INotificationRepository class

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity to the repository and returns the added entity.

3.12.1.22 IFinancialGoalRepository class

No	Method	Data Type	Visibility	Description
1	GetActiveGoalByUserAndSubcategory	Task<FinancialGoal?>	Public	Retrieves active goal for user and subcategory.
2	GetByConditionAsync	Task<List<TEntity>>	Public	Retrieves entities based on filter and conditions.
3	AddAsync	Task<TEntity>	Public	Adds a new entity to the database.
4	UpdateAsync	void	Public	Updates an existing entity in the database.

3.12.1.23 UserSpendingModelRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetCurrentSpendingModelByUserId	Task<UserSpendingModel?>	Public	Get current spending model of a user.

3.12.1.24 SpendingModelCategoryRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByModelAndCategory	Task<SpendingModelCategory?>	Public	Get category by model and category ID.
2	GetByConditionAsync	Task<List<TEntity>>	Public	Get list of entities with conditions.

3.12.1.25 CategorySubcategoryRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetCategoryBySubcategoryId	Task<Category?>	Public	Asynchronously retrieves the category associated with a subcategory ID.

2	GetByConditionAsync	Task<List<TEntity>>	Public	Retrieves a list of entities with optional filtering, including, and ordering.

3.12.1.26 SubcategoryRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByIdAsync	Task<TEntity>	Public	Asynchronously retrieves an entity by its ID.

3.12.1.27 TransactionRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetByConditionAsync	Task<List<TEntity>>	Public	Retrieves entities based on filter, conditions, and order.
2	GetTotalIncomeByUserSpendingModelAsync	Task<decimal>	Public	Retrieves total income for a specific user spending model.

3.12.1.28 UsersRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.

3.12.1.29 NotificationRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzConte xt	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	AddAsync	Task<TEntity>	Public	Asynchronously adds an entity to the repository and returns the added entity.

3.12.1.30 FinancialGoalRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzConte xt	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetActiveGoalByUserAn dSubcategory	Task<Financial Goal?>	Public	Retrieves active goal for user and subcategory.
2	GetByConditionAsync	Task<List<TEn tity>>	Public	Retrieves entities based on filter and conditions.
3	AddAsync	Task<TEntity>	Public	Adds a new entity to the database.
4	UpdateAsync	void	Public	Updates an existing entity in the database.

3.12.1.31 BaseResultModel class

No	Attribute	Data Type	Visibility	Description

1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies the specific error if any occurred.
3	Message	string	Public	A message that describes the result or error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.12.1.32 UserSpendingModel class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid?	Public	The unique identifier for the user.
2	SpendingModelId	Guid?	Public	The unique identifier for the spending model.
3	PeriodUnit	int?	Public	The unit of the period.
4	PeriodValue	int?	Public	The value of the period.
5	StartDate	DateTime?	Public	The start date of the spending model.
6	EndDate	DateTime?	Public	The end date of the spending model.
7	Status	UserSpending ModelStatus? = ACTIVE	Public	The status of the user spending model, default is ACTIVE.
8	SpendingModel	SpendingMode l?	Public	The spending model associated with the user.
9	User	User?	Public	The user associated with the spending model.

3.12.1.33 SpendingModelCategory class

No	Attribute	Data Type	Visibility	Description
1	SpendingModelId	Guid?	Public	The ID of the associated spending model.

2	CategoryId	Guid?	Public	The ID of the associated category.
3	PercentageAmount	decimal?	Public	The allocated percentage amount for the category.
4	Category	Category	Public	The category linked to the spending model.
5	SpendingModel	SpendingModel	Public	The spending model that includes this category.

3.12.1.34 CategorySubcategory class

No	Attribute	Data Type	Visibility	Description
1	CategoryId	Guid?	Public	The ID of the associated category.
2	SubcategoryId	Guid?	Public	The ID of the associated subcategory.
3	Category	Category?	Public	Navigation property to the category.
4	Subcategory	Subcategory?	Public	Navigation property to the subcategory.

3.12.1.35 Subcategory class

No	Attribute	Data Type	Visibility	Description
1	Name	string?	Public	The name of the subcategory
2	NameUnsign	string?	Public	The unsigned version of the name
3	Description	string?	Public	A short description of the subcategory
4	Code	string?	Public	The code representing the subcategory
5	Icon	string?	Public	Icon associated with the subcategory
6	RecurringTransactions	ICollection<RecurringTransaction>	Public	Related recurring transactions.
7	Transactions	ICollection<Transaction>	Public	Related one-time transactions.
8	CategorySubcategories	ICollection<CategorySubcategory>	Public	Links to categories through relationships.

3.12.1.36 Transaction class

No	Attribute	Data Type	Visibility	Description
1	GroupId	Guid?	Public	ID of the group fund.
2	UserId	Guid?	Public	ID of the user who made the transaction.
3	Amount	decimal	Public	The amount of the transaction.
4	Type	TransactionType	Public	Type of transaction.
5	SubcategoryId	Guid?	Public	ID of the subcategory linked to this transaction.
6	TransactionDate	DateTime?	Public	The date the transaction occurred.
7	Description	string?	Public	Description of the transaction.
8	ApprovalRequired	bool?	Public	Whether approval is required.
9	RequestCode	string?	Public	Unique request code for approval tracking.
10	Status	TransactionStatus	Public	Current status of the transaction.
11	UserSpendingModelId	Guid?	Public	Related user's spending model ID.
12	InsertType	InsertType	Public	Indicates how the transaction was added.
13	Group	GroupFund?	Public	Navigation property to the related group.
14	Subcategory	Subcategory?	Public	Navigation to the subcategory.
15	TransactionVotes	ICollection<TransactionVote>	Public	Related transaction votes for approval.
16	User	User?	Public	Navigation property to the user

3.12.1.37 User class

No	Attribute	Data Type	Visibility	Description
1	FullName	string?	Public	Full name of the user.
2	NameUnsign	string?	Public	Full name without diacritics

3	Email	string?	Public	Email address of the user.
4	Password	string?	Public	User's password (if not using Google login).
5	Dob	DateTime?	Public	Date of birth.
6	Gender	Gender?	Public	Gender of the user.
7	Address	string?	Public	User’s address.
8	PhoneNumber	string?	Public	User’s phone number.
9	AvatarUrl	string?	Public	URL to the user's avatar image.
10	GoogleId	string?	Public	Google account ID (if signed in with Google).
11	IsVerified	bool?	Public	Whether the user's email is verified.
12	Role	RolesEnum?	Public	Role of the user
13	DeviceToken	string?	Public	Device token for push notifications.
14	Status	CommonsStatus?	Public	Current status of the user.
15	Assets	ICollection<Asset>	Public	List of user's assets.
16	Liabilities	ICollection<Liability>	Public	List of user's liabilities.
17	ChatHistories	ICollection<ChatHistory>	Public	Chat history records.
18	FinancialGoals	ICollection<FinancialGoal>	Public	User’s financial goals.
19	FinancialReports	ICollection<FinancialReport>	Public	Generated financial reports.
20	GroupMembers	ICollection<GroupMember>	Public	User’s membership in groups.
21	Notifications	ICollection<Notification>	Public	Notifications received by the user.
22	RecurringTransactions	ICollection<RecurringTransaction>	Public	Scheduled recurring transactions.

23	Subscriptions	ICollection<Subscription>	Public	User's active subscriptions.
24	Transactions	ICollection<Transaction>	Public	All transactions linked to the user.
25	UserQuizResults	ICollection<UserQuizResult>	Public	Results from financial quizzes.
26	UserSettings	ICollection<UserSetting>	Public	Custom user settings/preferences.
27	UserSpendingModels	ICollection<UserSpendingModel>	Public	Linked spending models.
28	BankAccounts	ICollection<BankAccount>	Public	Bank accounts linked to the user.

3.12.1.38 Notification class

No	Attribute	Data Type	Visibility	Description
1	IsRead	bool	Public	Indicates whether the notification has been read.
2	Title	string?	Public	Title of the notification.
3	TitleUnsign	string?	Public	Title without diacritics (for search and filtering).
4	Message	string?	Public	Content/message of the notification.
5	Href	string?	Public	Link to redirect when the notification is clicked.
6	EntityId	Guid?	Public	Related entity ID
7	UserId	Guid?	Public	ID of the user who receives this notification.
8	Type	NotificationType?	Public	Type/category of the notification
9	User	User?	Public	Navigation property to the related User entity.

3.12.1.39 FinancialGoal class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid?	Public	ID of the user who owns the financial

				goal.
2	GroupId	Guid?	Public	Optional ID of the group associated with the goal.
3	SubcategoryId	Guid?	Public	Optional subcategory linked to the goal.
4	Name	string	Public	Name/title of the financial goal.
5	NameUnsign	string	Public	Name without diacritics.
6	TargetAmount	decimal	Public	The total amount to be saved or achieved.
7	CurrentAmount	decimal	Public	Current saved amount toward the goal.
8	StartDate	DateTime	Public	Date the goal was started.
9	Deadline	DateTime	Public	Deadline to reach the financial goal.
10	Status	FinancialGoalStatus	Public	Current status of the goal.
11	ApprovalStatus	ApprovalStatus	Public	Approval status for goals that require validation.
12	IsSaving	bool	Public	Whether the goal is currently in saving mode.
13	Group	GroupFund?	Public	Navigation property to related group, if any.
14	User	User	Public	Navigation property to the associated user.
15	Subcategory	Subcategory?	Public	Navigation property to linked subcategory.

3.12.1.40 ApprovalStatus enum

No	Enum	Visibility	Description
1	PENDING	Public	The entity is awaiting approval.
2	APPROVED	Public	The entity has been approved.
3	REJECTED	Public	The entity has been rejected.

3.12.1.41 TransactionStatus enum

No	Enum	Visibility	Description
1	PENDING	Public	The transaction is pending approval.
2	APPROVED	Public	The transaction has been approved.
3	REJECTED	Public	The transaction has been rejected.

3.12.1.42 FinancialGoalStatus enum

No	Enum	Visibility	Description
1	PENDING	Public	The financial goal is awaiting action or approval.
2	ACTIVE	Public	The financial goal is currently being pursued or worked on.
3	ARCHIVED	Public	The financial goal has been archived and is no longer active.
4	COMPLETED	Public	The financial goal has been successfully achieved.

3.12.1.43 NotificationType enum

No	Enum	Visibility	Description
1	SYSTEM	Public	System-generated notification for all users
2	USER	Public	Personal notification targeted to an individual user
3	GROUP	Public	Notification related to group activity
4	GROUP_INVITE	Public	Invitation to join a group
5	FINANCIAL_GOAL_PERSONAL	Public	Notification regarding a personal financial goal
6	FINANCIAL_GOAL_GROUP	Public	Notification related to a group financial goal

3.12.2 Sequence Diagram

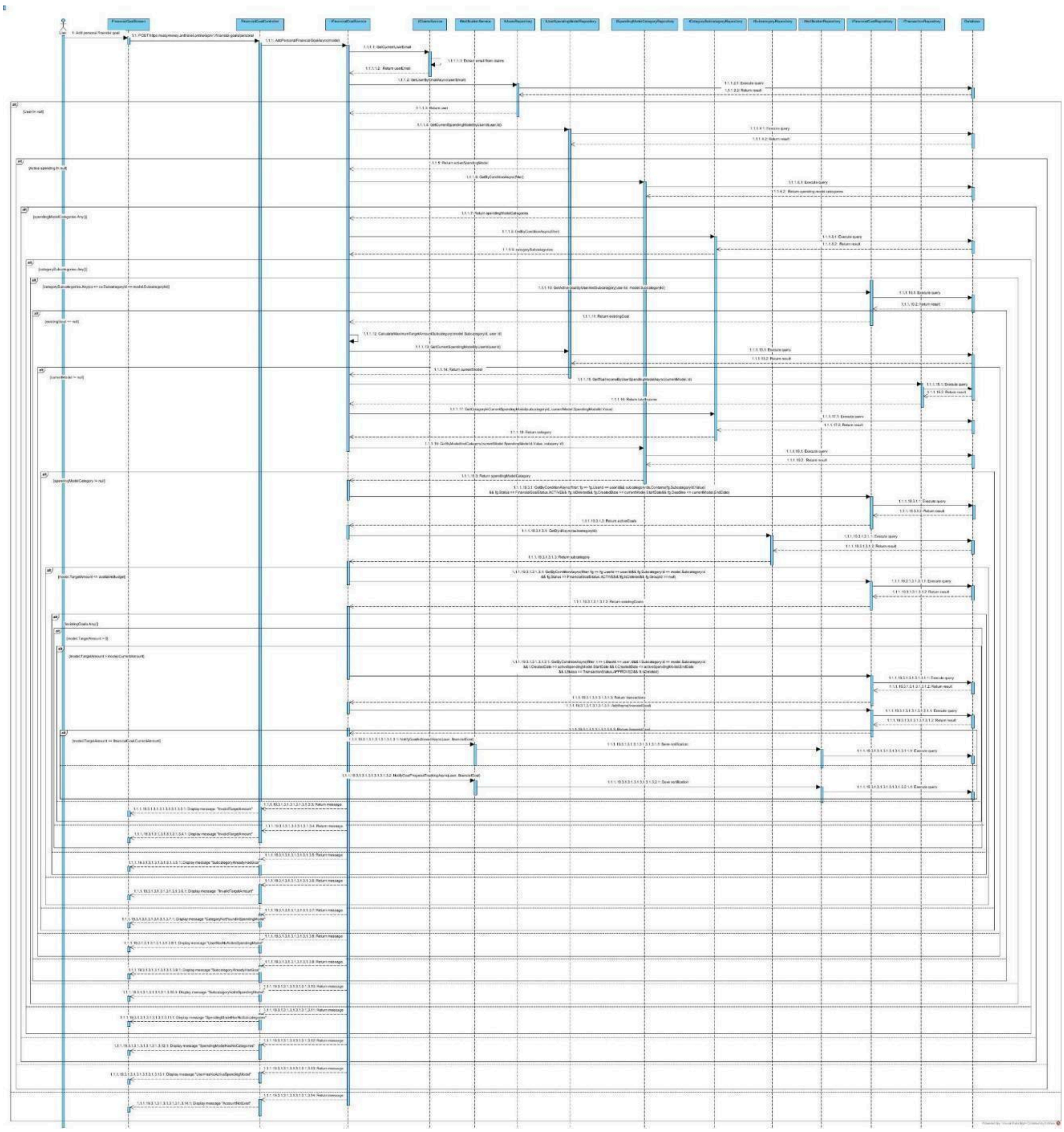


Figure 37 - Sequence Diagram

3.13 View personal financial goal

3.13.1 Class Diagram

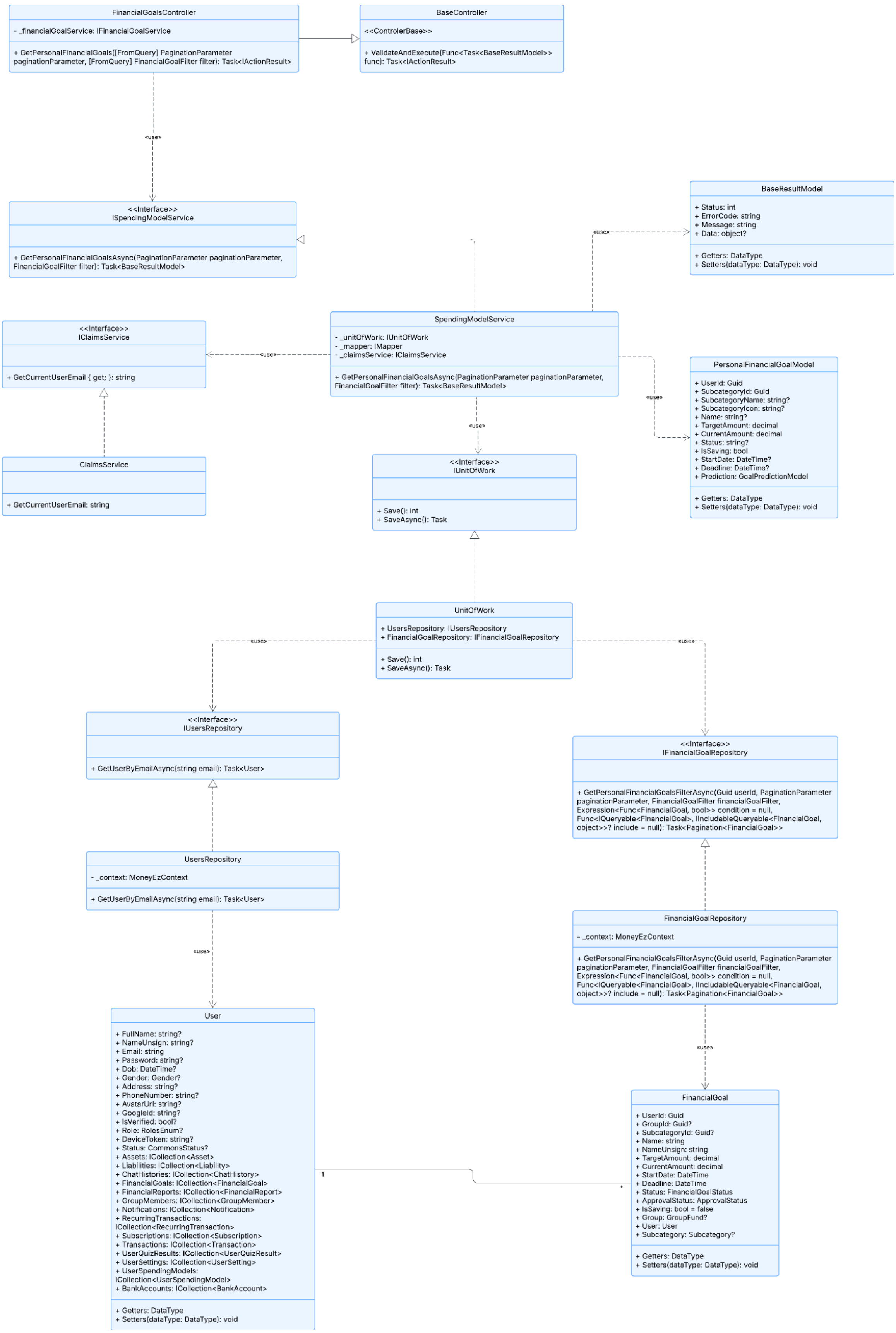


Figure 38 - Class Diagram

3.13.1.1 FinancialGoalsController class

No	Attribute	Data Type	Visibility	Description
1	_transactionService	ITransactionService	Private	Service responsible for processing transaction logic.

No	Method	Data Type	Visibility	Description
1	GetPersonalFinancialGoals	Task<IActionResult>	Public	Retrieves personal financial goals with pagination.

3.13.1.2 BaseController class

No	Method	Data Type	Visibility	Description
1	ValidateAndExecute	Task<IActionResult>	Public	Validates model and executes given function.

3.13.1.3 ISpendingModelService class

No	Method	Data Type	Visibility	Description
1	GetPersonalFinancialGoalsAsync	Task<BaseResultModel>	Public	Retrieves personal financial goals with pagination.

3.13.1.4 SpendingModelService class

No	Attribute	Data Type	Visibility	Description
1	_unitOfWork	IUnitOfWork	Private	Handles data access operations.
2	_mapper	IMapper	Private	Maps between models and entities.
3	_claimsService	IClaimsService	Private	Service for handling user claims extracted from the token.

No	Method	Data Type	Visibility	Description
1	GetPersonalFinancialGoalsAsync	Task<BaseResultModel>	Public	Retrieves personal financial goals with pagination.

3.13.1.5 ClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Email extracted from current user claims.

3.13.1.6 IClaimsService class

No	Method	Data Type	Visibility	Description
1	GetCurrentUserEmail	string	Public	Gets the email of the current user.

3.13.1.7 IUnitOfWork class

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.13.1.8 UnitOfWork class

No	Attribute	Data Type	Visibility	Description
1	FinancialGoalRepository	IFinancialGoal Repository	Public	Repository for managing financial goal data.
2	UsersRepository	IUsersReposito ry	Public	Repository for managing user data.

No	Method	Data Type	Visibility	Description
1	Save()	int	Public	Saves changes and returns the number of affected rows.
2	SaveAsync()	Task	Public	Asynchronously saves changes.

3.13.1.9 IUsersRepository class

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.

3.13.1.10 IFinancialGoalRepository class

No	Method	Data Type	Visibility	Description
1	GetPersonalFinancialGoalsFilterAsync	Task<Pagination<FinancialGoal>>	Public	Retrieves filtered personal financial goals with pagination.

3.13.1.11 UsersRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetUserByEmailAsync	Task<User>	Public	Asynchronously retrieves a user by their email.

3.13.1.12 FinancialGoalRepository class

No	Attribute	Data Type	Visibility	Description
1	_context	MoneyEzContext	Private	Database context used to interact with the database.

No	Method	Data Type	Visibility	Description
1	GetPersonalFinancialGoalsFilterAsync	Task<Pagination<FinancialGoal>>	Public	Retrieves filtered personal financial goals with pagination.

3.13.1.13 BaseResultModel class

No	Attribute	Data Type	Visibility	Description
1	Status	int	Public	Represents the status code of the response.
2	ErrorCode	string	Public	Code that identifies the specific error if any occurred.
3	Message	string	Public	A message that describes the result or

				error in detail.
4	Data	object?	Public	The result data payload returned from the operation.

3.13.1.14 User class

No	Attribute	Data Type	Visibility	Description
1	FullName	string?	Public	Full name of the user.
2	NameUnsign	string?	Public	Full name without diacritics
3	Email	string?	Public	Email address of the user.
4	Password	string?	Public	User's password (if not using Google login).
5	Dob	DateTime?	Public	Date of birth.
6	Gender	Gender?	Public	Gender of the user.
7	Address	string?	Public	User’s address.
8	PhoneNumber	string?	Public	User’s phone number.
9	AvatarUrl	string?	Public	URL to the user's avatar image.
10	GoogleId	string?	Public	Google account ID (if signed in with Google).
11	IsVerified	bool?	Public	Whether the user's email is verified.
12	Role	RolesEnum?	Public	Role of the user
13	DeviceToken	string?	Public	Device token for push notifications.
14	Status	CommonsStatus?	Public	Current status of the user.
15	Assets	ICollection<Asset>	Public	List of user's assets.
16	Liabilities	ICollection<Liability>	Public	List of user's liabilities.
17	ChatHistories	ICollection<ChatHistory>	Public	Chat history records.

18	FinancialGoals	ICollection<FinancialGoal>	Public	User's financial goals.
19	FinancialReports	ICollection<FinancialReport>	Public	Generated financial reports.
20	GroupMembers	ICollection<GroupMember>	Public	User's membership in groups.
21	Notifications	ICollection<Notification>	Public	Notifications received by the user.
22	RecurringTransactions	ICollection<RecurringTransaction>	Public	Scheduled recurring transactions.
23	Subscriptions	ICollection<Subscription>	Public	User's active subscriptions.
24	Transactions	ICollection<Transaction>	Public	All transactions linked to the user.
25	UserQuizResults	ICollection<UserQuizResult>	Public	Results from financial quizzes.
26	UserSettings	ICollection<UserSetting>	Public	Custom user settings/preferences.
27	UserSpendingModels	ICollection<UserSpendingModel>	Public	Linked spending models.
28	BankAccounts	ICollection<BankAccount>	Public	Bank accounts linked to the user.

3.13.1.15 FinancialGoal class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid?	Public	ID of the user who owns the financial goal.
2	GroupId	Guid?	Public	Optional ID of the group associated with the goal.
3	SubcategoryId	Guid?	Public	Optional subcategory linked to the goal.
4	Name	string	Public	Name/title of the financial goal.
5	NameUnsign	string	Public	Name without diacritics.
6	TargetAmount	decimal	Public	The total amount to be saved or achieved.
7	CurrentAmount	decimal	Public	Current saved amount toward the goal.

8	StartDate	DateTime	Public	Date the goal was started.
9	Deadline	DateTime	Public	Deadline to reach the financial goal.
10	Status	FinancialGoalStatus	Public	Current status of the goal.
11	ApprovalStatus	ApprovalStatus	Public	Approval status for goals that require validation.
12	IsSaving	bool	Public	Whether the goal is currently in saving mode.
13	Group	GroupFund?	Public	Navigation property to related group, if any.
14	User	User	Public	Navigation property to the associated user.
15	Subcategory	Subcategory?	Public	Navigation property to linked subcategory.

3.13.1.16 PersonalFinancialGoalModel class

No	Attribute	Data Type	Visibility	Description
1	UserId	Guid	Public	User's unique identifier.
2	SubcategoryId	Guid	Public	Subcategory's unique identifier.
3	SubcategoryName	string?	Public	Name of the subcategory.
4	SubcategoryIcon	string?	Public	Icon representing the subcategory.
5	Name	string?	Public	Name of the financial goal.
6	TargetAmount	decimal	Public	Target amount for the financial goal.
7	CurrentAmount	decimal	Public	Current amount accumulated towards the goal.
8	Status	string?	Public	Current status of the financial goal.
9	IsSaving	bool	Public	Indicates if the goal is saving-focused.
10	StartDate	DateTime?	Public	Start date of the financial goal.

11	Deadline	DateTime?	Public	Deadline for achieving the goal.
12	Prediction	GoalPrediction Model	Public	Model containing the predicted financial progress.

3.13.2 Sequence Diagram

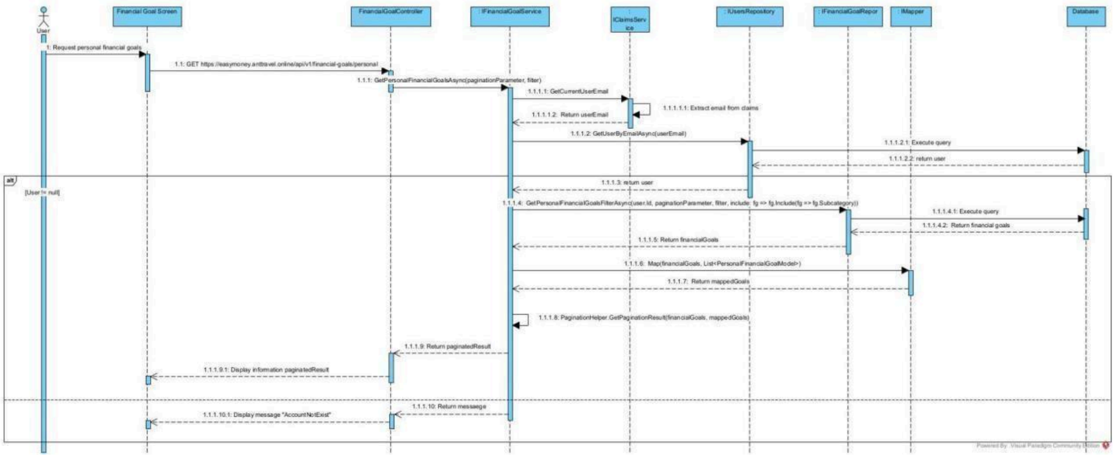


Figure 39 - Sequence Diagram

V. Software Testing Documentation

1. Scope of Testing

Table 31 - Scope of testing

Feature	Function	Role
Account	Register	User
	Login	Admin/User
	Logout	
	Change password	
	Forgot password	
	Get list account	Admin
	Create account	
	Delete account	
	Get account information	
	Update account information	
Notification	Alerts and reminders	System handler
	Transaction update	
	Send mail	
Quiz management	Create quiz	Admin
	Modify quiz	
	Update quiz	
	Remove quiz	
Fund management	Raising fund	User
	Withdraw fund	

Statistic	Modify statistic	Admin/User

2. Test Strategy

2.1 Testing Types

Table 32 - Testing types

Objective	**Technique**	**Completion criteria**
Testing the verification and validation of units of code	Function Testing	Method, class, etc.
Testing the entire software system to ensure its functionality	System Testing	Software requirements, executed test cases, functional requirements, etc.
Testing different components of the system when they are combined together	Integration testing	Modules
Testing application environment in real situations	End-to-end testing	Database, third-party payment, etc.

2.2 Test Levels

Table 33 - Test levels

Type of Tests	**Test Level**			
	Unit	**Integration**	**System**	**Acceptance**
Function Testing	X			
Iteration Testing	X	X		
System Testing			X	

2.3 Supporting Tools

Table 43 - Supporting Tools

Purpose	Tool	Vendor/In-house	Version
API	Postman	Postman	11.22
API	Swagger	Open Source	

3. Test Plan

3.1 Human Resources

Table 34 - Human Resources

Worker/Doer	Role	Specific Responsibilities/Comments
Phạm Nhật Linh	Leader	Planning, verifying test deliverables, do interface testing as planned
Nguyễn Thương Huyền	Member	Planning, verifying test deliverables, do interface testing as planned
Đặng Ngọc Minh Trí	Member	Planning, verifying test deliverables, do interface testing as planned
Dương Tôn Bảo	Member	Planning, verifying test deliverables, do interface testing as planned
Đặng Phan Gia Đức	Member	Planning, verifying test deliverables, do interface testing as planned

3.2 Test Environment

Table 35 – Test Environment

Purpose	Tool	Provider	Version
Browser Testing	Google Chrome	Google Chrome	134.0.6998.178
Browser Testing	Internet Explorer	Internet Explorer	11
Mobile Testing		Vivo 5a	OS 14.0

3.3 Test Milestones

Table 36 - Test Milestones

Milestone	Task	Start Date	End Date
Milestone 1	Test Register (Usẻ)	14/02/2025	20/02/2025

	Test Login (All roles)	14/02/2025	20/02/2025
	Test Logout (All roles)	14/02/2025	20/02/2025
	Test Change Password	14/02/2025	20/02/2025
	Test Forgot Password	14/02/2025	20/02/2025
	Test Get List Account (Admin)	14/02/2025	20/02/2025
	Test Create Account (Admin)	14/02/2025	20/02/2025
	Test Delete Account (Admin)	14/02/2025	20/02/2025
	Test Update Account Information	14/02/2025	20/02/2025
	Test Get Account Information	14/02/2025	20/02/2025
Milestone 2	Test Raising Fund (User)	20/02/2025	28/02/2025
	Test Withdraw Fund (User)	20/02/2025	28/02/2025
Milestone 3	Test Transaction History	24/10/2024	30/10/2024
Milestone 4	Test Create Quiz	01/03/2025	06/03/2025
	Test Modify Quiz	01/03/2025	06/03/2025
	Test Remove Quiz	01/03/2025	06/03/2025
Milestone 5	Test Statistic	07/03/2025	12/03/2025
Milestone 6	Test Group Management	13/03/2025	20/03/2025
Milestone 7	Test Account Management Integration	21/03/2025	31/03/2025
	Test Transaction and Group Integration	21/03/2025	31/03/2025
	Test Notification and Send Mail	21/03/2025	31/03/2025
Milestone 8	Perform End-to-End Testing for all features	21/03/2025	31/03/2025
	Test Performance and Load Handling	21/03/2025	31/03/2025
	Test Usability and Compatibility	21/03/2025	31/03/2025

2. Test Cases & Test Reports

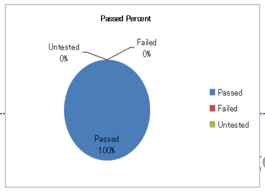
Table 37 - Test Case & Test Report

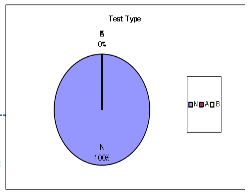
Unit Test Cases	Test Report

SP25SE065_GSP25_Unit_Test

Sub total	56	0	0	56	0	0	56

Test coverage	**100.00** %
Test successful coverage	**100.00** %
Normal case	**100.00** %
Abnormal case	**0.00** %
Boundary case	**0.00** %





SP25SE065_GSP25_Test_Report

TEST STATISTICS

Project Name	*MoneyEasy - Financial Management A*	**Creator**		
Project Code	*SP25SE065*	**Reviewer/Approver**		
Document Code	*SP25SE065_Test Report_v1.0*	**Issue Date**		*2/25/2025*
Notes	*<List modules included in this release> ex: Release 1 includes 2 modules: Module1 and Module2*			

No	Module code	Passed	Failed	Pending	N/A	Number of test cases
1	Login	32	0	0	0	32
2	Account Registration	53	0	0	0	53
3	Create Group	27	0	0	0	27
4	Raising Fund	68	0	0	0	68
5	Withdraw Fund	61	0	0	0	61
6	Quiz Management	37	0	0	0	37
7	Statistic	35	0	0	0	35
	Sub total	313	0	0	0	313

Test coverage	**100.00** %
Test successful coverage	**100.00** %

VI. Release Package & User Guides

1. Deliverable Package

No.	Deliverable Item	Description
1	Schedule/Task Tracking	GSP25SE47_SP25SE065_MoneyEasy
2	Project Backlog	GSP25SE47_SP25SE065_MoneyEasy
3	Source Codes	SRC_CapstoneProjectSP25_SP25SE065
4	Database Script(s)	SRC_CapstoneProjectSP25_SP25SE065
5	Final Report Document	CapstoneProjectSP25_SP25SE065_Final_Project_Report
6	Test Cases Document	GSP25SE47_SP25SE065_Test_Report GSP25SE47_SP25SE065_Unit_Test
7	Slide	GSP25SE47_SP25SE065_Slide

2. Installation Guides

2.1 System Requirements

2.1.1 Software Requirements

Backend System

Table 38 – Backend System

Component	Name and Version
Programming Language	.NET 8
Database	SQL Server
ORM	Entity Framework
Caching	Redis
Authentication	JWT
IDE	Visual Studio
API	Swagger

Front-end System

Table 39 – Front-end System

Component	Name and Version
Programming Language	Typescript
IDE	Android Studio, Visual Studio Code

Cloud & Infrastructure

Table 40 – Cloud & Infrastructure

Component	Name and Version
Cloud Provider	Digital Ocean
Containerization	Docker Container
CI/CD	GitHub Actions, Selfhost runner

Third-Party Integrations

Table 41 – Third-Party Integrations

Component	Name and Version
Firebase	for authentication and user management.
Email Services	SendGrid
SignalR	Real-time communication
Quartz.NET	Quartz is used for job scheduling

2.1.2 Hardware Requirements

Web Application

Table 42 – Web Application

PC	Minimum	Recommended
Internet connection	Cable, Wi-fi	Cable, Wi-fi
Processor	Intel Core i5	Intel Core i7
Memory	8GB RAM	16GB RAM
Storage	HDD 256GB	SSD 256GB
Web Browser	Google Chrome	Google Chrome 131.0.6778.108

Mobile Application

Table 43 – Mobile Application

Component	Name and Version
Internet connection	Cable, Wi-fi
Operating system	Android: Version 8.0+

Minimum Screen Size	5 inches
Storage	At least 1GB

2.2 Installation Instruction

2.2.1 Prerequisites

- Windows 10/11 or (Latest Version)
- SQL Server 2022 or Azure Data Studio
- Git

2.2.2 Backend Setup

Prerequisites

- .NET 8 SDK
- Visual Studio 2022
- Redis

1. Clone the source code:

git clone https://github.com/MoneyEZ-FUHCM/moneyez-api.git
cd moneyez-api

2. Open appsettings.json
3. Configure connection string in appSettings:

"MoneyEzLocal":
"Server={your_server_name};Uid=sa;Pwd={your_password};Database=MoneyEz;TrustServerCertificate=True",

4. Entity Framework Migration:

dotnet ef database update

5. Install Dependencies:

dotnet restore

6. Test the project in your browser:

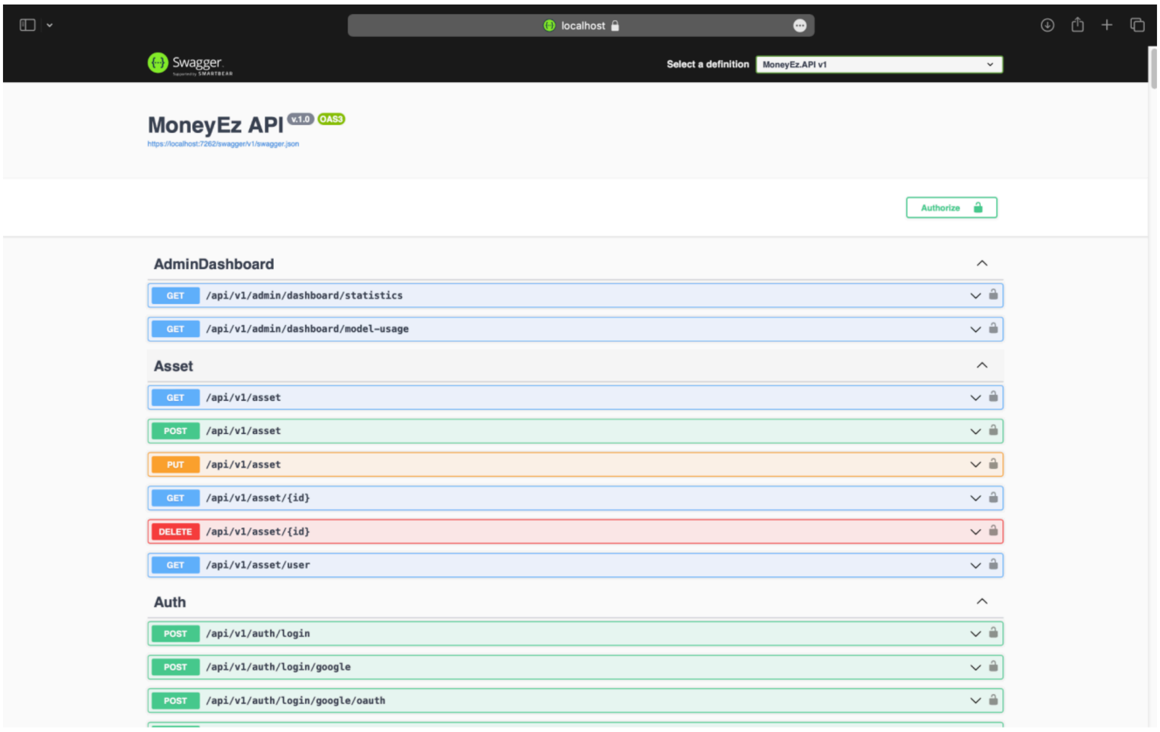


Figure 40 - Test swagger

2.2.3 Mobile Setup

Prerequisites

- Visual Studio code
- Android Studio
- Npm or yarn
- Firebase account

1. Clone the source code :

git clone https://github.com/MoneyEZ-FUHCM/moneyez-mobile.git

cd moneyez-mobile

2. Setup Dependencies

Run command: yarn install

3. Configuration

Create .env with code:

EXPO_PUBLIC_API_KEY=your_api_key
EXPO_PUBLIC_AUTH_DOMAIN=your_auth_domain
EXPO_PUBLIC_PROJECT_ID=your_project_id
EXPO_PUBLIC_STORAGE_BUCKET=your_storage_bucket
EXPO_PUBLIC_MESSAGING_SENDER_ID=your_messaging_sender_id
EXPO_PUBLIC_APP_ID=your_app_id
EXPO_PUBLIC_MEASUREMENT_ID= your_mesasurement_id
EXPO_PUBLIC_BASE_URL=https://easymoney.anttravel.online/api/v1
EXPO_PUBLIC_GOOGLE_CLIENT_ID=your_google_client_id

create image.d.ts with code:
declare module "*.png" { const value: any; export default value; }
declare module "*.jpg" { const value: any; export default value; }

create env.d.ts with code:
declare module "@env" {
export const WEBCLIENT_ID: string;
export const API_KEY: string;
export const AUTH_DOMAIN: string;
export const PROJECT_ID: string;
export const STORAGE_BUCKET: string;
export const MESSAGING_SENDER_ID: string;
export const APP_ID: string;
export const MEASUREMENT_ID: string;
export const BASE_URL: string; }

generate google-services.json in the Firebase website

4. Project run:
yarn android

2.2.3 Web Setup

Prerequisites

- Nextjs 14 or (Latest Version)
- Npm or yarn
- Node.js version >= v18.17.0

1. git clone https://github.com/MoneyEZ-FUHCM/moneyez-web.git
cd moneyez-web
2. Setup Dependencies
Run command: *yarn install* or *npm i*
3. Project run:
npm run dev or *yarn dev*
Project will shown on http://localhost:3000/moneyez-web

2.2.4 Bank-Web Setup

- Npm or yarn
- Nextjs 14 or (Latest Version)

1. git clone https://github.com/MoneyEZ-FUHCM/moneyez-bank-web.git
cd moneyez-bank-web
2. Setup Dependencies

Run command: *yarn install* or *npm i*
3. Project run:
npm run dev or *yarn dev*
Project will shown on http://localhost:3000/moneyez-bank-web/auth

3. User Manual

3.1 Overview

Table 44 - Overview

No.	Workflow	Actor
1	Register/Login and Choose Preferred Spending Model	User
2	Personal Transaction Management & Smart Goal Tracking - AI Drived	User
3	Group Transaction Management	User

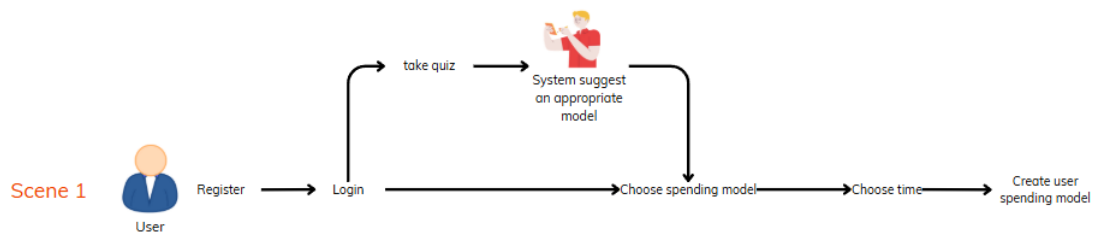


Figure 41 – Register/Login and Choose Preferred Spending Model

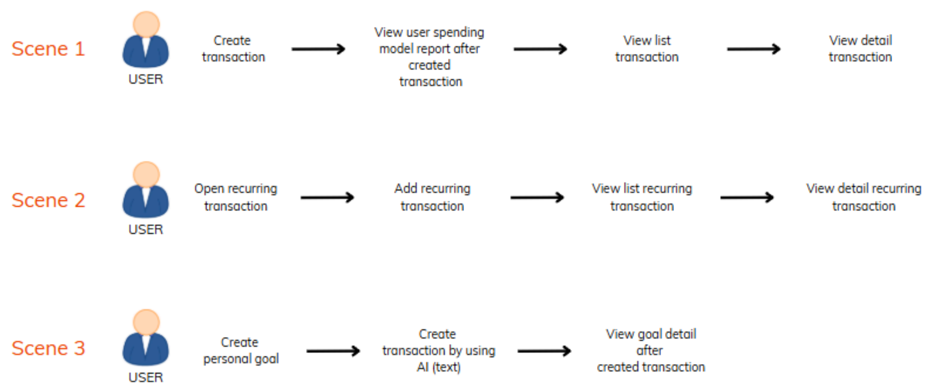


Figure 42 – Personal Transaction Management & Smart Goal Tracking - AI Drived

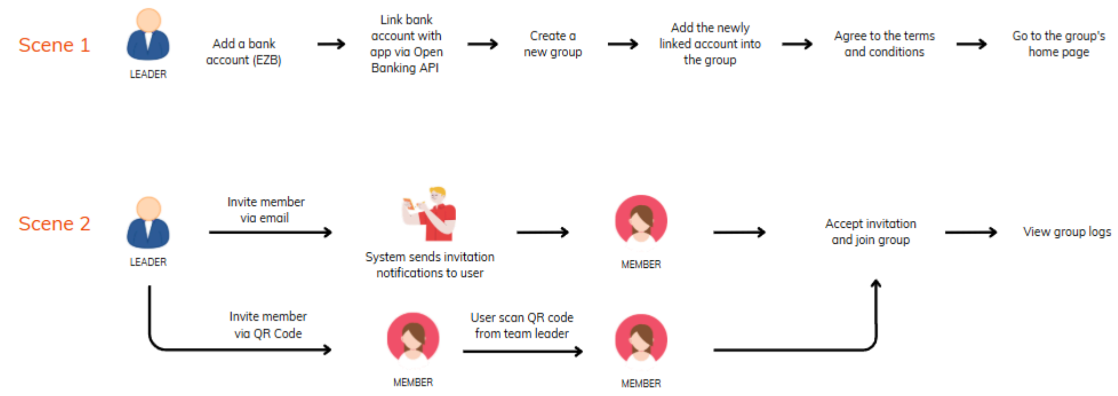


Figure 43 – Group Transaction Management

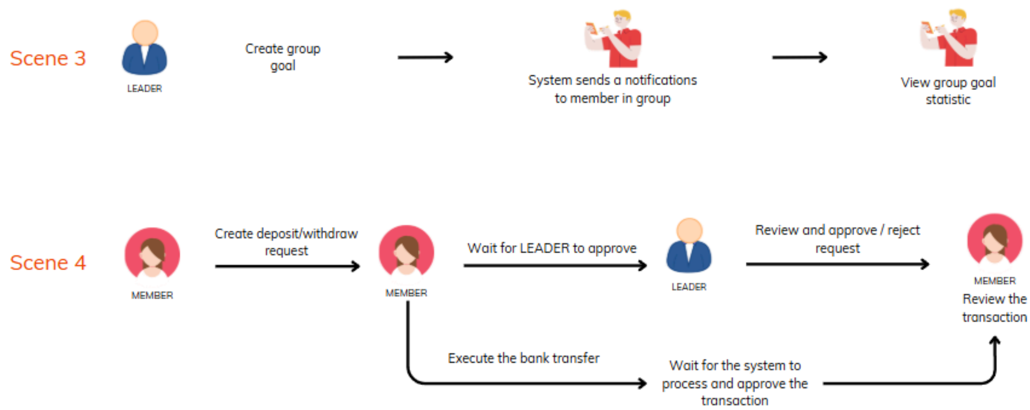
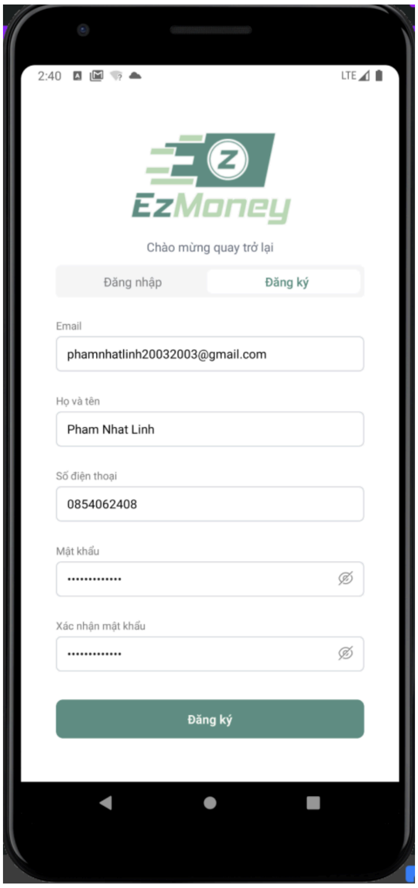


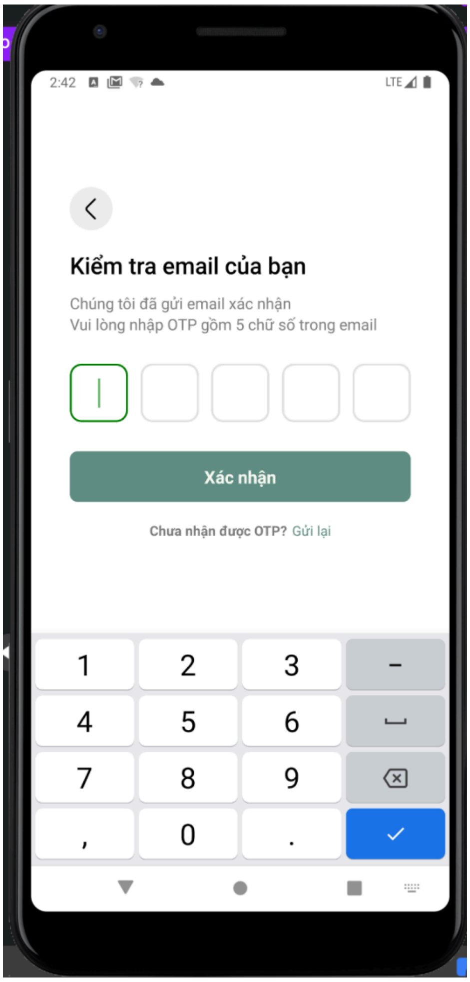
Figure 44 – Group Transaction Management

3.2 Choose Preferred Spending Model

3.2.1 Customer register account

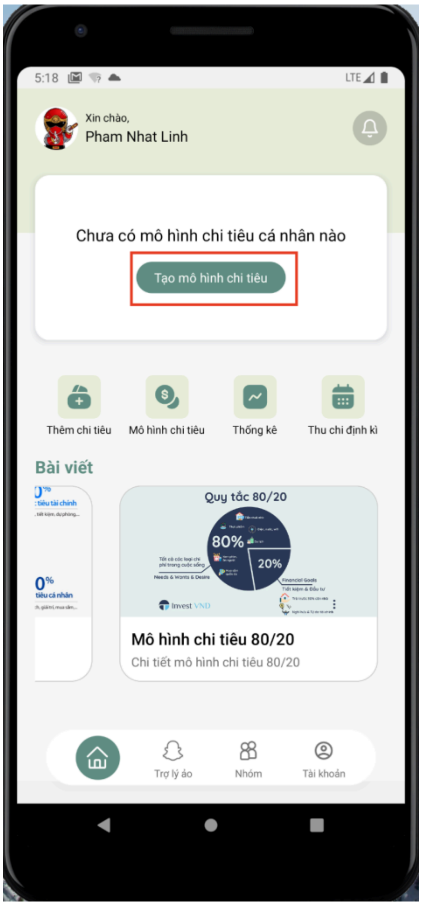


1. Touch on [Trải nghiệm ngay] button on splash screen
2. Touch on [Đăng ký] button on Login screen
3. Input valid value in all field
4. Touch on [Đăng ký] button

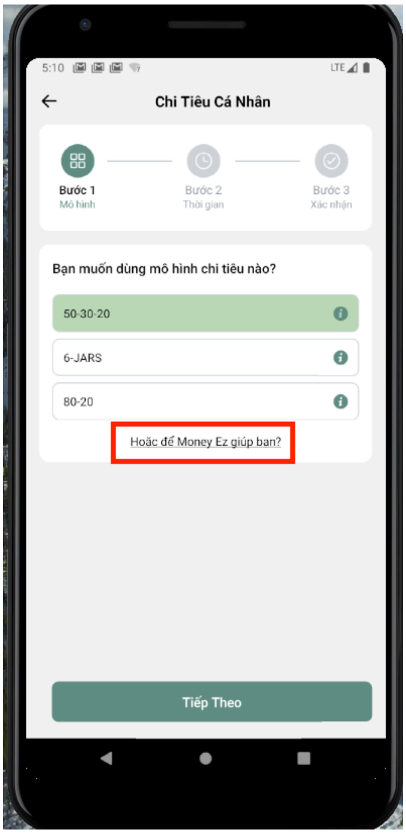


5. Input valid OTP is send to your email
6. Touch on [Xác nhận] button

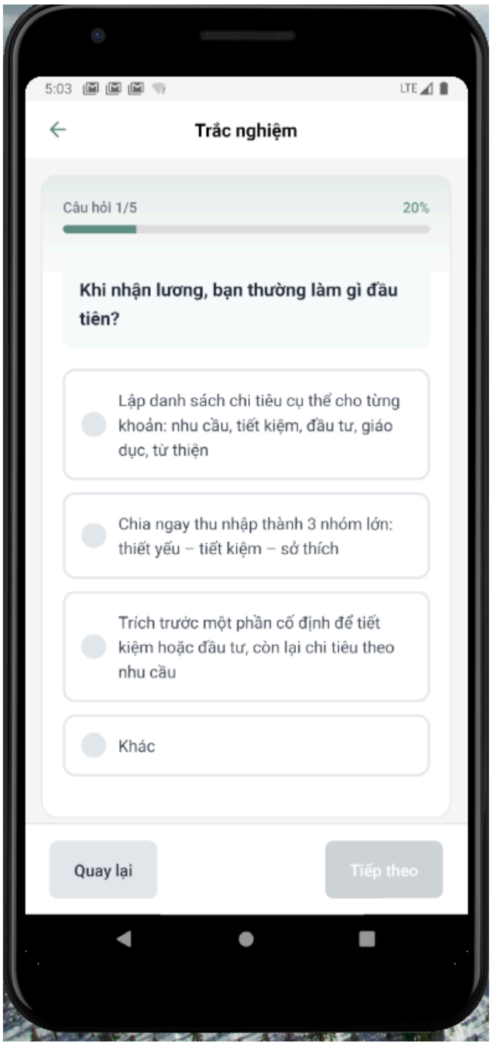
3.2.2 Choose Preferred Spending Model

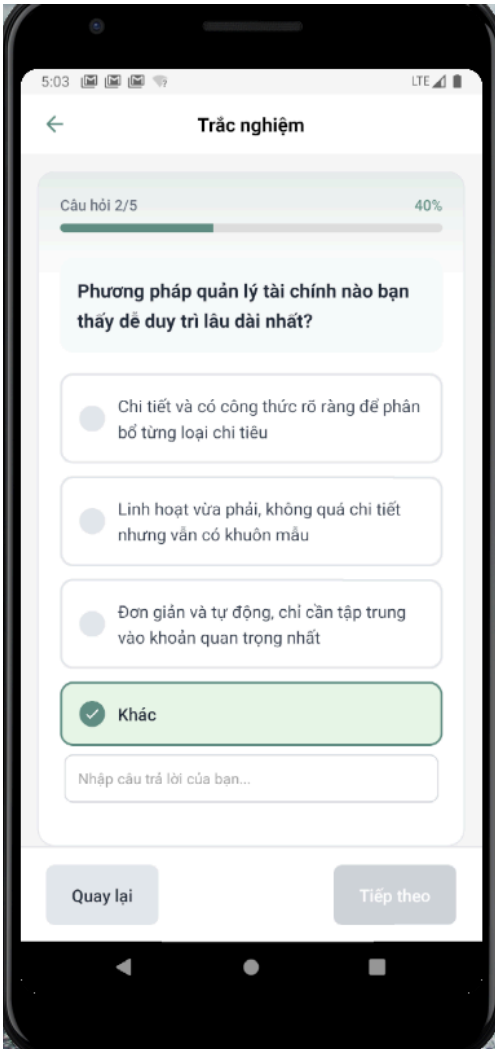


1. Touch on [Tạo mô hình chi tiêu] on main screen

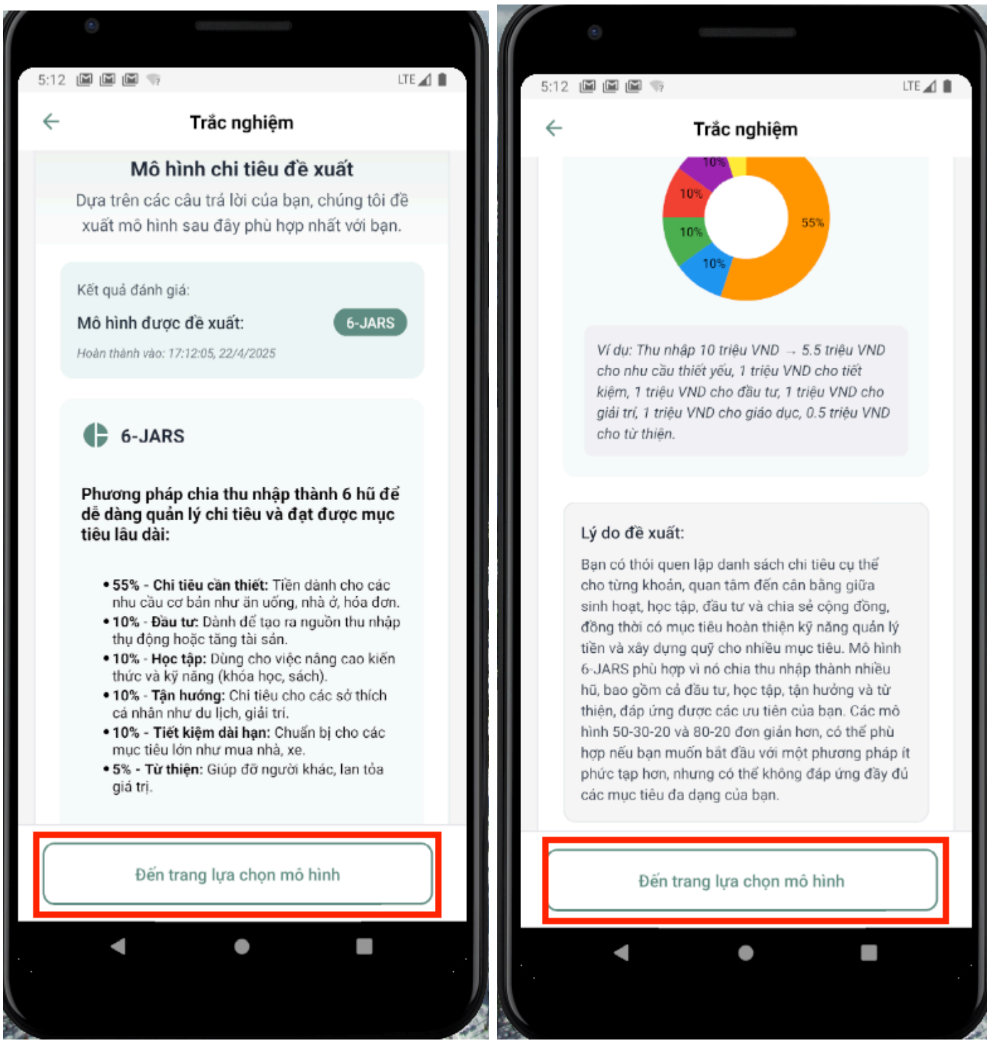


2. If you're not sure which spending model suits you best, take the quiz to discover the recommended model from MonyEZ, touch on [Hoặc để MoneyEz giúp bạn].

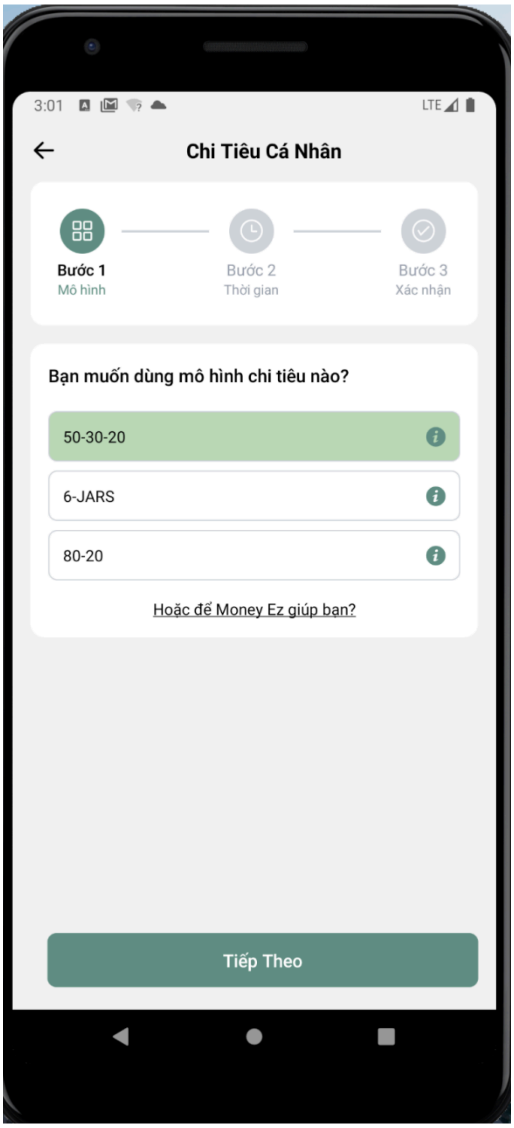




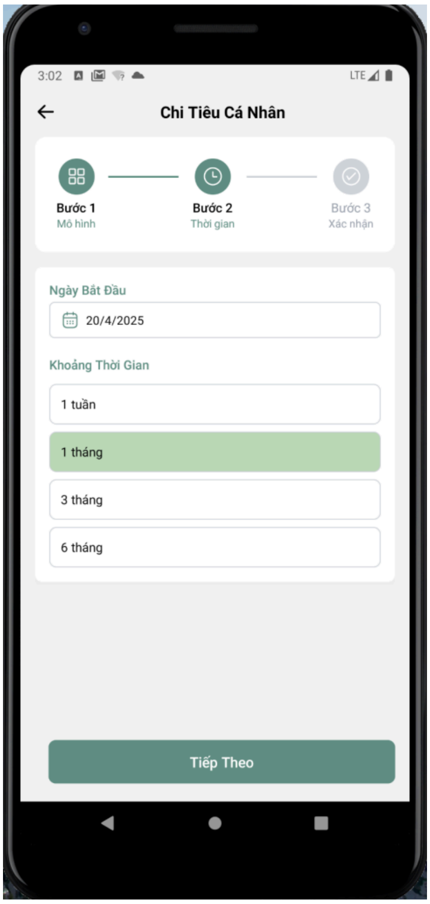
3. Continue to answer all the question.



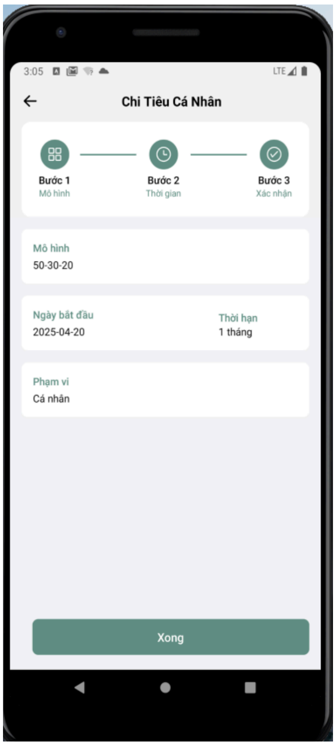
4. MoneyEZ will suggest the most suitable model for you.



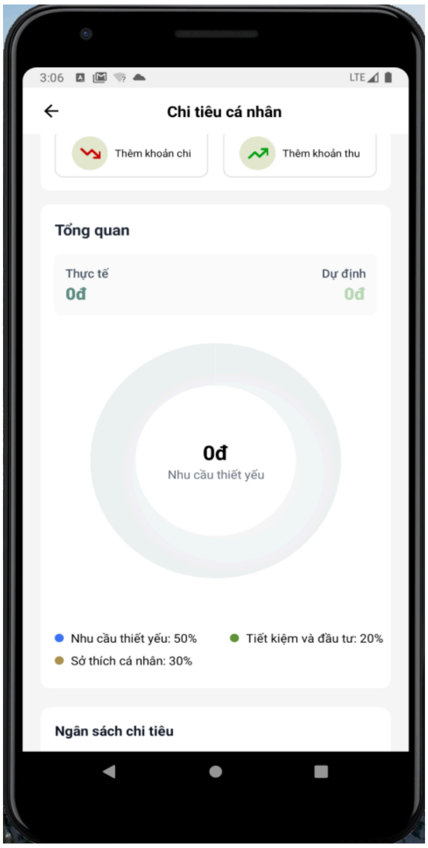
5. Select Spending model on [Chi tiêu cá nhân] screen



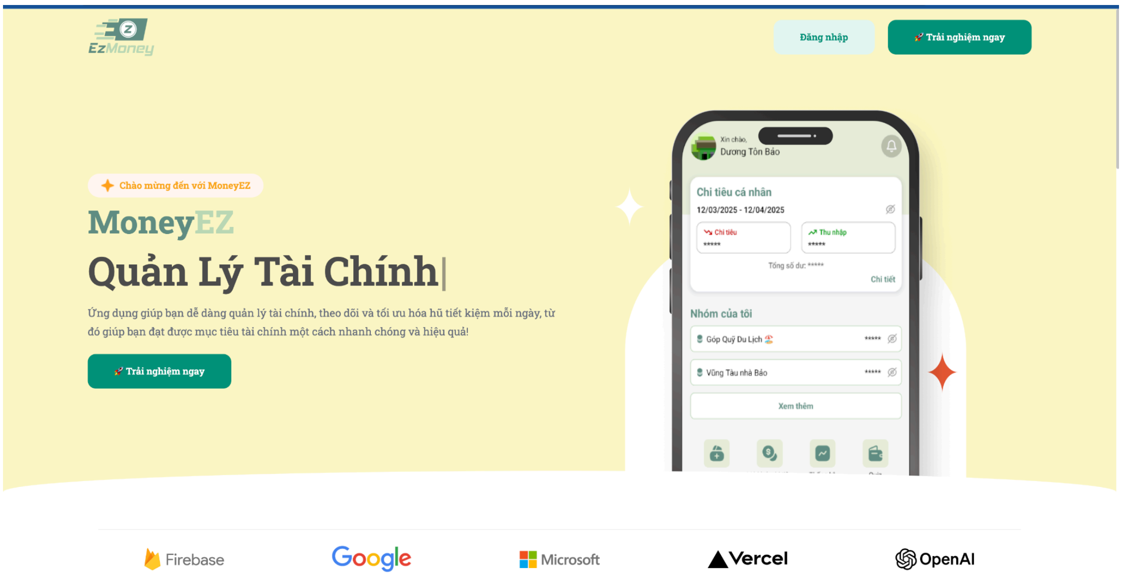
6. Select period time and start date with valid value



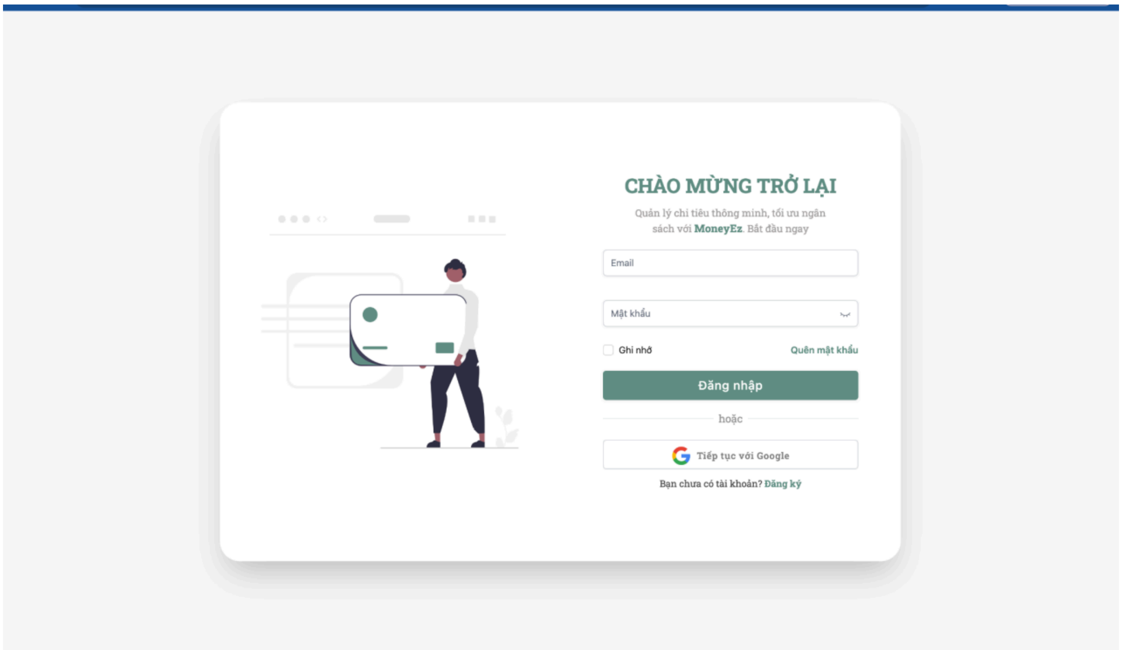
7. Touch on [Xong] button



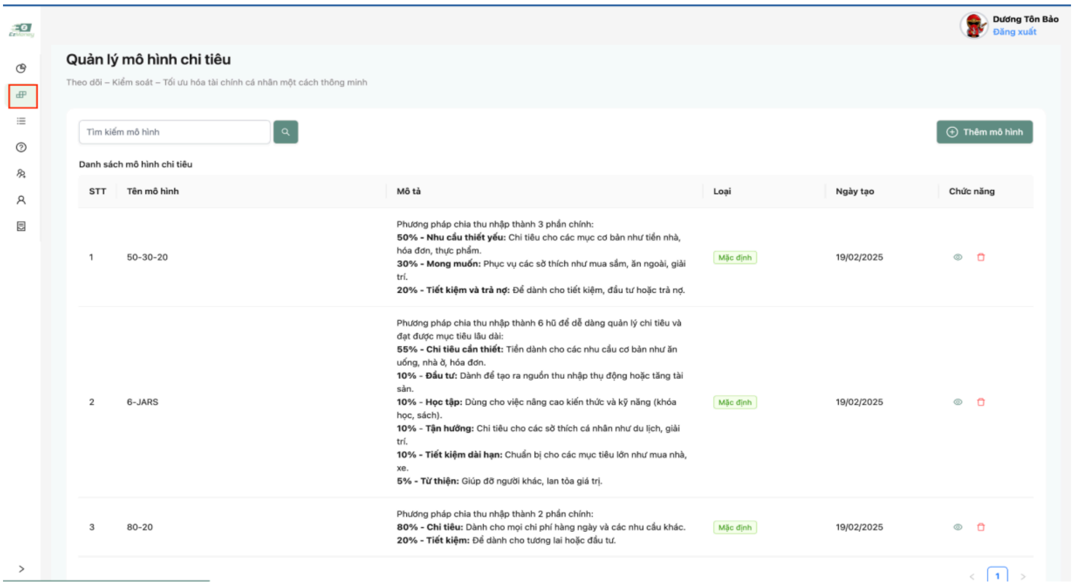
8. Personal expense statistic shown



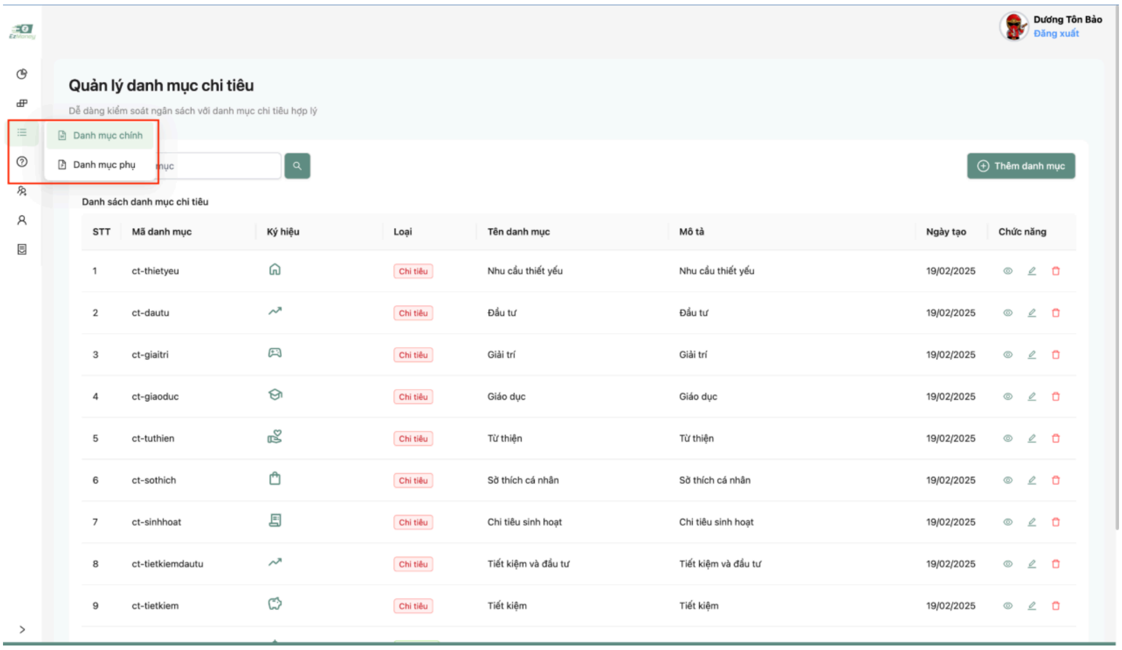
9. Continue to Web with administrator role



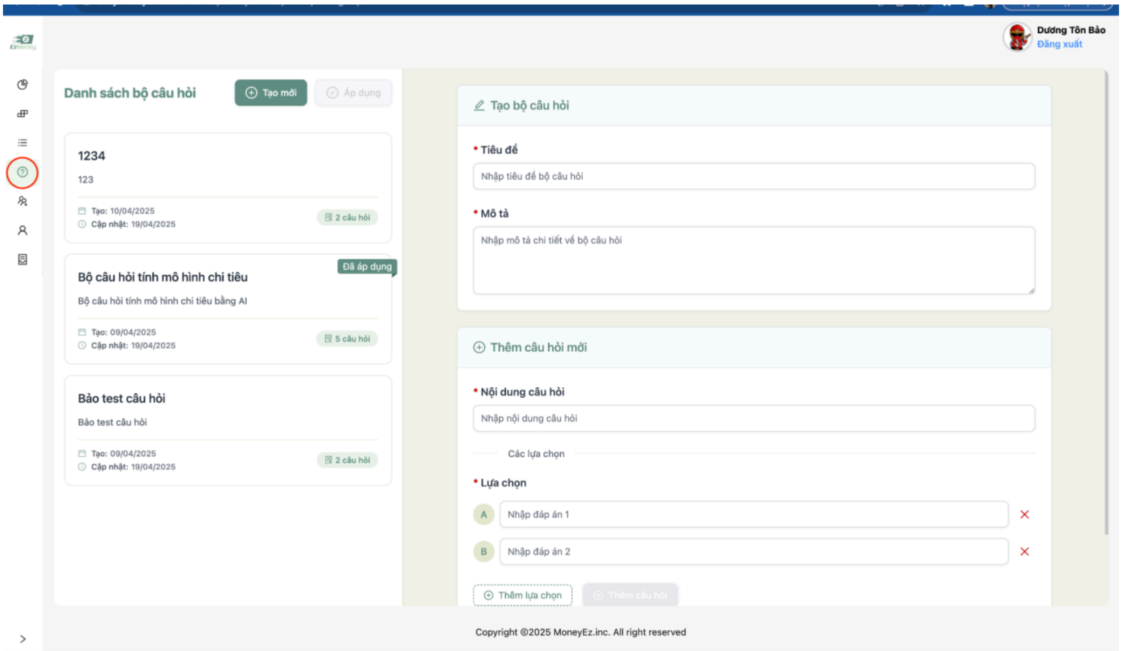
10. Log in by click [Đăng nhập]



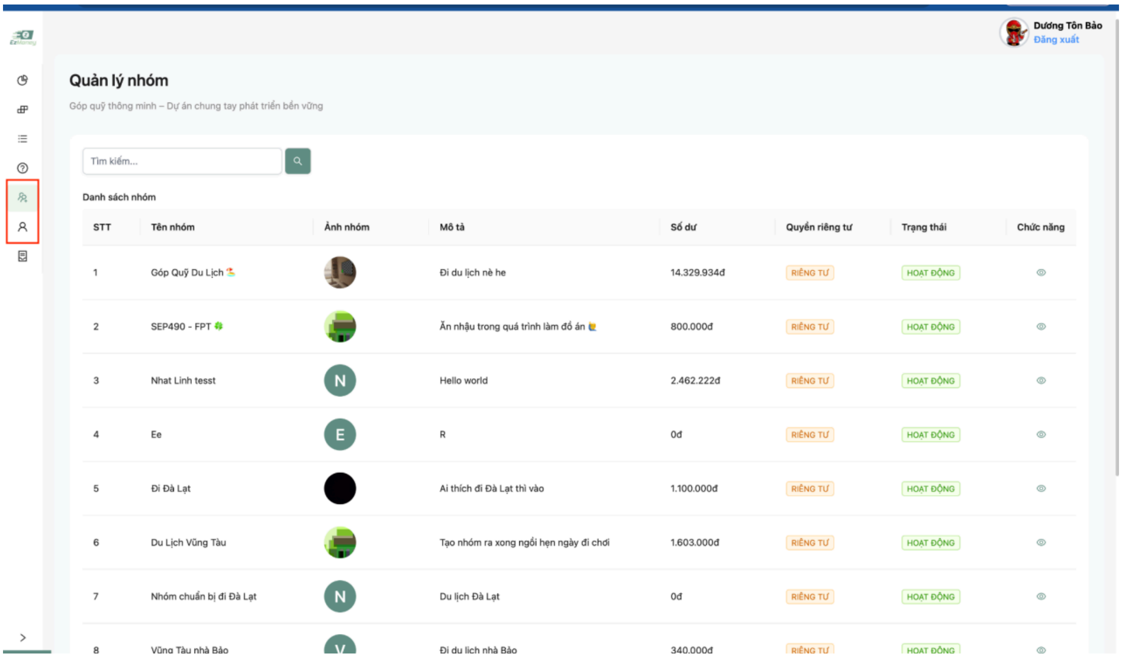
11. The statistics for whole app will appear, click on the highlight button to config spending model at highest level



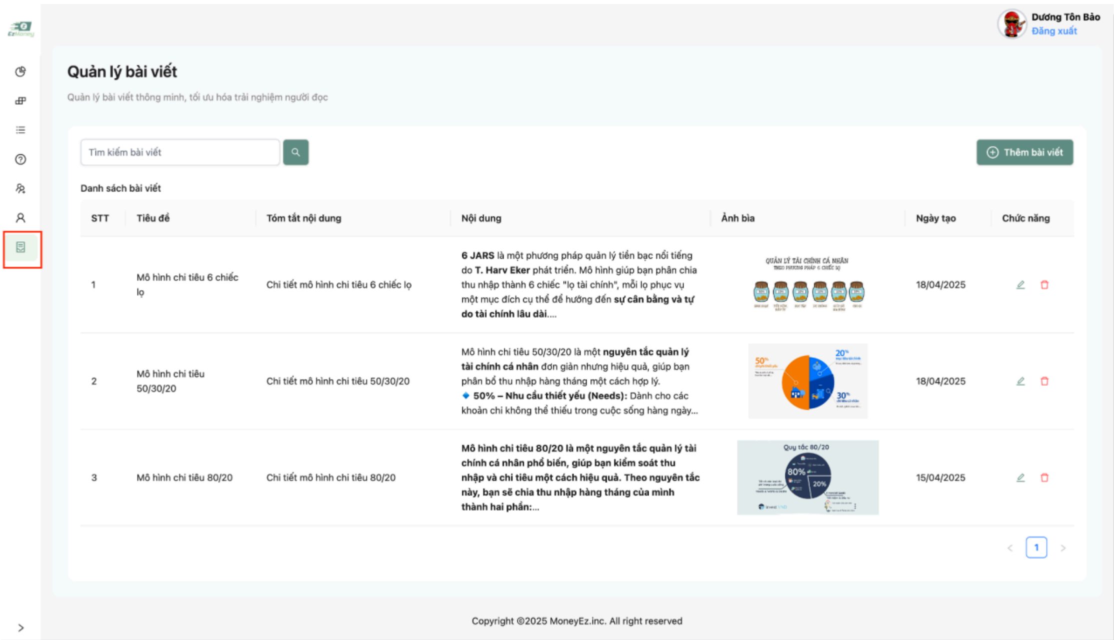
12. Click on the highlight button to config the category and subcategory at highest level



13. Click on the highlight button to modify quizzes to establish models at highest level

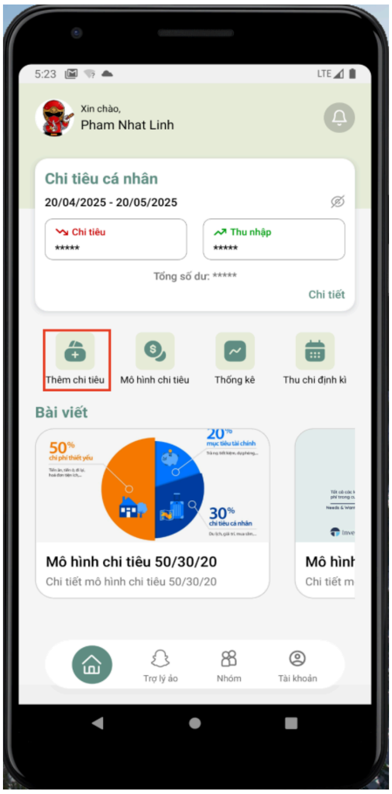


14. Click on the highlight button to config the user and group at highest level

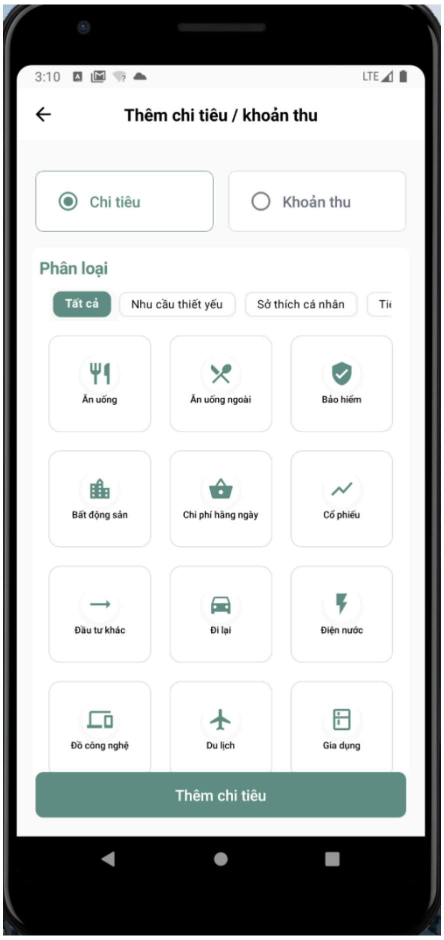


15. Click on the highlight button to config post at highest level

3.3 Personal Transaction Management - AI Drived

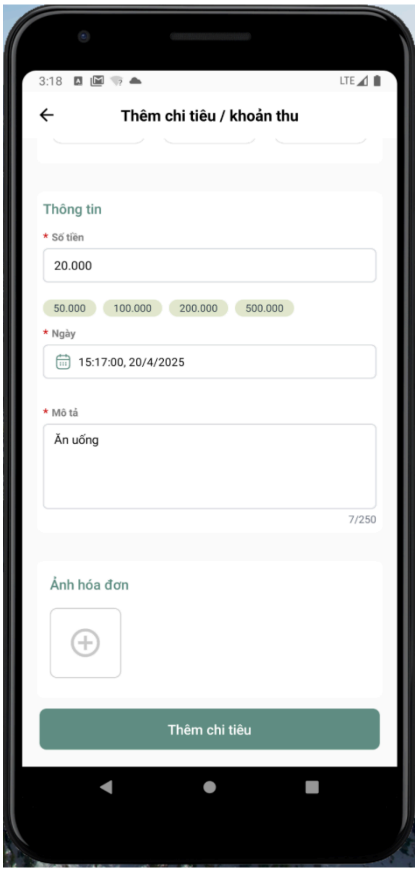


1. Create transaction [Chi Tiêu]

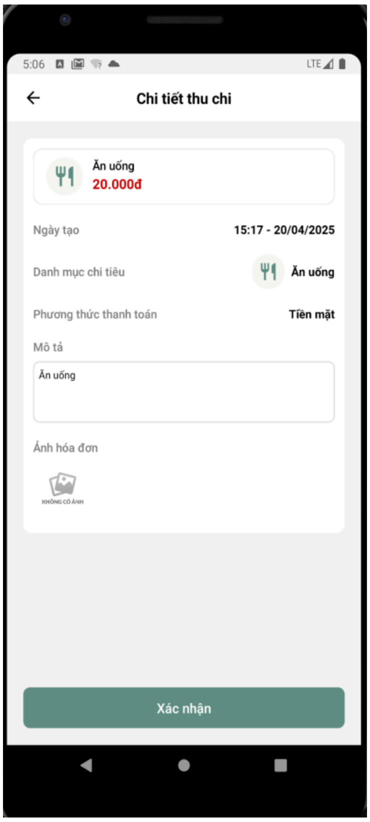


2. Input validate value in [Thông tin] and touch on [Thêm chi tiêu]

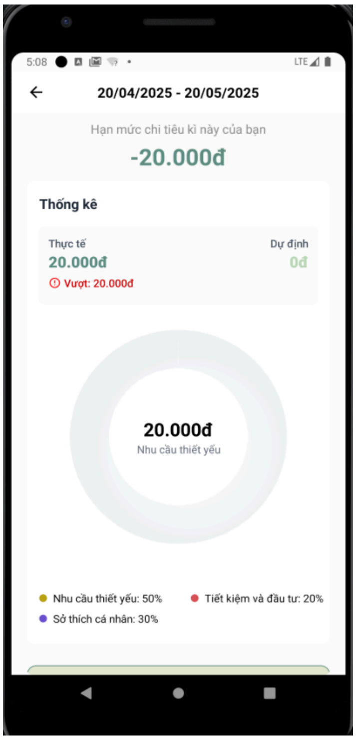
Pre-condition: Date required input today



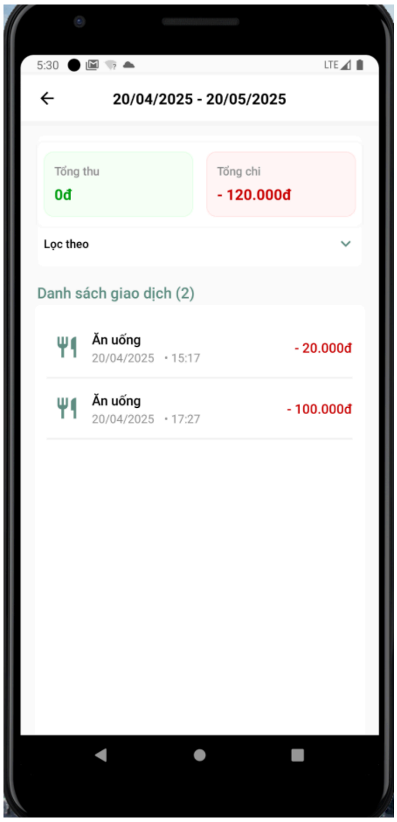
3. Confirm with the [Xác nhận] button



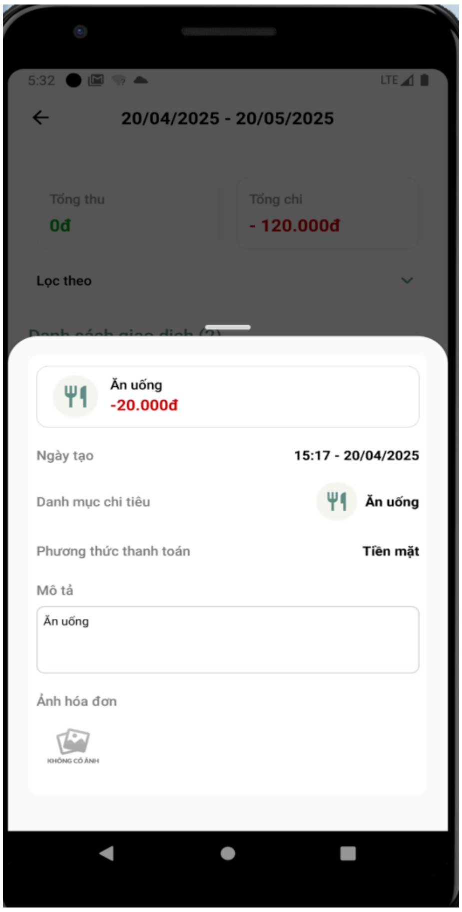
4. View user spending model report after created transaction



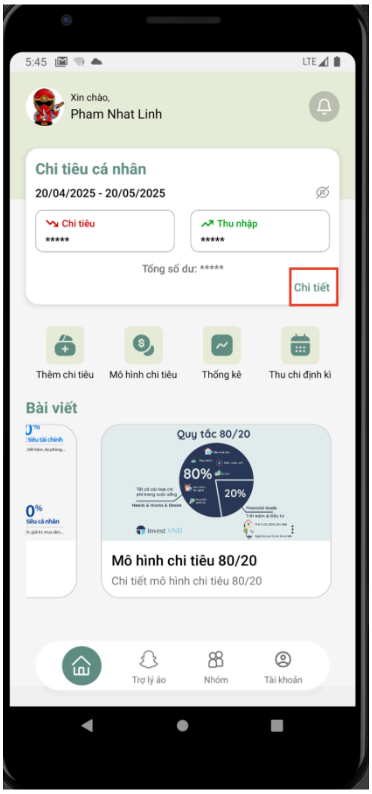
5. View all transactions



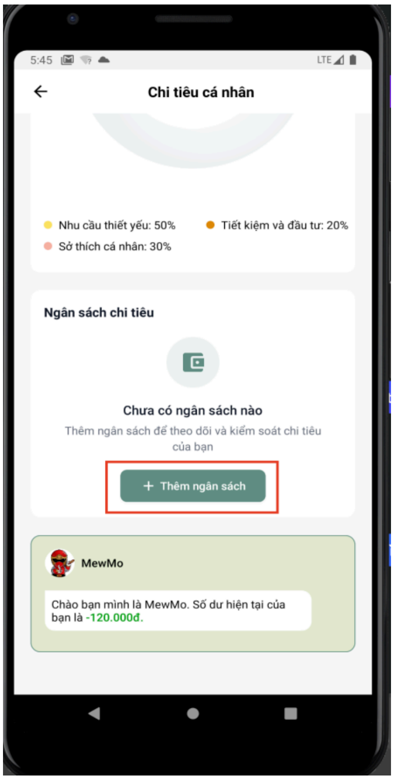
6. View detail transaction



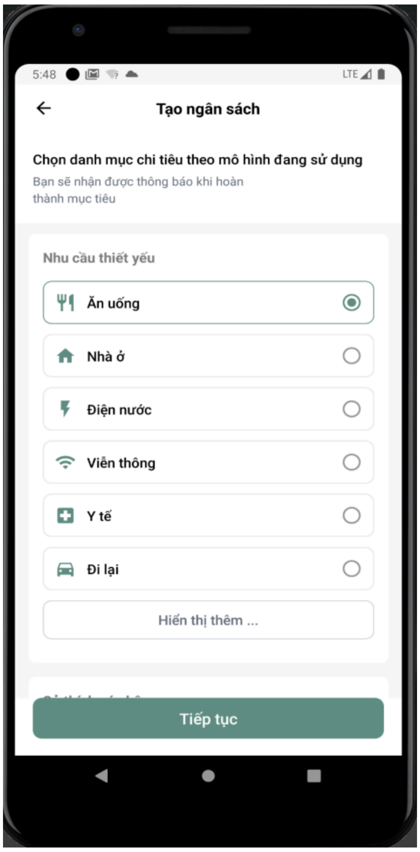
3.4 Smart Goal Tracking - AI Drived



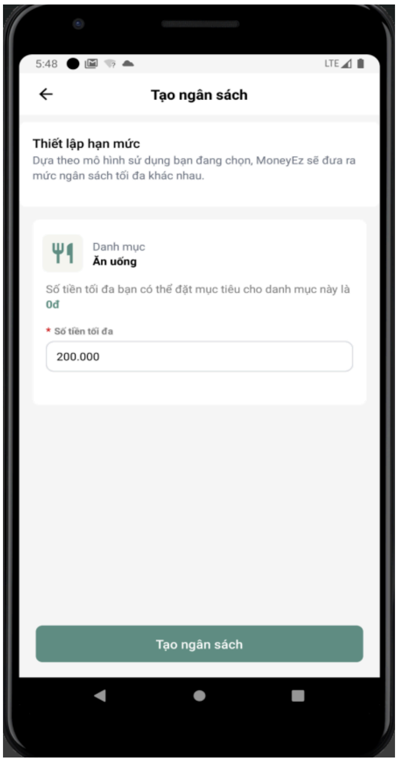
1. Touch on [Chi tiết] on the home screen



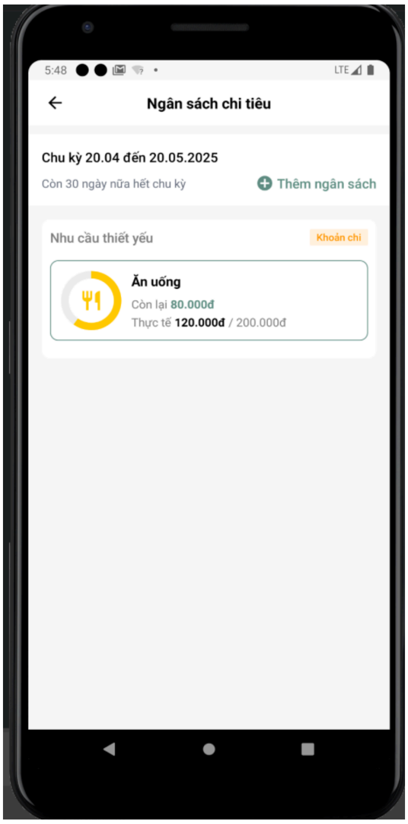
2. Touch on [Thêm ngân sách] to set up goal



3. Select each category want to deeper configuration



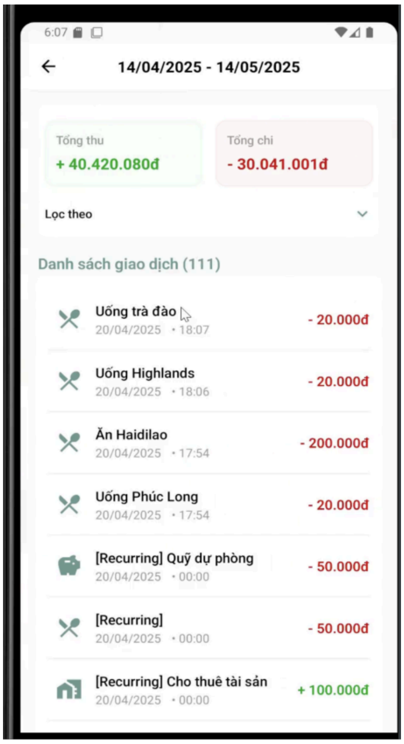
4. Continute with that limit for each category



5. See the category's limit after configuration step

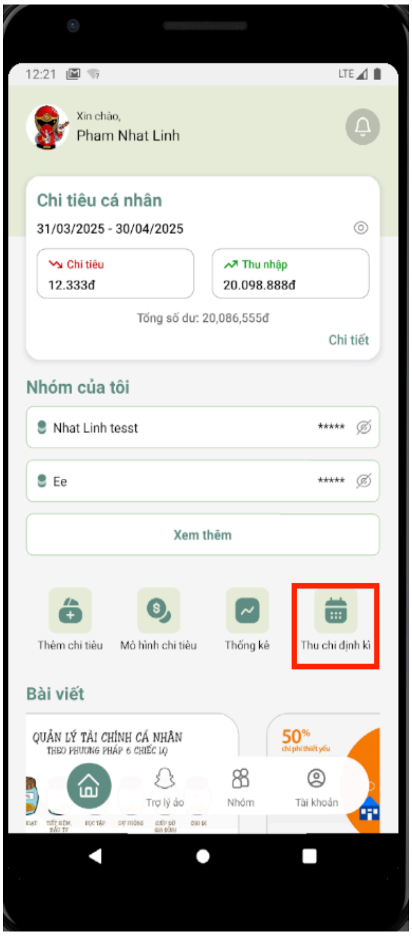
6. **(Quick expense entry using AI)**

7. Try to promt something to emphasise your transaction (eg: Uong tra dao 20k)

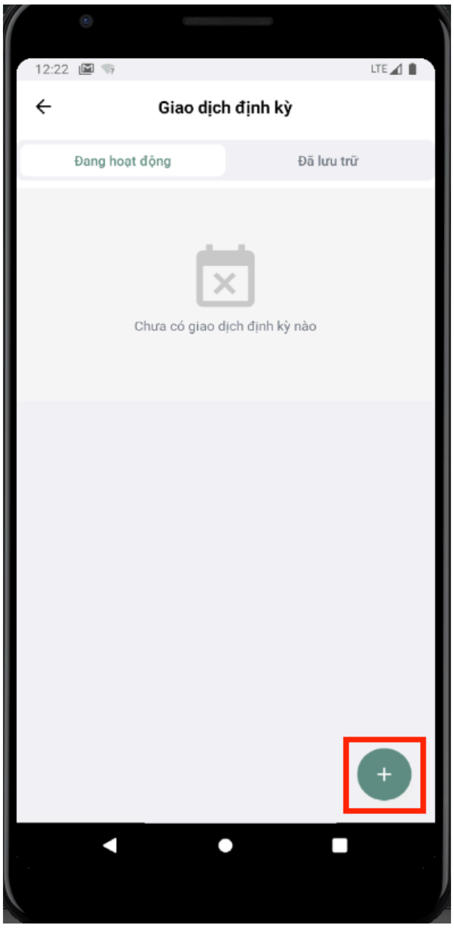


8. The expense will shown in your history transaction

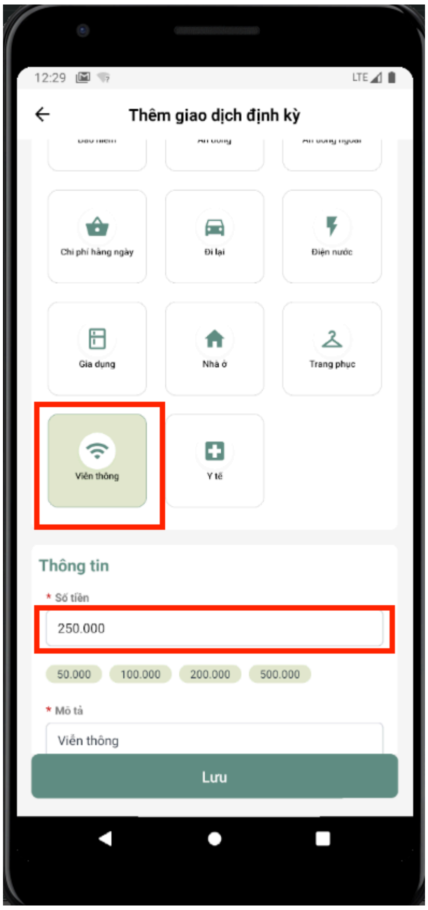
9. In other hand, you can interact with MeoMo AI to find out your financial health (eg: Khoản thu trong tháng này, Từ thời gian nào vậy,..)

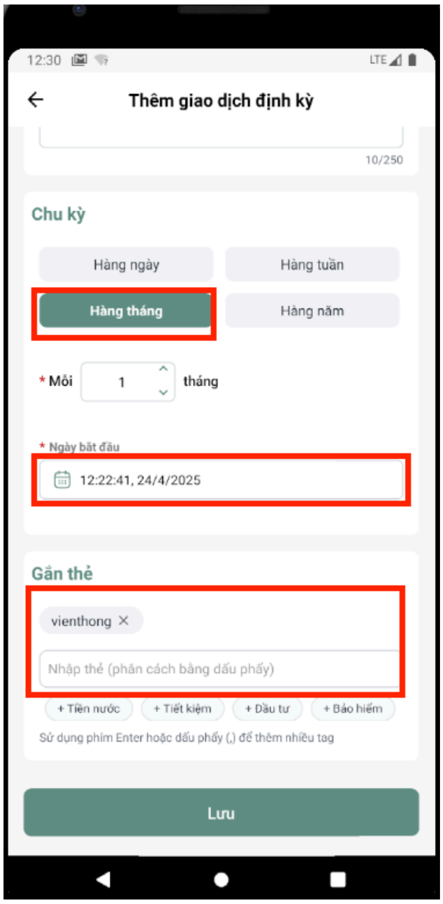


10. If you want to set up a recurring expense such as internet fees or tuition fees, them touch on [thu chi định kỳ]

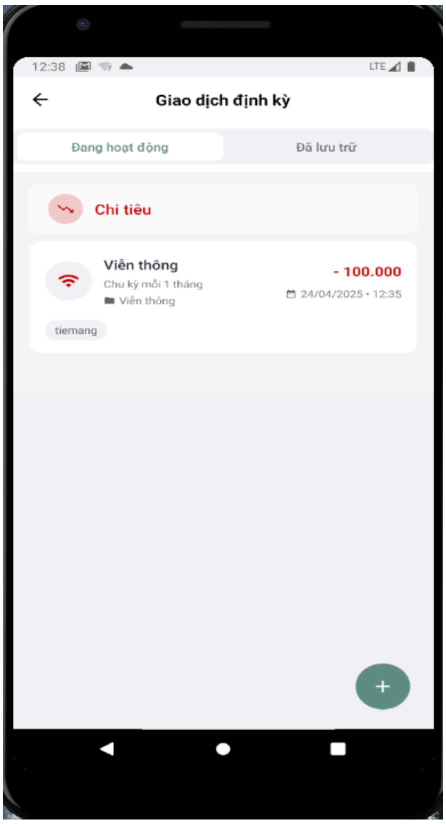


11. Touch on [+] to create recurring transaction





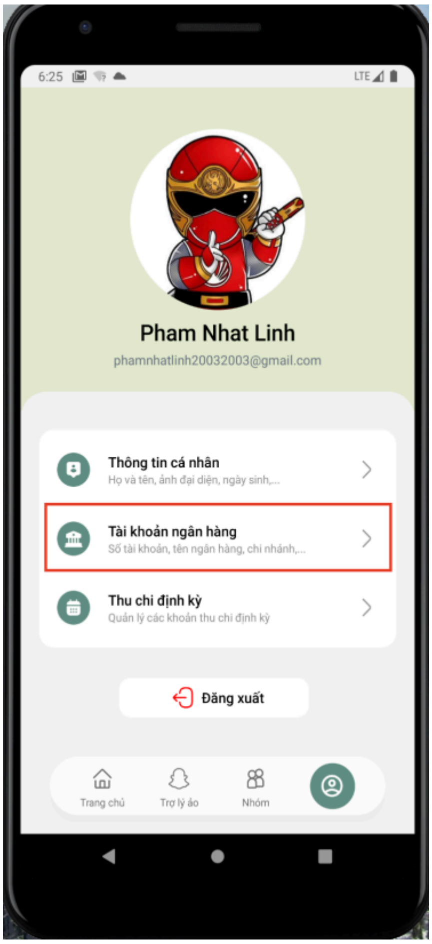
12. Fill in all the information about that recurring expense and validate it.



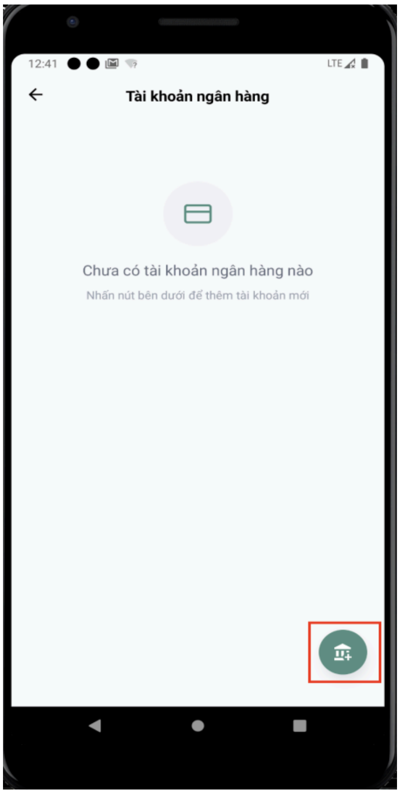
13. Your recurring transaction created.

3.4 Group Transaction Management

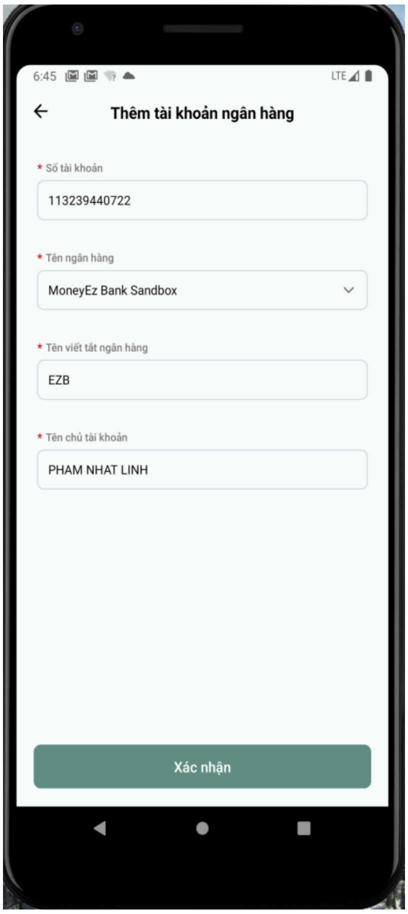
3.4.1 Initialize the group and link the bank account.



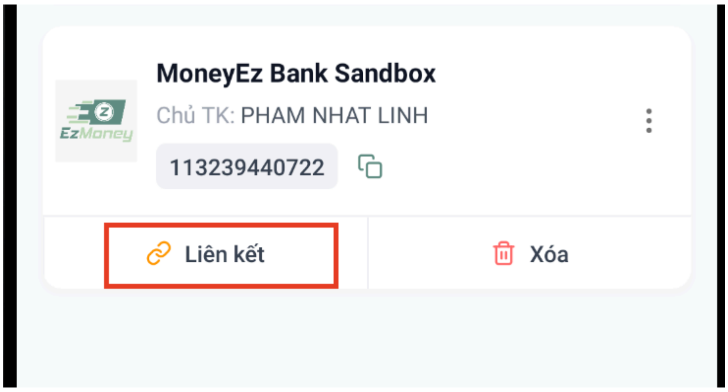
1. As a group leader, first you create a bank account to access all transactions



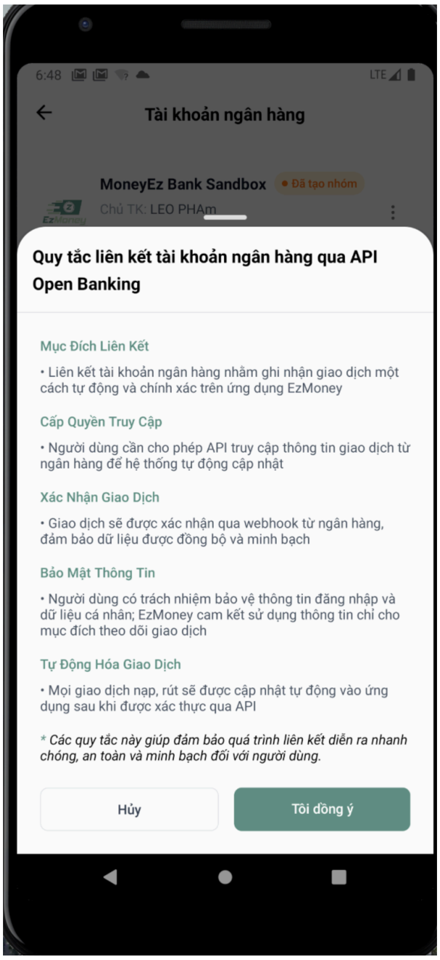
2. Touch on the right bottom corner to connect to bank account



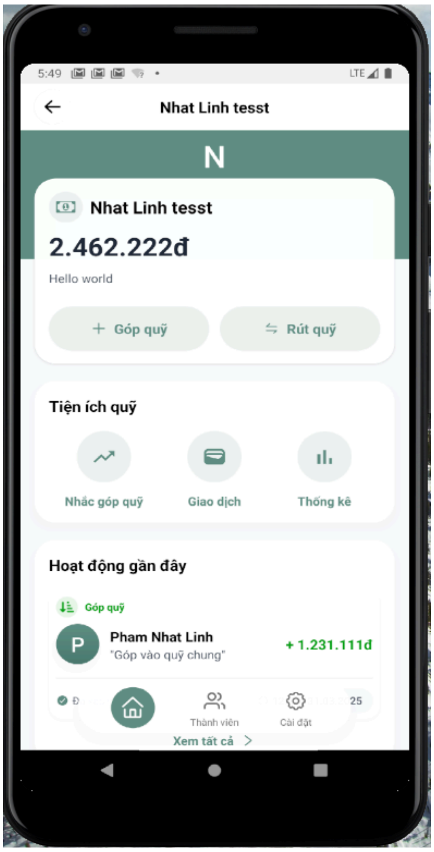
3. Fill in an account's information.



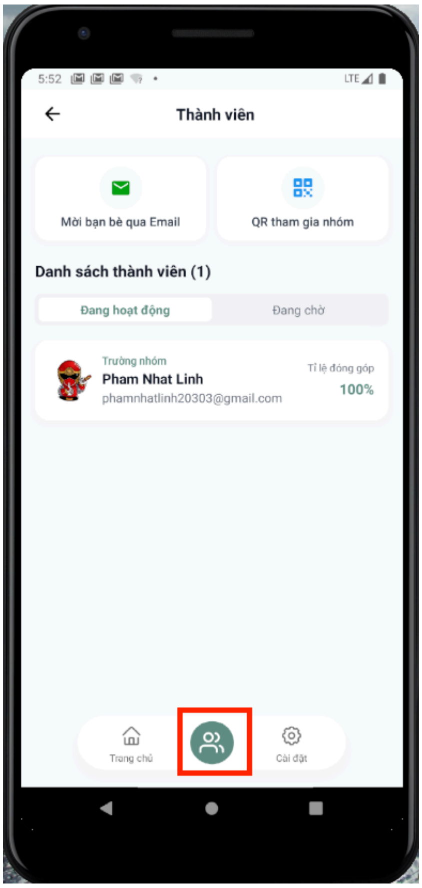
4. Connect a bank account by touch on the highlight button.



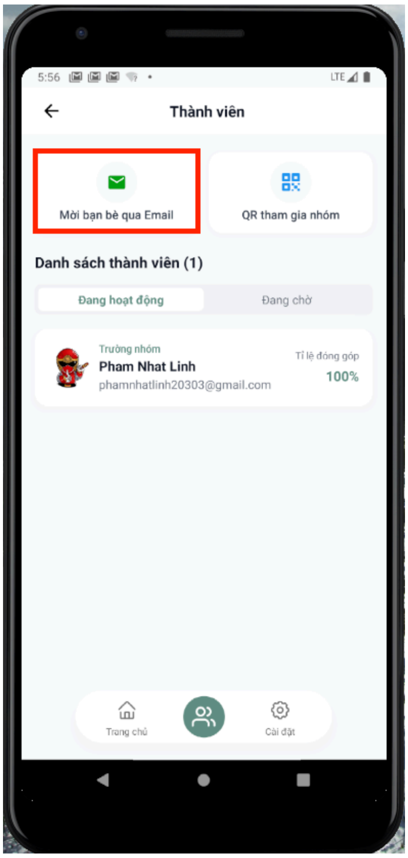
5. Please read and agree to the following terms before linking your bank account to the group, then touch on [Tôi đồng ý].



6. *Group home page*

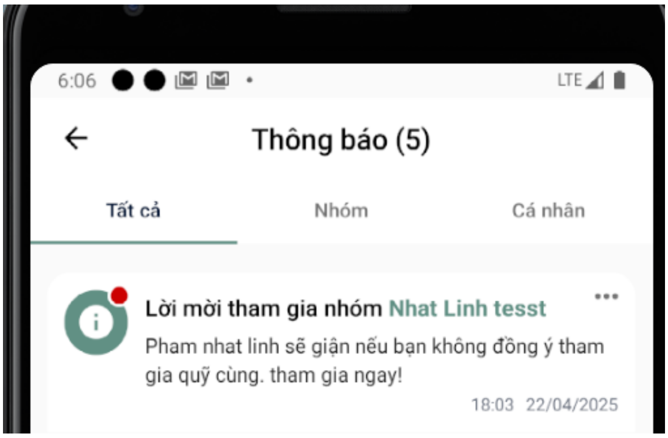


7. On the [Thành viên] tab, show all current members in the group.

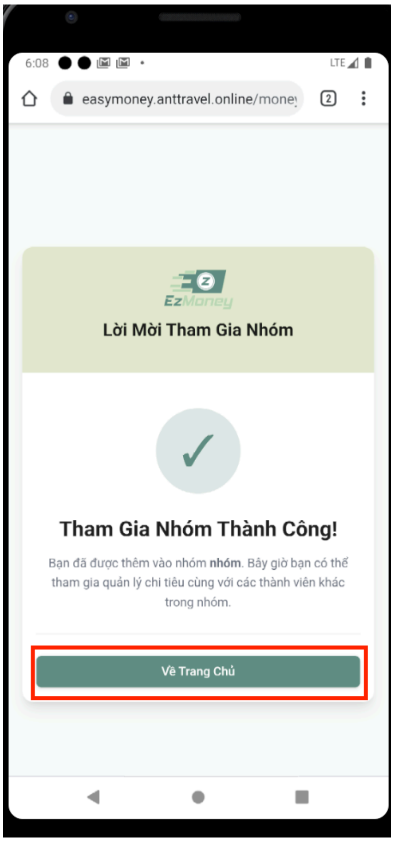


8. To add members, please touch on the button [Mời bạn bè qua email] or [QR tham gia nhóm

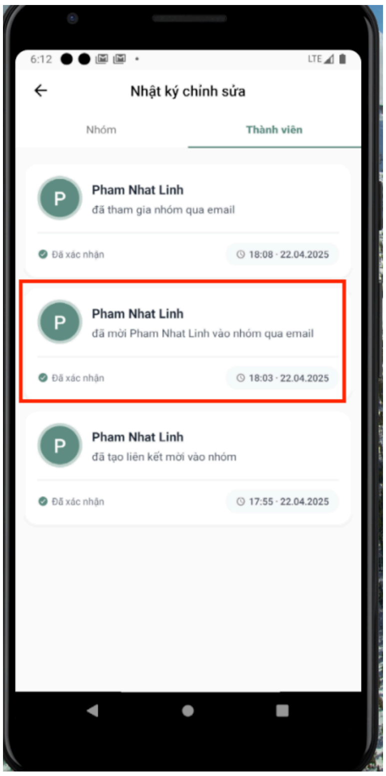
9. Fill in the search bar your friend's email to create a invitation to he/she.

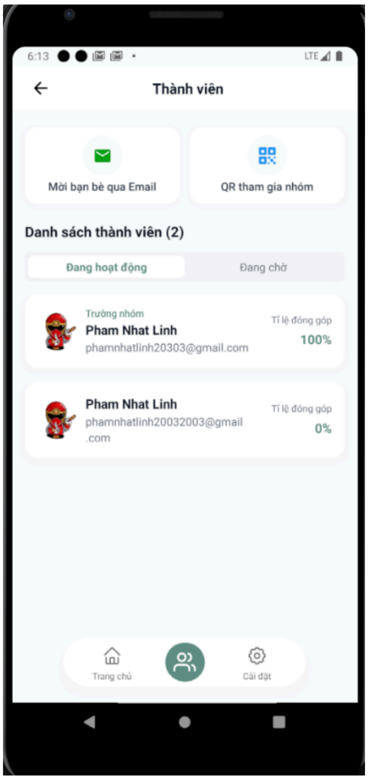


10. The invited person will see the invitation through a notification.

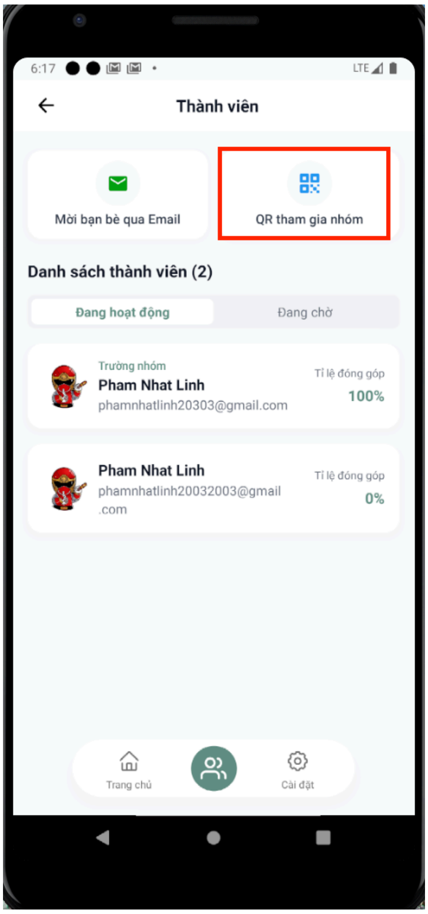


11. They'll be redirected to a webpage where they can join the group.



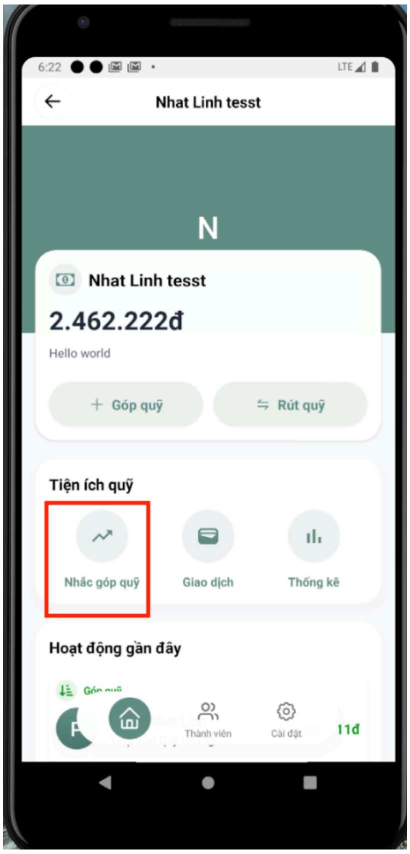


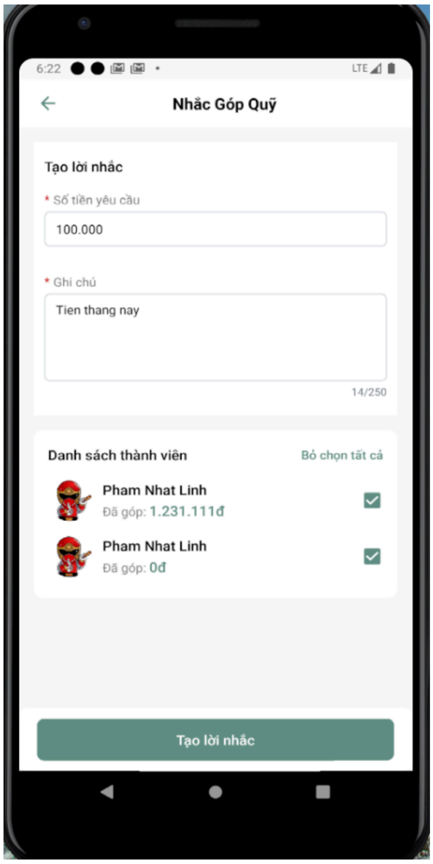
12. Now, your group have 2 members



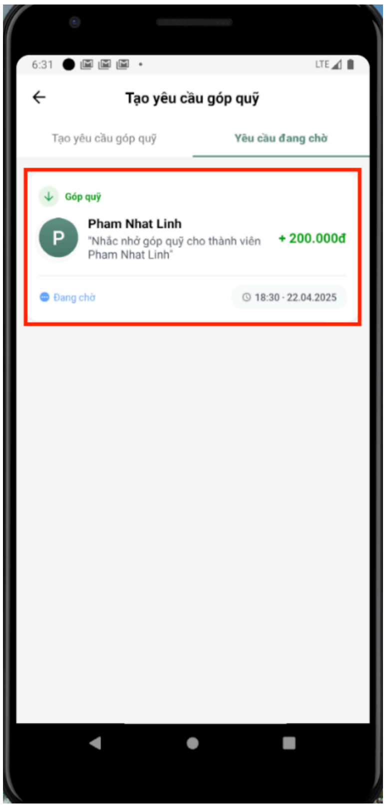
13. Or you can create a QR invitation by touch on [QR Tham gia nhóm]

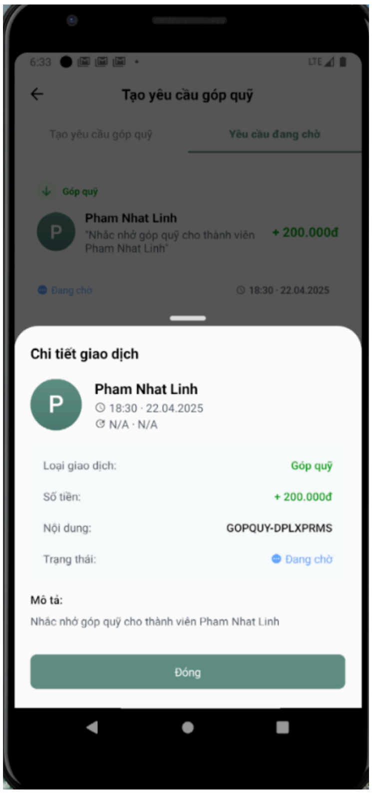
14. Scan the code to join the group.



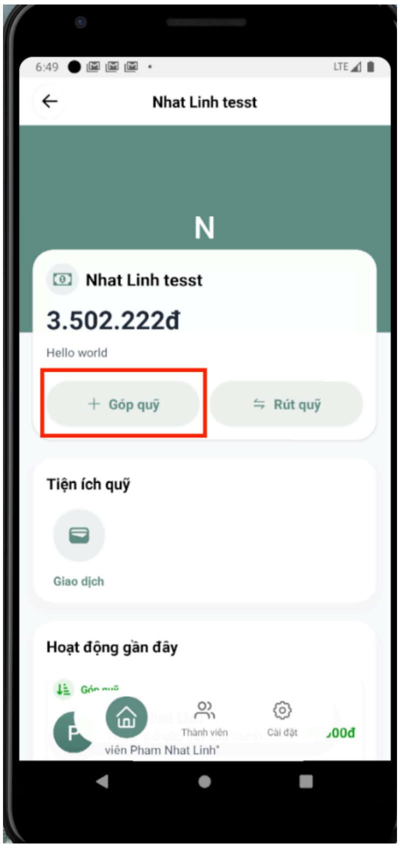


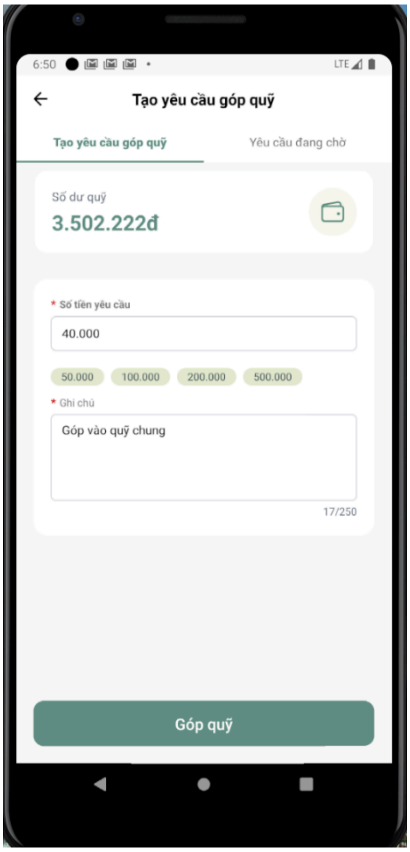
15. Touch on [nhắc góp quỹ] to Remind members to contribute to the fund.



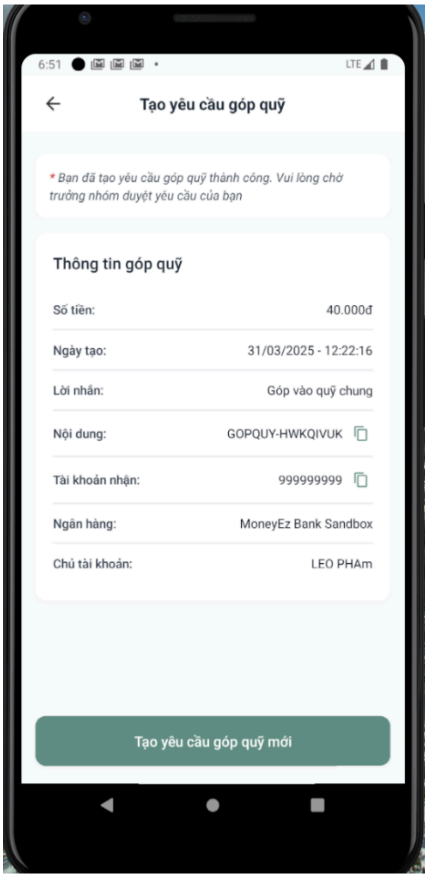


16. A fund contribution reminder will be sent to all members.

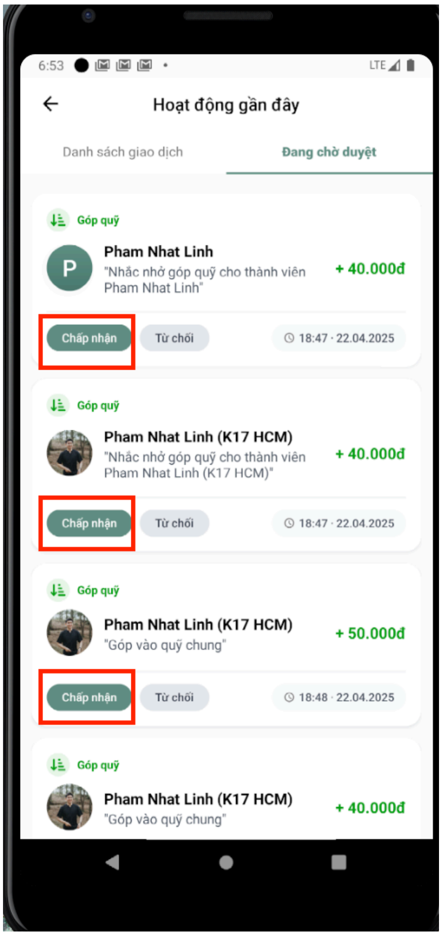




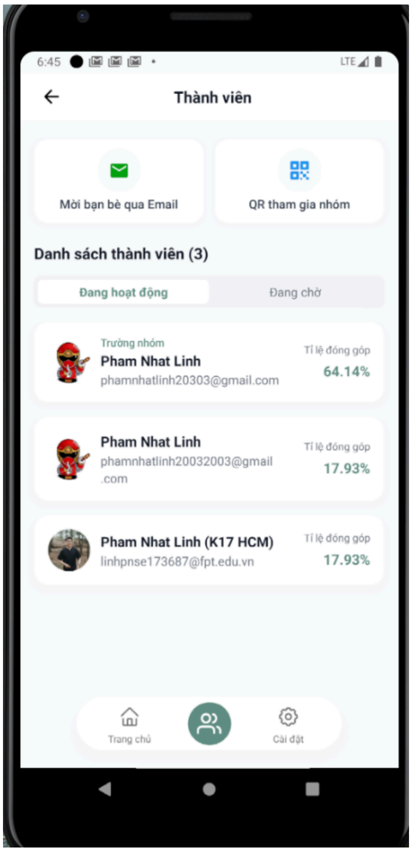
17. *Members* will contribute to the fund based on the reminder.

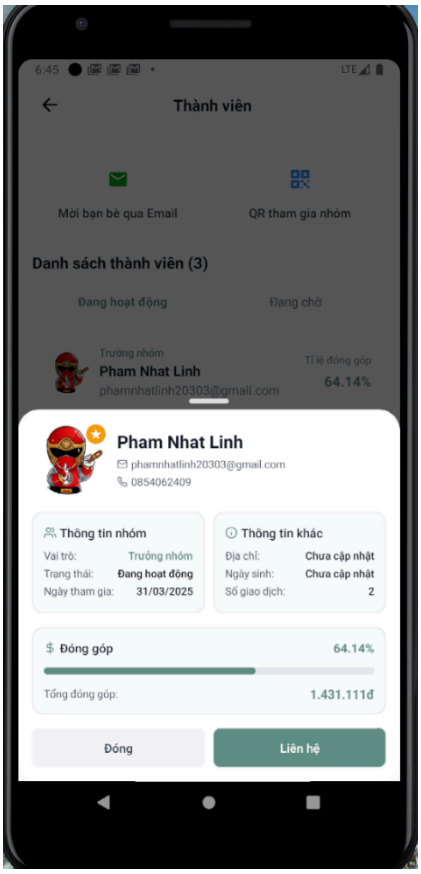


18. *Leader* will approve the request.



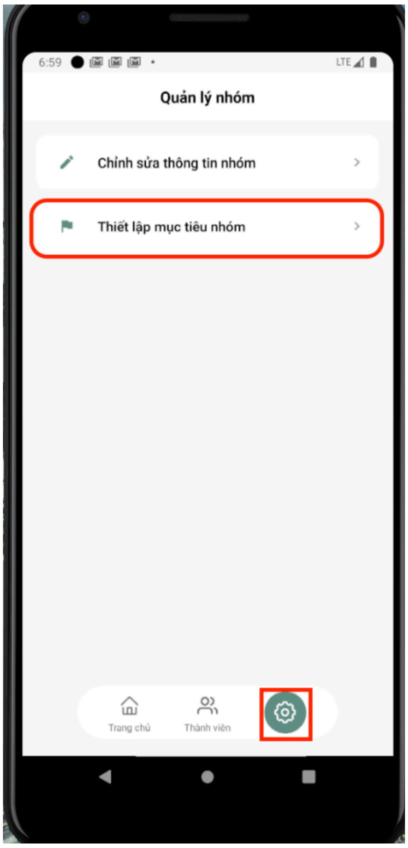
19. The contribution request will be sent to the group leader.





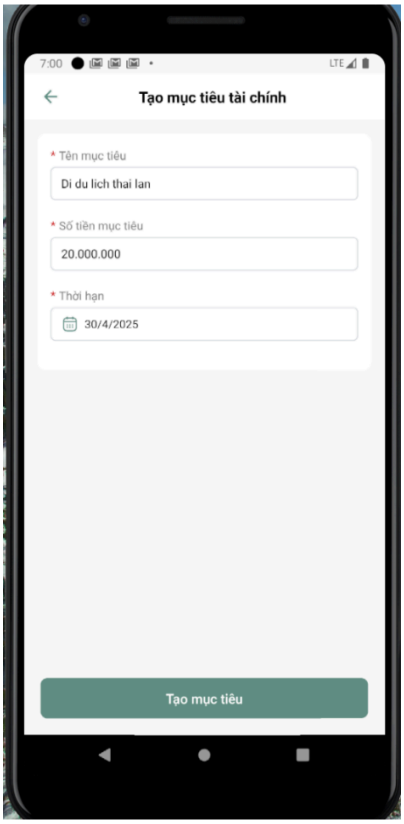
20. Once members have contributed, the group leader can view each member's contribution.

3.4.2 Set the team's common goals.

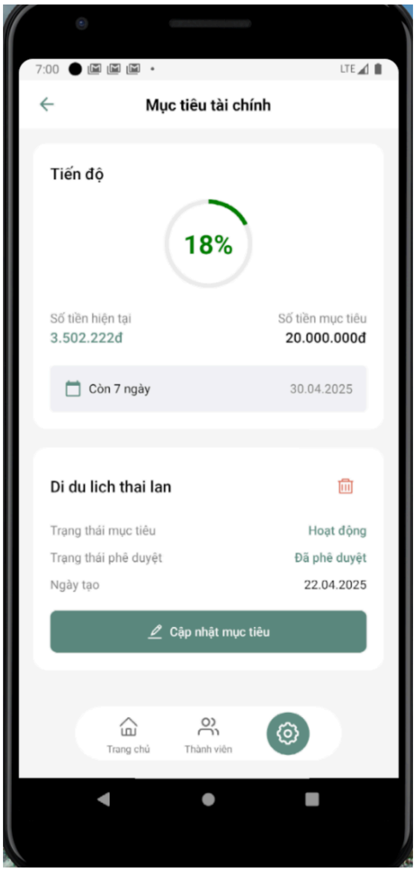


1. The leader will be the one to set the common goals for the whole team.

2. Create a new goal touch on [Tạo mục tiêu].



3. Fill in the input with valid value.



4. Monitor the progress of the common goals.