Information system for finance handling and budget planning for students

(Project for Systems III)

Nikola Gostovikj

Definition of the problem

In today’s fast-paced and rising economy, university students face massive obstacles in managing their personal finances. This issue becomes more relevant among domestic and especially international students at University of Primorska. Income is limited, prior experience of finance handling is not seen quite often, students struggle to maintain the financial equilibrium which is the golden goal that every student strives for.

The problem is seen and experienced daily, students often lose track of the amount they spent on essential items, food and goods, rent or alcohol and other leisure activities. Many students fail to plan ahead for recurring costs like monthly rent, leading to situations where they may run out of funds mid-month or accumulate unexpected debt. Students are in need of a system, which doesn’t lack context and a clear display, or fail to account for rising prices in supermarkets like Lidl or Mercator.

Consequences of poor financial planning affect every aspect of a student’s life. Students may feel stress, their inability to cover the rest of the month may affect their academic success. The family of the students, often need to provide unexpected financial support and one poor ability of budgeting from one student may have a Domino Effect on his surroundings. University support services and student advisors can feel the burden of students in financial distress, requiring more time to provide aid or solutions.

The scope of solving this problem lies in creating a simple and student oriented application which will provide real time tracking and analysis of one’s finances. Users should be allowed to enter their input and categorize their expenses while automatically integrating expenses. The goal is to improve financial literacy, develop better saving habits and reduce financial anxiety.

Through the Systems III Seminar, my goal is to develop a system that appropriately addresses these issues and design a solution that bridges the gap between financial responsibility and student life by offering an intelligent and contextual solution for budget management.

Functional and nonfunctional requirements of new system

Functional requirements

The system should enable the following functionalities:

1. Monthly Income and Expense Management
    - Users can input and edit their monthly income and categorize expenses (e.g. food, rent, transport).
    - The system tracks remaining funds based on all recorded income and expenses.
    - Users can delete individual expense entries or reset the entire budget at the beginning of each month or annually.
    - The system also stores the date, time, name, supermarket name or similar and additional information about the expense.

2. Real-Time Budget Alerts and Notifications

    - The system provides alerts / warnings when the remaining budget falls below a threshold.
    - The system alerts the user if a large purchase is made while a necessary expense (e.g. rent) is still pending.
    - Users receive monthly reports and warnings for irregular spending patterns.
    - Users receive an overview at the end of the day / month / year and statistics of their finance spending.

3. Integration with the Local Environment

    - The system updates and displays current prices of commonly purchased items from Lidl, Mercator, Spar and other supermarkets, coffee prices from local coffee shops and the prices of clothes from the local department stores based on user shopping data.
    - When adding new expenses, users can select from a list of real prices fetched from the stores.
    - The user can also add a different expense that hasn't been taken into account and write a short description and comment about that expense.

4. User profiles and Roles

    - The system should authenticate valid University of Primorska students using a valid university email.
    - Each user profile includes their name, email, income category, and spending behaviour trends which can be graphically shown to represent the amount of money spent on a daily basis.

    - The system provides personal insights based on financial habits and comparisons with similar users. (Similar users are the users which can be close friends or relatives, and the relationship goes both ways)

5. Goal Setting and Affirmations

- Users can set financial goals, a simple example is saving 50 euros by the end of the month.
- The app provides motivational messages, financial affirmations based on spending progress.
- The system logs progress toward each goal and visually displays the remaining amount needed.

Non-functional requirements

1. The system must handle up to 6000 (University of Primorska students) active users simultaneously without significant delays.

2. The system should generate and display full budget analysis reports for each user in under 2 seconds, including visual graphs and alerts.

3. The system must integrate with price update APIs provided by the supermarket chains such as Lidl and Hoffer and store this data daily.

4. The system handles sensitive financial and personal data, therefore it must ensure secure communication and data exchanged between users and the platform.

5. User data and financial logs must be stored in a secure manner for a minimum of 12 months. Data must be saved redundantly on at least two separate storage systems, and weekly external backups must be performed.

6. The system must be available to users 24/7 and they must be able to access the application using any modern browser on desktop or mobile with an internet connection.

7. The system must reuse modular components (e.g. charts, notification templates, category handlers) to reduce processing load.

8. The system should have Help and FAQ sections for onboarding and documentation for installation, maintenance and additional information must be provided in PDF and web formats.

9. The system should support scalability and future integration with banking APIs or student discount services (Študentska Prehrana), ensuring easy expansion of functionalities without the need for major architectural changes.

10. The system should support localization and multilingual interfaces, with Slovene and English available by default, allowing users to interact with the application in their preferred language.

Feasibility study

Financial management is a valid problem and requires a proper solution. It can be easily solvable with development of a modern and responsive application tailored for University of Primorska students. This application would allow users to track income and expenses, visualize spending patterns, receive alerts, and integrate real-time pricing from Customer-facing businesses. Looking at the problem from a technical point of view, the application will rely on well-established technologies such as React (for the frontend), Node.js and Express.js (for the backend), and MySQL or MongoDB (for data storage). The price data of customer-facing businesses can be accessed through publicly available or commercial APIs, and the integration of these services is technically straightforward with regular HTTP requests or scheduled scrapers, where APIs are not available.

The most challenging technical requirement is ensuring the efficient integration of real-time pricing data and secure financial data handling. However, existing tools and frameworks support encryption and secure authentication, making it technically feasible with current web technologies. Data analytics, graphical reports, and financial insights will be computed on the server side using standard data processing libraries and visualized through lightweight libraries like Chart.js or D3.js. The application must be in compliance with relevant European Union regulations, including the General Data Protection Regulation. All personal and financial data will be transmitted over encrypted connections, securely stored and never transferred outside the EU unless handled by GDPR-compliant services. User's data will be anonymized where appropriate, and access will be restricted to authenticated and authorized individuals only.

The system must be and is economically viable. It will be built using open-source technologies, reducing licensing and maintenance costs. The user traffic remains moderate (up to 6000 students as of 2025), monthly operating costs will remain low. The system's real-world benefits will drastically reduce student stress, improve financial literacy and potentially lowering university counselling workload.

The system is organizationally and socially acceptable. It aligns with the digital habits of today's students and integrates seamlessly into their daily lives without requiring behavioural overhauls. The system encourages responsible financial management through motivational messages and visual progress tracking. Supports multilingual use aligning with the languages spoken in the University of Primorska. The tool presents itself as a natural extension of their existing digital environment.

Logical Design

Data modelling

Matrix User role / functions

Table 1: Matrix user role/functions

Functions	Student	Administrator
Track income and expenses	Yes	No
Set and manage monthly budgets	Yes	No
Receive alerts and financial insights	Yes	No
View aggregated financial trends	Yes	Yes
Visualize personal financial reports	Yes	No
Manage system settings	No	Yes
Manage user roles and permissions	No	Yes
Generate reports on system usage	No	Yes
Set financial goals and receive affirmations	Yes	No

Data dictionary

Table2: Data dictionary

Entity	Description	Attribute	Type	Description of attribute
User	User of the system	user_id	int	Identified of the user
		name	varchar(255)	First name of the user
		last_name	varchar(255)	Last name of the user
		email	varchar(255)	University email
		password	varchar(255)	Password
		role	varchar(20)	Role of user (student, admin)
		amount	decimal(10,2)	Current amount
Expense	A financial expense entry	expense_id	int	Identifier of the expense
		user_id	int	ID of the user who made the expense
		amount	decimal(10,2)	Cost of the expense
		category	varchar(100)	Expense category(food, rent, activities)
		store_name	varchar(255)	Name of store
		description	text	Additional information
		date_time	datetime	Date and time of the expense
		store_id	int	ID of store
Income	An income entry	income_id	int	ID of income
		user_id	int	ID of user who received income
		amount	decimal(10,2)	Amount of income

		date_time	datetime	Income received
Alert	System-generated budget alerts	alert_id	int	ID of alert
		user_id	int	ID of user who received alert
		type	varchar(100)	Type of alert (threshold, pattern)
		message	text	Content of alert
		timestamp	datetime	Date and time of alert
		status	varchar(20)	Seen or not seen
Store	Information about stores	store_id	int	ID of store
		store_name	varchar(255)	Name of store
		location	varchar(255)	Location of store
		date_fetched	date	Date of the price
Item	Items	item_id	int	Unique item ID
		name	varchar(255)	Name of product
		category	varchar(255)	Food, drink, hygiene
		store_id	int	Links to store
		price	decimal(10,2)	Price in EUR
		timestamp	datetime	Last time of updated price
Goal	A financial saving goal	goal_id	int	ID of goal
		user_id	int	ID of user
		target_amount	decimal(10,2)	Target amount to be saved
		progress_amount	decimal(10,2)	How much is already saved

		deadline	date	Deadline to reach the goal

Entity relational diagram (ERD)

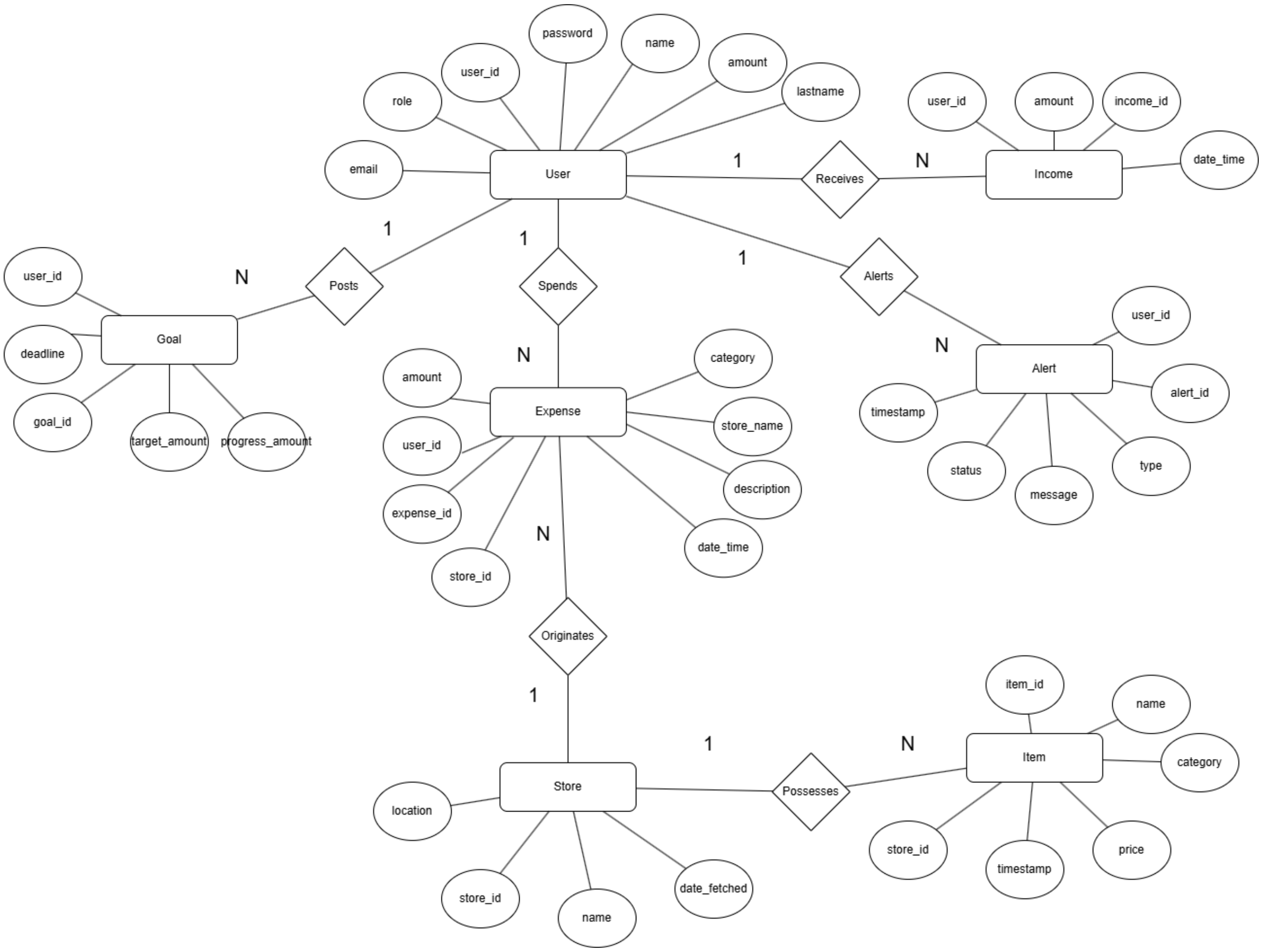

Figure 1: Entity relationship diagram

Relational model

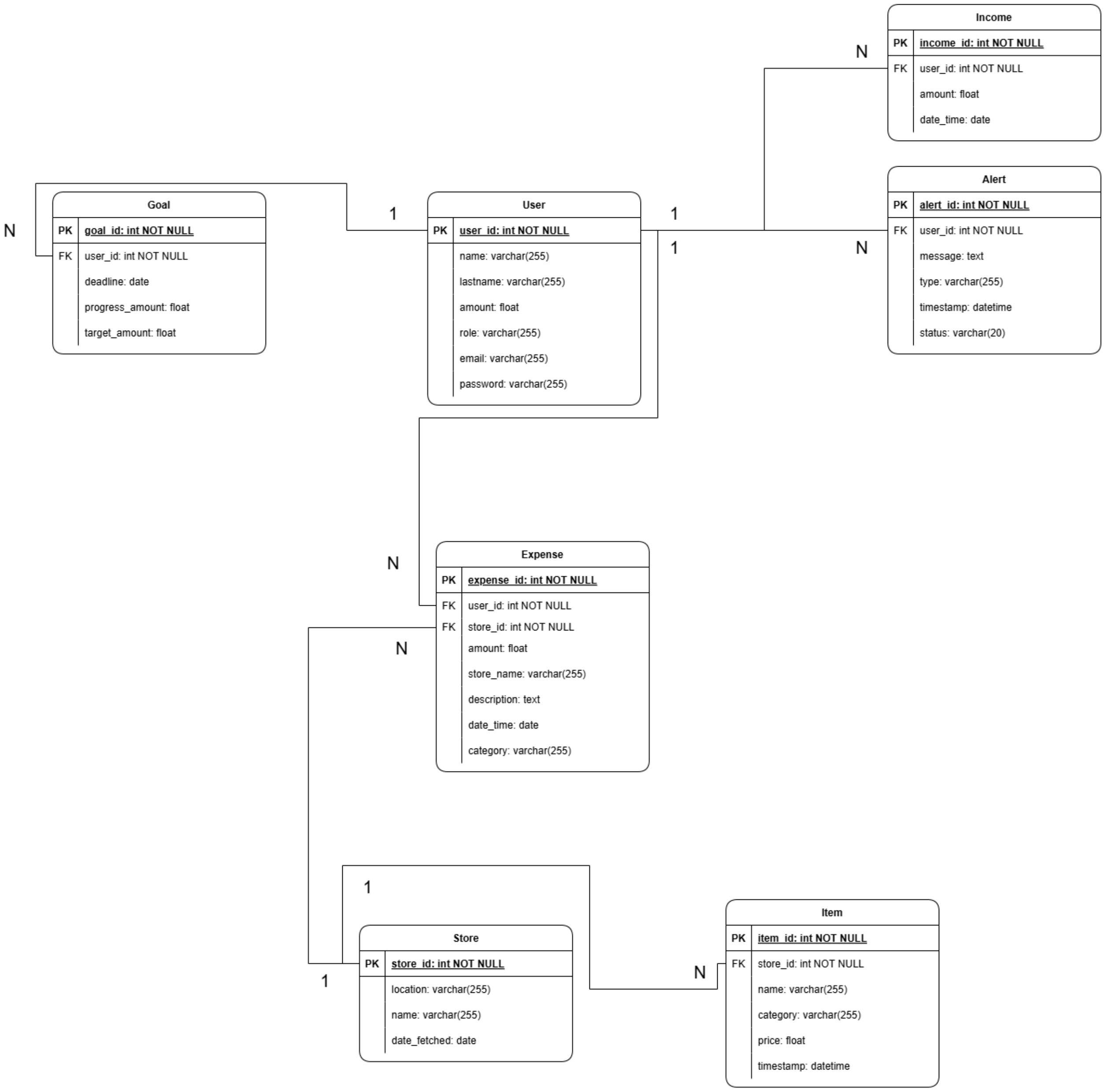

Figure 2: Relational model

Data analysis and optimisation

The data model is in third normal form. No further optimisation is required.

Physical Design Phase

Physical Data Model

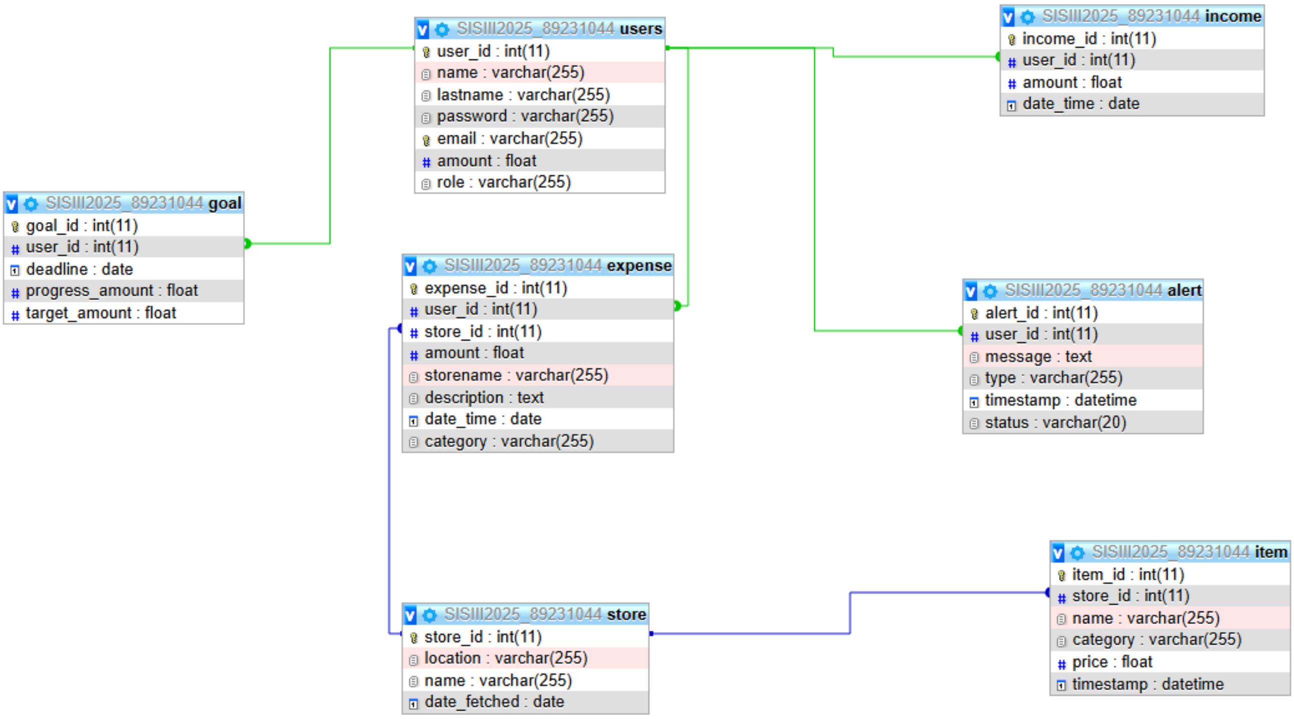

Figure 3: Physical database model

Object-oriented Analysis

UML Class Diagram

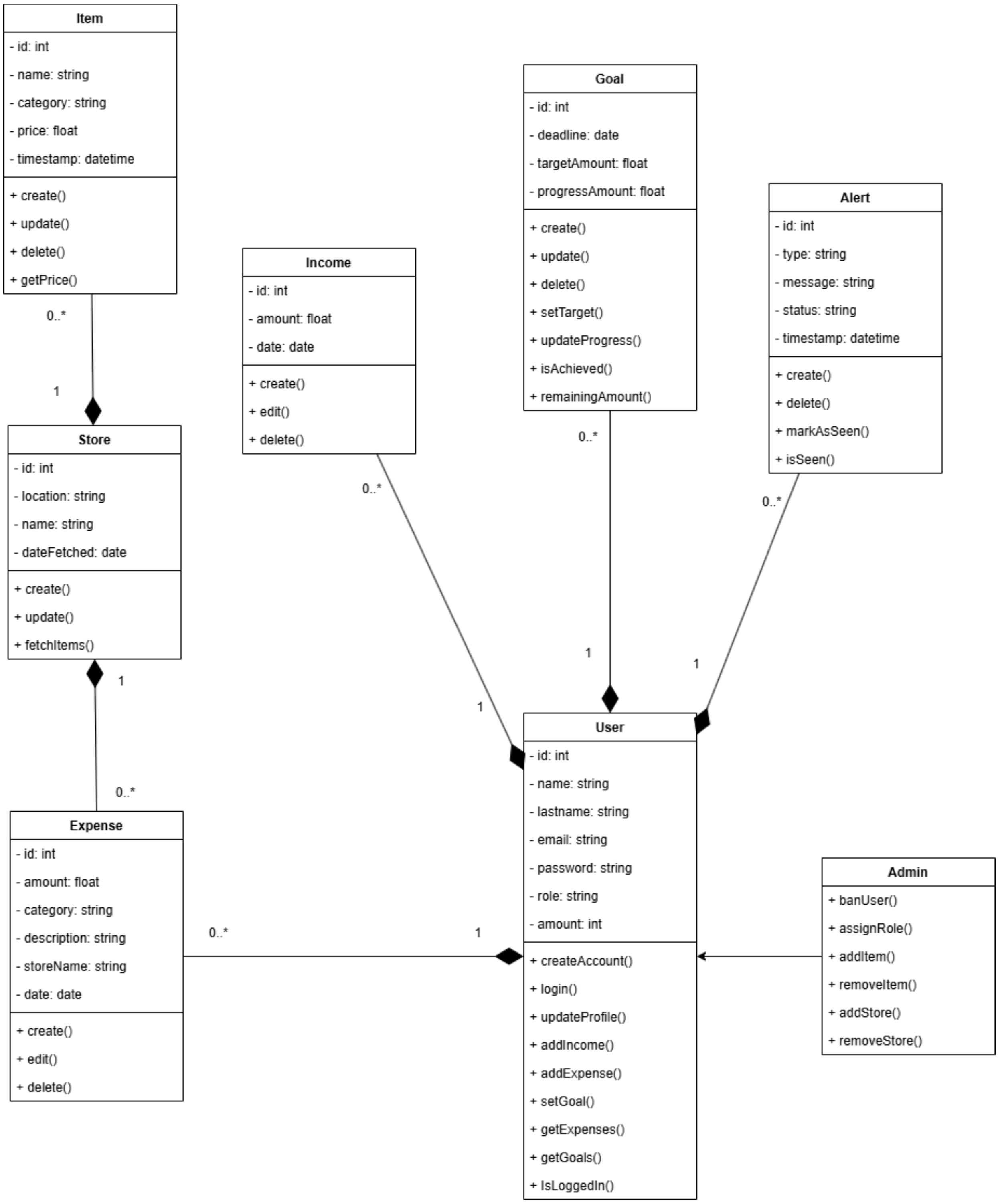

Figure 4: UML Class Diagram

UML Sequence Diagrams

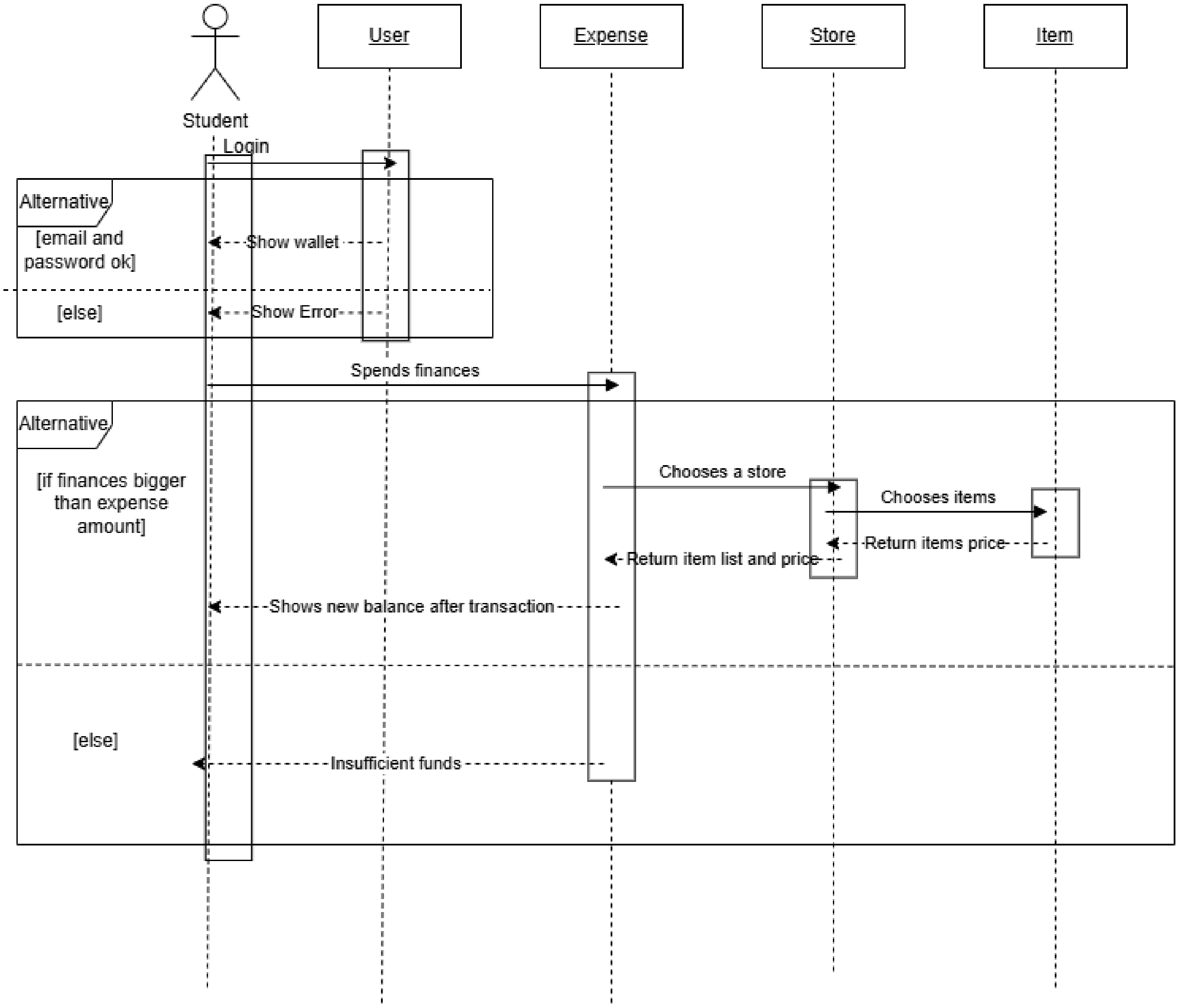

Figure 5: UML Sequence Diagram

UML Use-case Diagrams

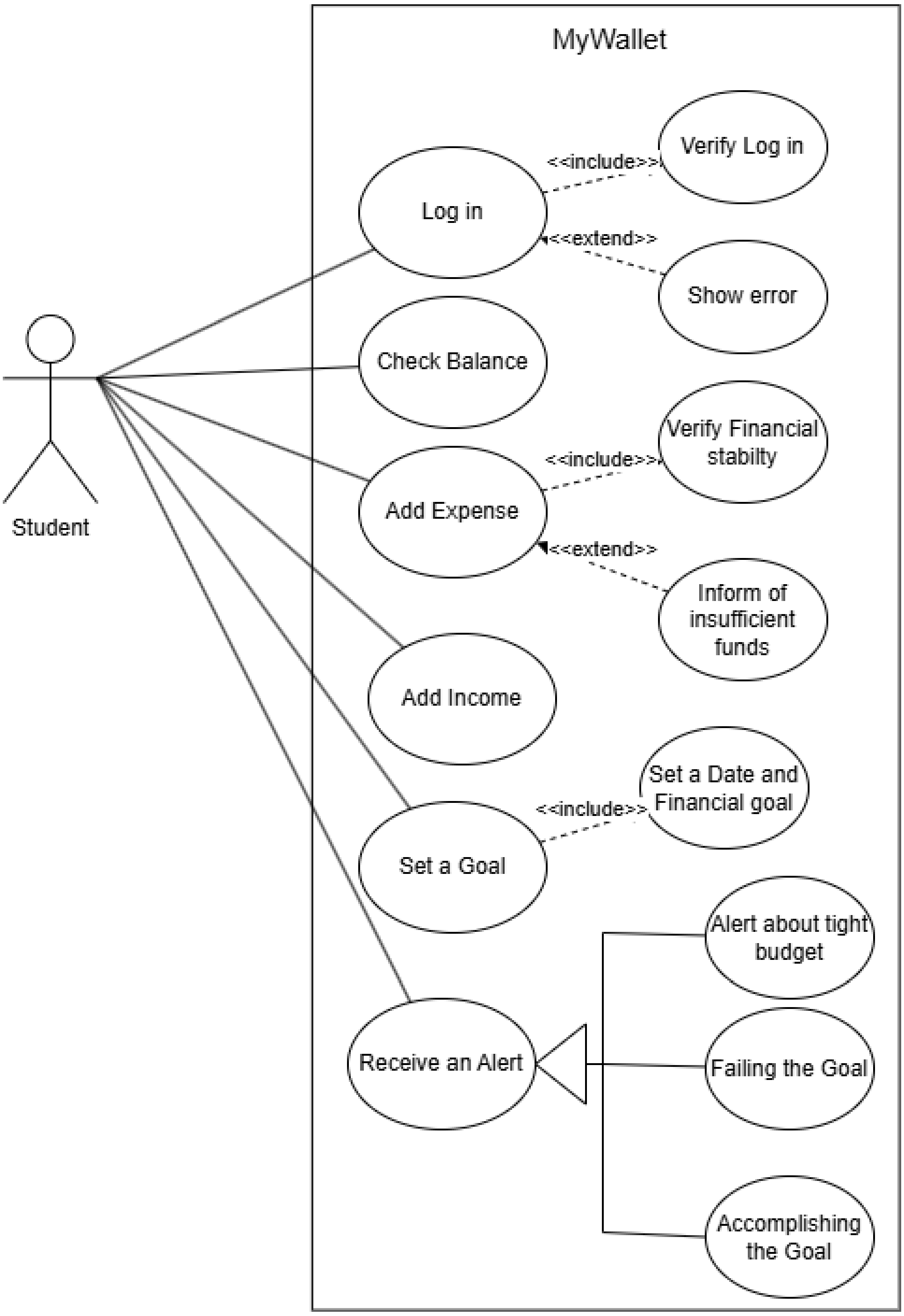

Figure 6: UML Use-case Diagram