**Software Design Document**

**Minimalist Finance Tracker for College Students**

**Prepared by:**

**Mariyam Noor ul ain (NUM-BSCS-2022-25)**

**Eshna Eman (NUM-BSCS-2022-51)**

**Namal University, Mianwali**

**Date: 23 - 05 - 2025**

### 

**Table of Contents**

[1. Introduction 3](#_Toc200926050)

[**1.1 Purpose** 3](#_Toc200926051)

[**1.2 Scope** 3](#_Toc200926052)

[**1.3 Overview** 3](#_Toc200926053)

[**1.4 Reference Material** 3](#_Toc200926054)

[**1.5 Definitions and Acronyms** 3](#_Toc200926055)

[2. System Overview 4](#_Toc200926056)

[3. System Architecture 4](#_Toc200926057)

[**3.1 Architectural Design** 4](#_Toc200926058)

[**3.2 Decomposition Description** 5](#_Toc200926059)

[**3.3 Design Rationale** 10](#_Toc200926060)

[4. Data Design 10](#_Toc200926061)

[**4.1 Data Description** 10](#_Toc200926062)

[**4.2 Data Dictionary** 10](#_Toc200926063)

[**5. Component Design** 10](#_Toc200926064)

[6. Human Interface Design 11](#_Toc200926065)

[**6.1 Overview of User Interface** 11](#_Toc200926066)

[**6.2 Screen Images** 11](#_Toc200926067)

[7. Requirements Matrix 12](#_Toc200926068)

[8. Appendices 21](#_Toc200926069)

[● Appendix A: Diagrams 21](#_Toc200926070)

## 

# 1. Introduction

### **1.1 Purpose**

This Software Design Document (SDD) describes the architecture and system design of the Minimalist Finance Tracker for College Students. It serves as a blueprint for developers to implement the system based on the requirements outlined in the Software Requirements Specification (SRS) document. The intended audience includes the development team, project managers, quality assurance personnel, and stakeholders.

### **1.2 Scope**

The Minimalist Finance Tracker is a mobile and web-based application designed to help college students manage their finances with simplicity and efficiency. The system supports manual expense entry, budget management, visual analytics, parent-student financial connections, fund transfers, and semester-wise expense tracking. Optional bank integration and offline functionality ensure accessibility. The application aims to foster responsible financial habits among students while maintaining a minimalist design to avoid feature bloat.

### **1.3 Overview**

This document is organized as follows:

* Section 2 provides a high-level overview of the system’s functionality and context.
* Section 3 details the system architecture, including subsystems and their interactions.
* Section 4 describes the data structures and storage mechanisms.
* Section 5 outlines the detailed design of individual components.
* Section 6 covers the user interface design.
* Section 7 maps SRS requirements to system components.
* Section 8 includes appendices for supporting details, such as diagrams.

### **1.4 Reference Material**

* Software Requirements Specification: Minimalist Finance Tracker for College Students (Version 1.0, 06/05/2025)
* IEEE Std 1016-2009: IEEE Recommended Practice for Software Design Descriptions
* IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
* Comprehensive Proposal: Minimalist Finance Tracker for College Students
* WCAG 2.1 Guidelines for Accessibility

### **1.5 Definitions and Acronyms**

* Category: A classification for expenses (e.g., Food, Books).
* Emergency Fund Request: A formal request from a student to a parent/guardian for additional funds.
* Semester: An academic period defined by start and end dates.
* Transaction: Any movement of funds into or out of a student’s account.
* Visualization: Graphical representation of financial data.
* Goal: A financial target with a specified amount and deadline.
* Budget Period: Timeframe for budget allocation (monthly, weekly, semester).
* SRS: Software Requirements Specification.
* SDD: Software Design Document.
* DFD: Data Flow Diagram.
* API: Application Programming Interface.
* WCAG: Web Content Accessibility Guidelines.

# 2. System Overview

The Minimalist Finance Tracker is a standalone application with optional bank integration, designed to provide college students with a simple, intuitive platform for financial management. It includes a mobile app for students and parents/guardians and a web portal for detailed analysis. Key features include manual expense logging, budget tracking, visual reports, parent-student fund transfers, emergency fund requests, and semester-specific expense tracking. The system emphasizes minimalism, offline functionality, and compliance with data privacy regulations to ensure accessibility and security.

# 3. System Architecture

### **3.1 Architectural Design**

The system follows a client-server architecture with a modular design to ensure scalability, maintainability, and simplicity. The architecture is divided into the following high-level subsystems:

* Client-Side Applications:
  + Student Mobile App: Handles expense entry, budget management, goal setting, and semester tracking.
  + Parent/Guardian Mobile App: Manages fund transfers, emergency requests, and student spending oversight.
  + Web Portal: Provides advanced analytics and report exports for both user types.
* Server-Side Backend:
  + Application Logic Layer: Processes user requests, manages business logic, and coordinates data flow.
  + Data Storage Layer: Stores user data, transactions, budgets, and semester records in a database.
  + API Gateway: Facilitates secure communication with external banking APIs (optional) and client apps.
* External Interfaces: Banking APIs, push notification services, and cloud storage for backups.

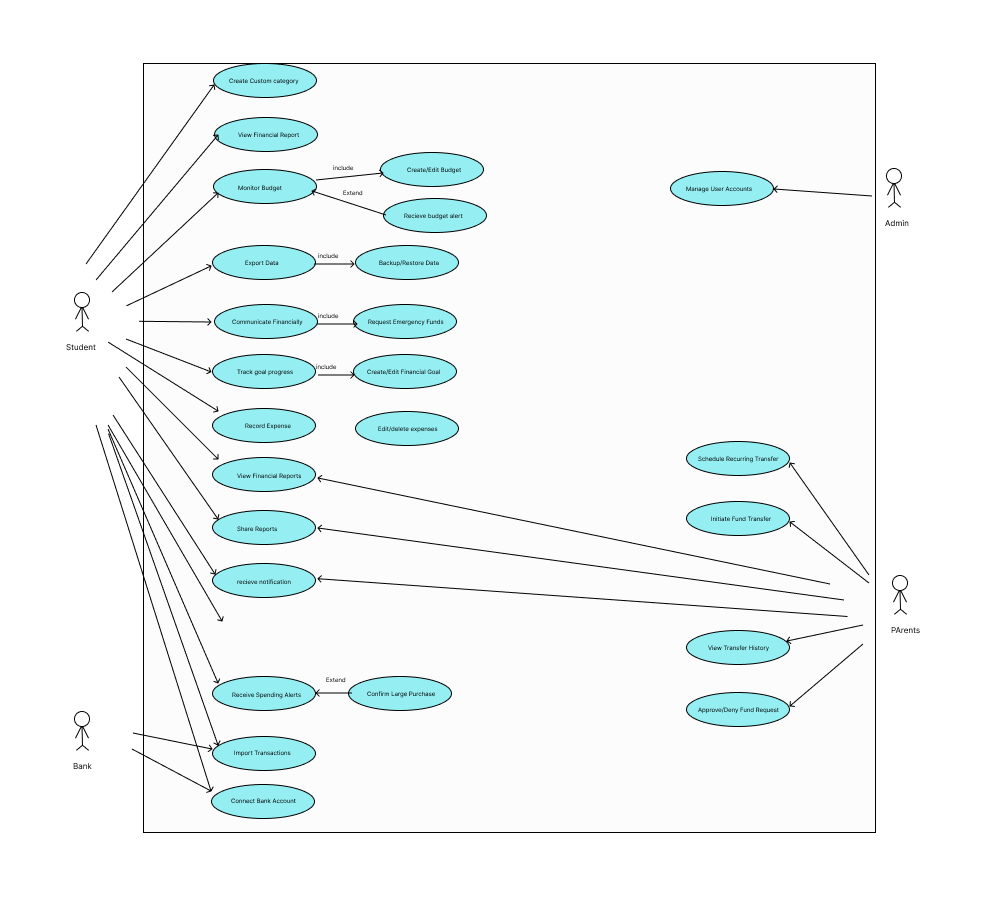
Diagram: Include a high-level architecture diagram showing the client apps, backend server, database, and external interfaces, with arrows indicating data flow (e.g., HTTPS for client-server communication, RESTful APIs for banking integration)

### **3.2 Decomposition Description**

The system is decomposed into the following subsystems:

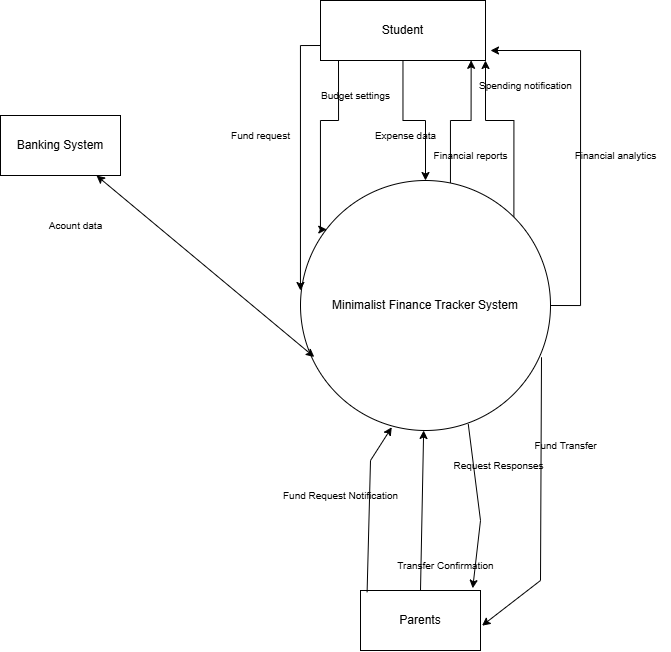
1. User Interface Subsystem: Manages the student and parent mobile apps and web portal interfaces.
2. Expense Management Subsystem: Handles manual expense entry, categorization, and editing
3. Budget Management Subsystem: Manages budget creation, tracking, and alerts
4. Analytics and Reporting Subsystem: Generates visual reports and exportable data
5. Parent-Student Connection Subsystem: Facilitates account linking and communication
6. Fund Transfer Subsystem: Processes fund transfers and emergency requests
7. Withdrawal Tracking Subsystem: Logs and tracks cash withdrawals
8. Notification Subsystem: Sends spending alerts and notifications
9. Goal Setting Subsystem: Manages financial goal creation and tracking
10. Bank Integration Subsystem: Handles optional bank API connections
11. Semester Tracking Subsystem: Tracks and analyzes semester-specific expenses Authentication and Security Subsystem: Ensures secure access and data protection.

**Use Case Diagram:**

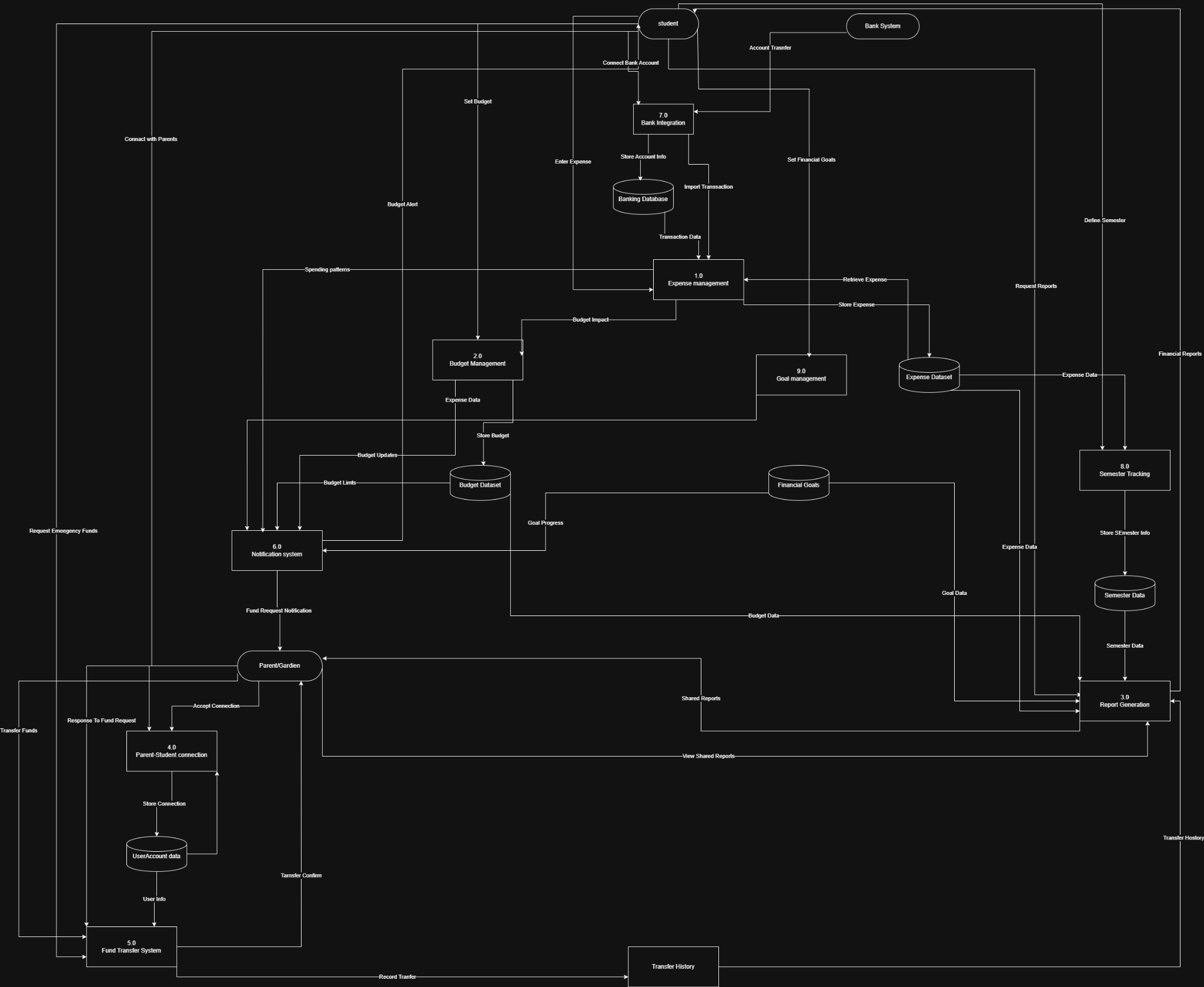
****

**Data Flow Diagram:**

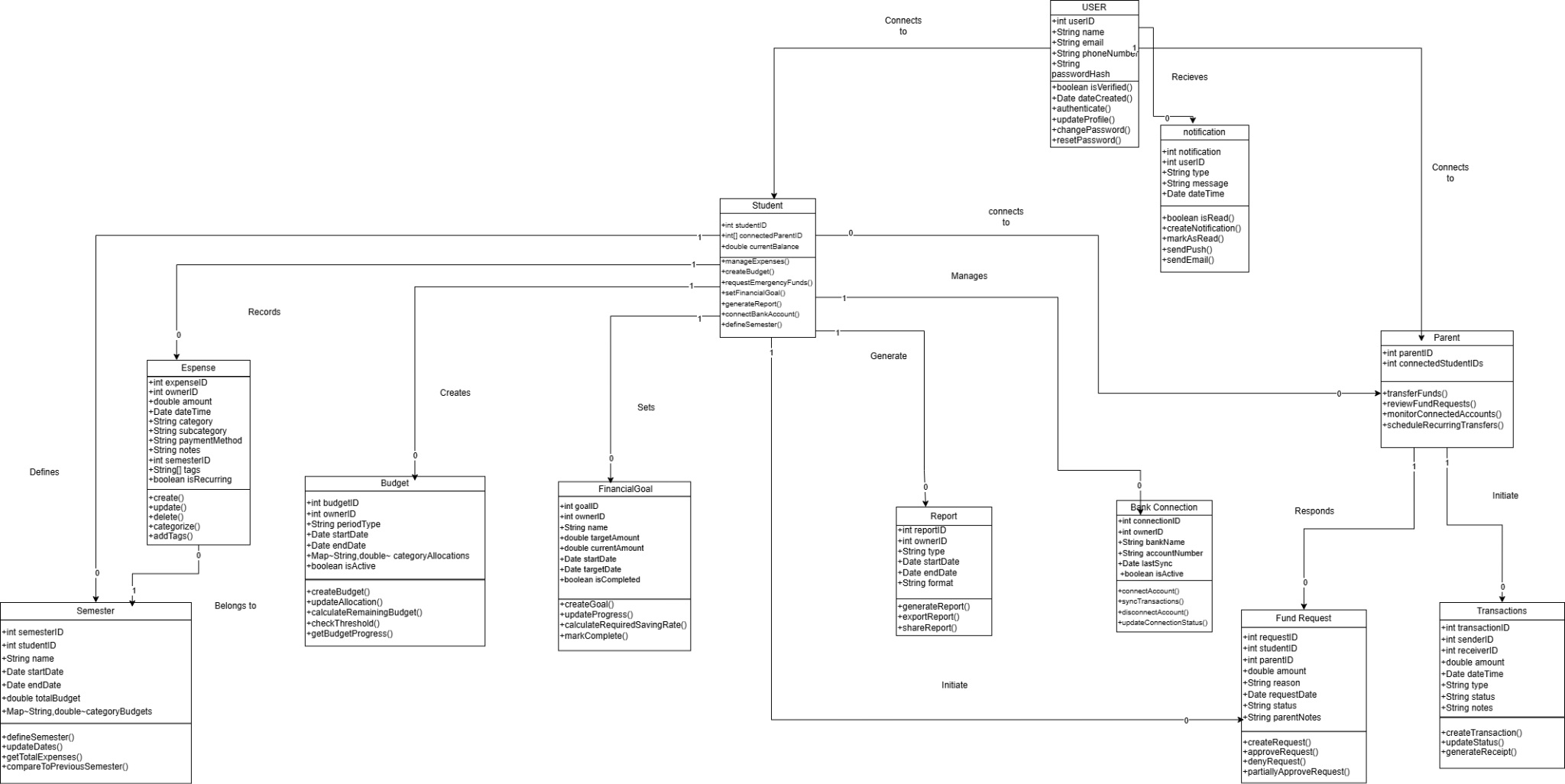
**Level 0:**

****

**Level 1:**

****

**Class Diagram:**

****

### **3.3 Design Rationale**

The client-server architecture was chosen for its scalability, support for cross-platform apps, and ability to handle offline functionality through local caching. A modular design ensures maintainability and allows independent updates to subsystems. Alternative architectures, such as a fully offline desktop application, were considered but rejected due to the need for real-time parent-student interactions and optional bank integration. Trade-offs include increased complexity for offline synchronization but improved accessibility for users with limited connectivity.

# 4. Data Design

### **4.1 Data Description**

The system uses a relational database to store user data, transactions, budgets, and semester records. Key entities include:

* User: Stores student and parent/guardian profiles (e.g., ID, name, email, role).
* Expense: Records expense details (amount, date, category, notes, payment method).
* Budget: Tracks budget allocations (category, period, amount, progress).
* Transaction: Logs fund transfers and withdrawals (amount, date, sender, receiver).
* Semester: Defines academic periods and associated expenses (start date, end date, expenses).
* Goal: Stores financial goals (target amount, deadline, progress).
* Notification: Manages alerts (type, threshold, recipient).

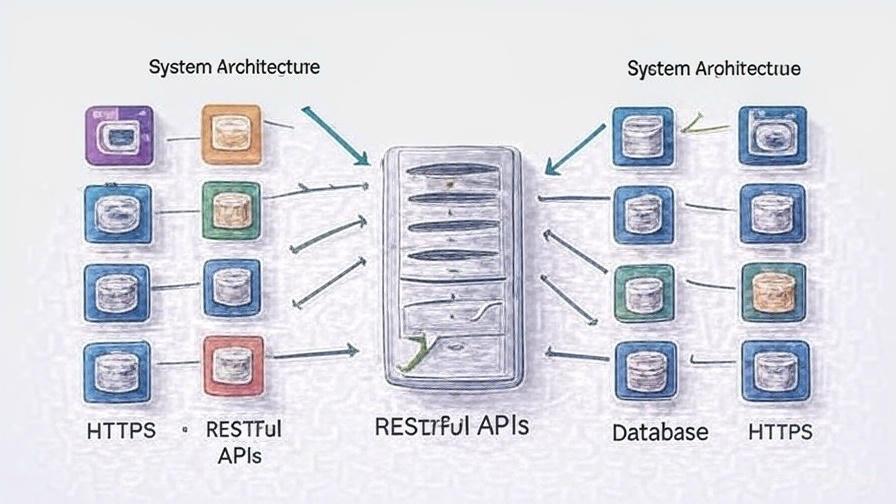
Data is stored in a cloud-based relational database (e.g., PostgreSQL) with local caching on mobile devices for offline access. Backup data is stored in a cloud storage service.

### **4.2 Data Dictionary**

### **5. Component Design**

Each subsystem is implemented as a module with specific responsibilities. Below is a high-level description of key components (detailed algorithms to be provided in pseudocode during implementation):

* Expense Management: Handles expense CRUD operations (create, read, update, delete).
* Budget Management: Calculates budget progress and triggers alerts.
* Analytics and Reporting: Generates charts (e.g., budget vs. actual, expense heat maps).
* Parent-Student Connection: Manages secure account linking and communication threads.
* Fund Transfer: Processes transfers with confirmation and history logging.
* Semester Tracking: Associates expenses with semesters and generates summaries.
* Notification: Sends push/email/SMS notifications based on thresholds.
* Authentication: Implements OAuth 2.0 and multi-factor authentication.



### 

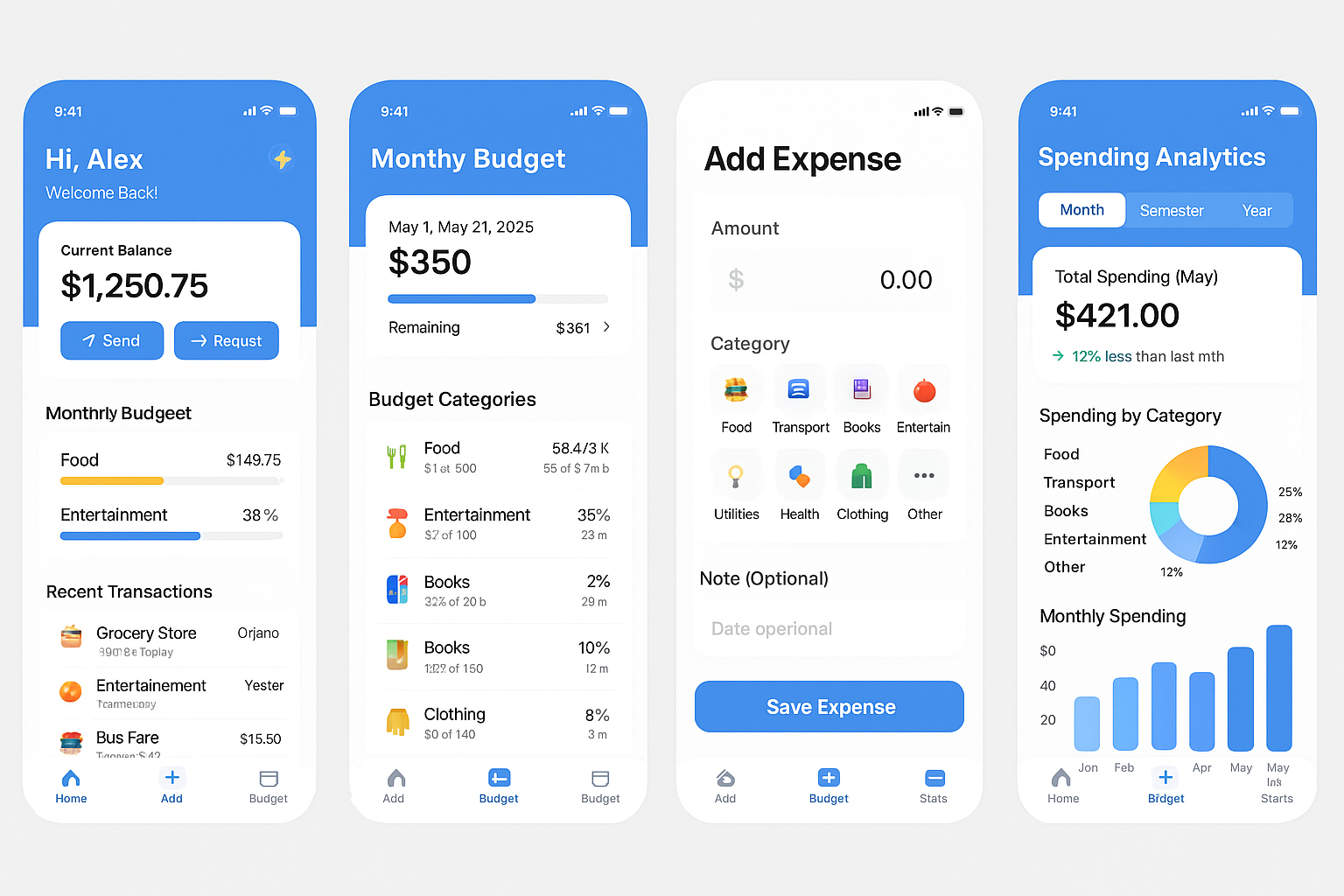
# 6. Human Interface Design

### **6.1 Overview of User Interface**

The user interface follows a minimalist design with intuitive navigation:

* Student Mobile App: Features a dashboard with recent expenses, budget status, and quick-add expense buttons. Supports dark/light modes and WCAG 2.1 accessibility.
* Parent/Guardian Mobile App: Displays fund transfer options, emergency request notifications, and student spending summaries (if shared).
* Web Portal: Offers detailed analytics, report exports, and account management with responsive design.

### **6.2 Screen Images**



# 7. Requirements Matrix

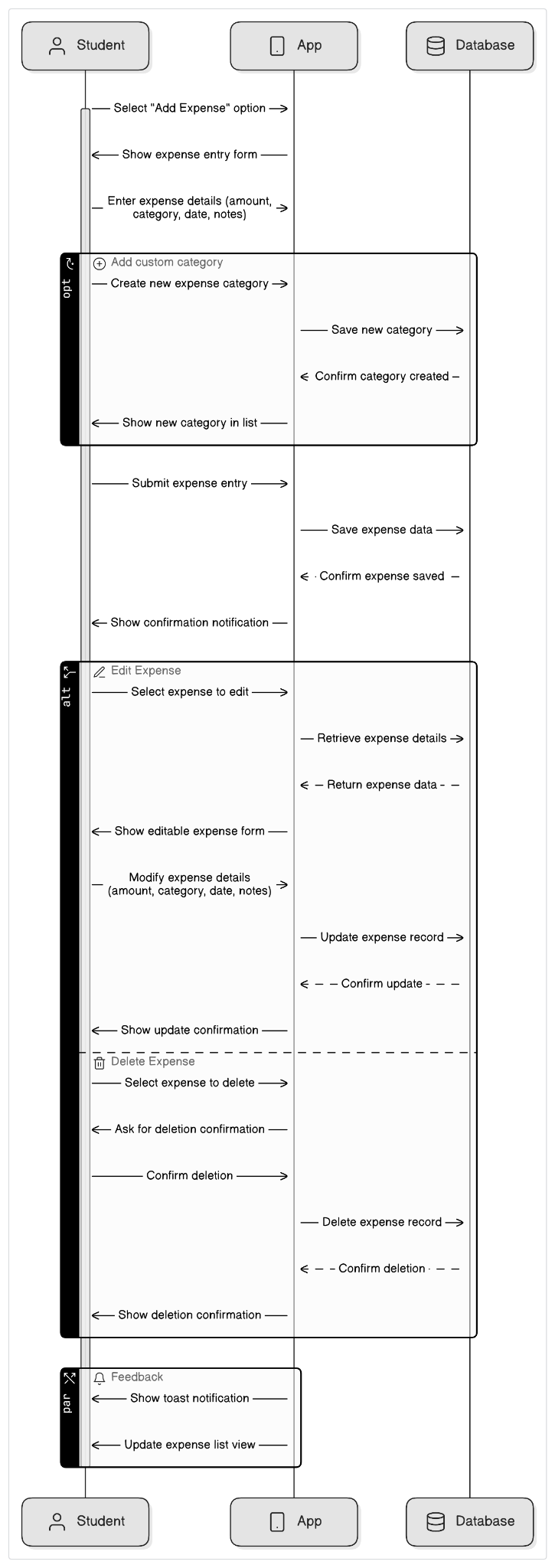
The following table maps SRS functional requirements to system components:

|  |  |  |
| --- | --- | --- |
| Requirement ID | Description | Component/Subsystem |
| REQ-1-1 to 1-5 | Manual Expense Entry | Expense Management Subsystem |
| REQ-2-1 to 2-6 | Budget Management | Budget Management Subsystem |
| REQ-3-1 to 3-6 | Visual Reports & Analytics | Analytics and Reporting Subsystem |
| REQ-4-1 to 4-5 | Export Functionality | Analytics and Reporting Subsystem |
| REQ-5-1 to 5-6 | Parent-Student Connection | Parent-Student Connection Subsystem |
| REQ-6-1 to 6-6 | Fund Transfer System | Fund Transfer Subsystem |
| REQ-7-1 to 7-5 | Withdrawal Tracking | Withdrawal Tracking Subsystem |
| REQ-8-1 to 8-6 | Emergency Fund Requests | Fund Transfer Subsystem |
| REQ-9-1 to 9-5 | Spending Notifications | Notification Subsystem |
| REQ-10-1 to 10-6 | Financial Goal Setting | Goal Setting Subsystem |
| REQ-11-1 to 11-6 | Bank Account Integration | Bank Integration Subsystem |
| REQ-12-1 to 12-20 | Semester-wise Expense Tracking | Semester Tracking Subsystem |

### 

**Sequence Diagram:**

Manual Expense Entry:

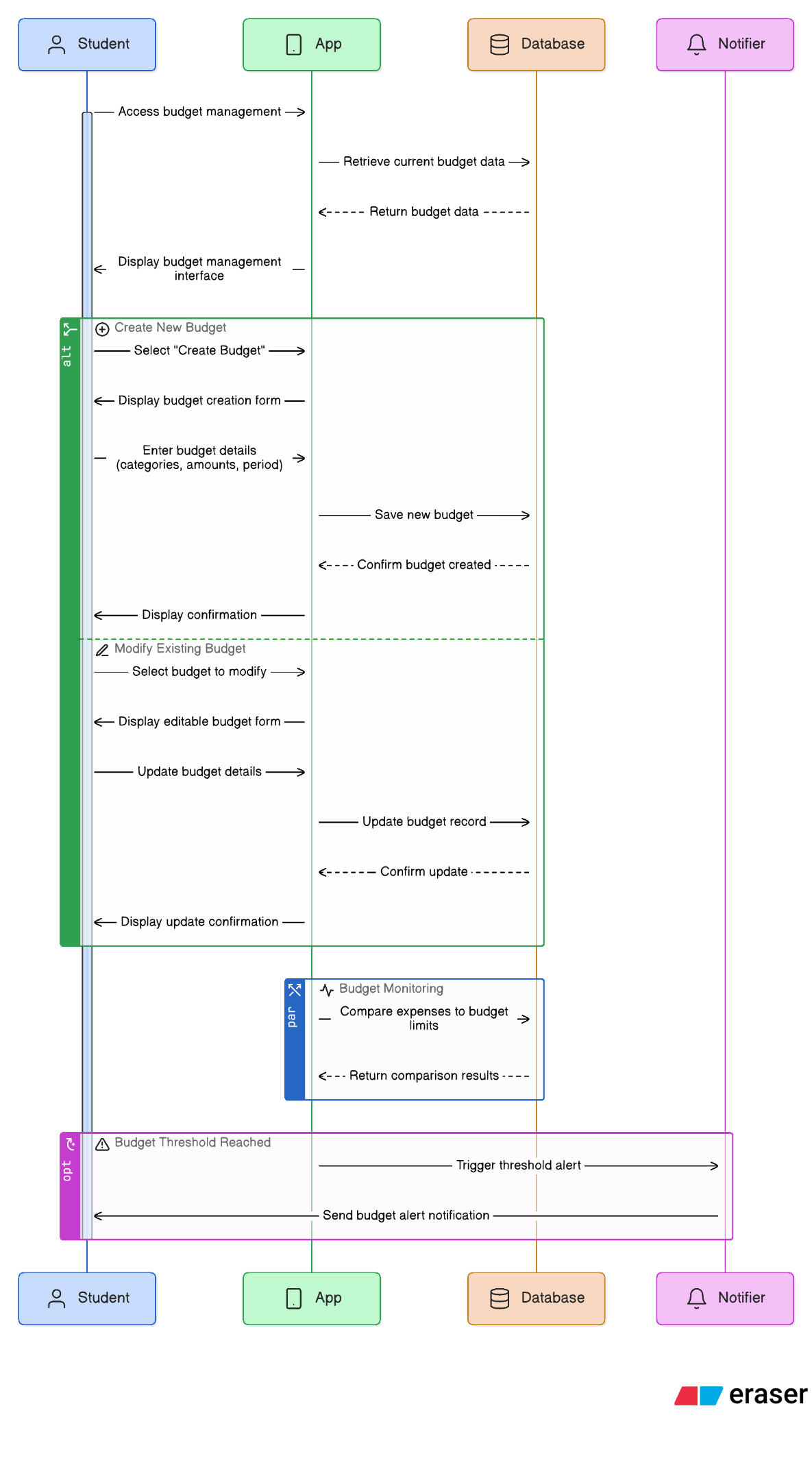
****

Financial Report Generating:

A white sheet of paper with black text

AI-generated content may be incorrect.

Budget Management:



Emergency Fund Request:

A screenshot of a computer

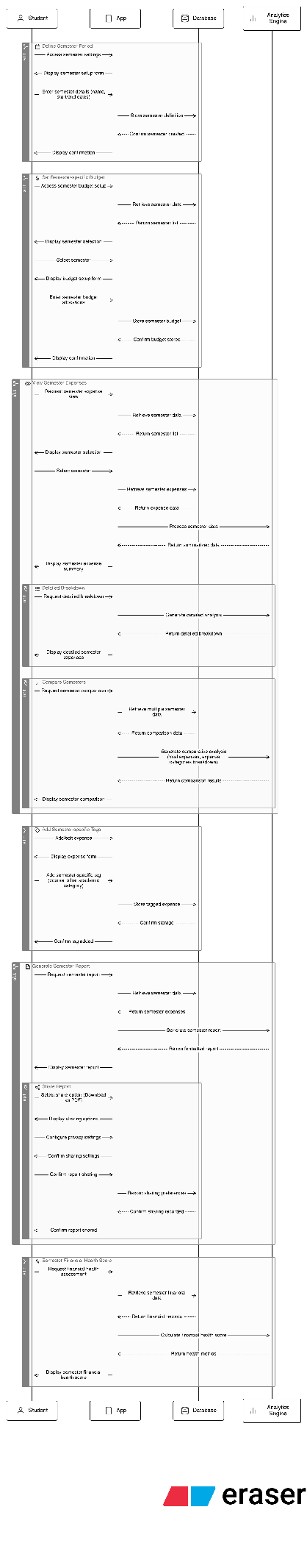
AI-generated content may be incorrect.

Cash Withdrawal Tracking:

A screenshot of a computer

AI-generated content may be incorrect.

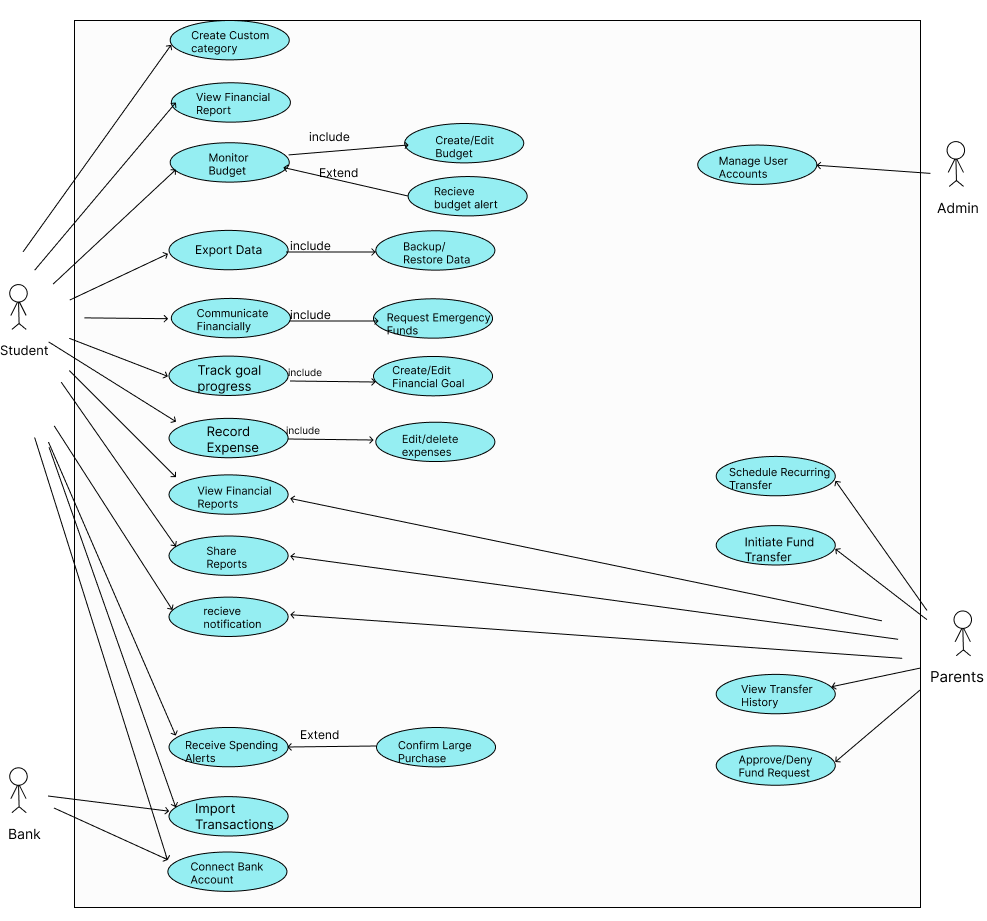
Semester Wise Tracking:



# 8. Appendices

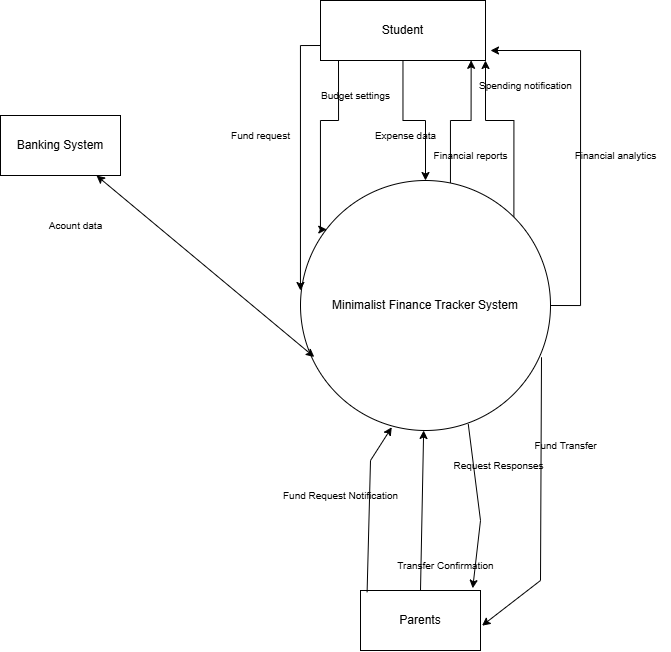
### Appendix A: Diagrams

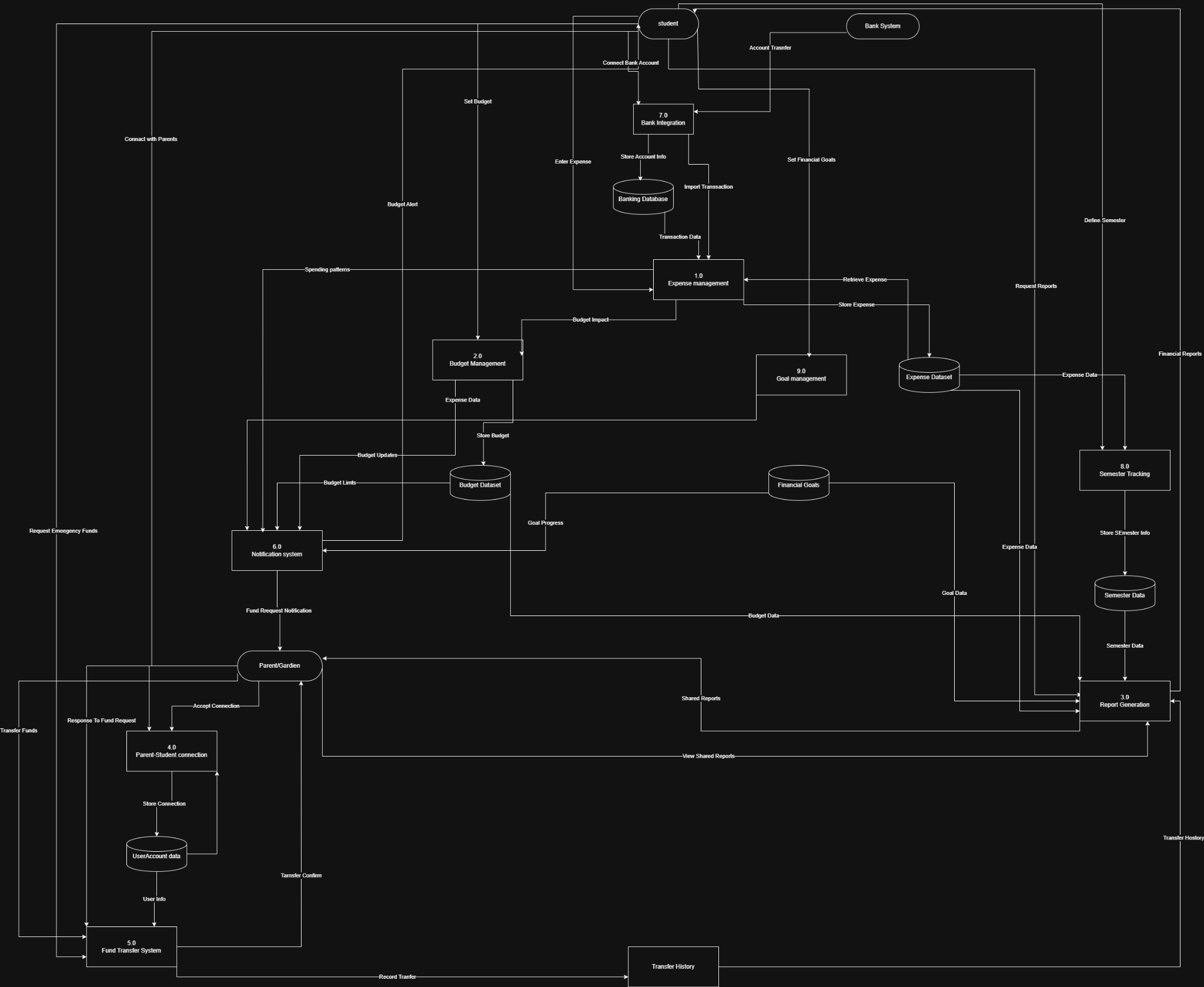
Use Case diagram:



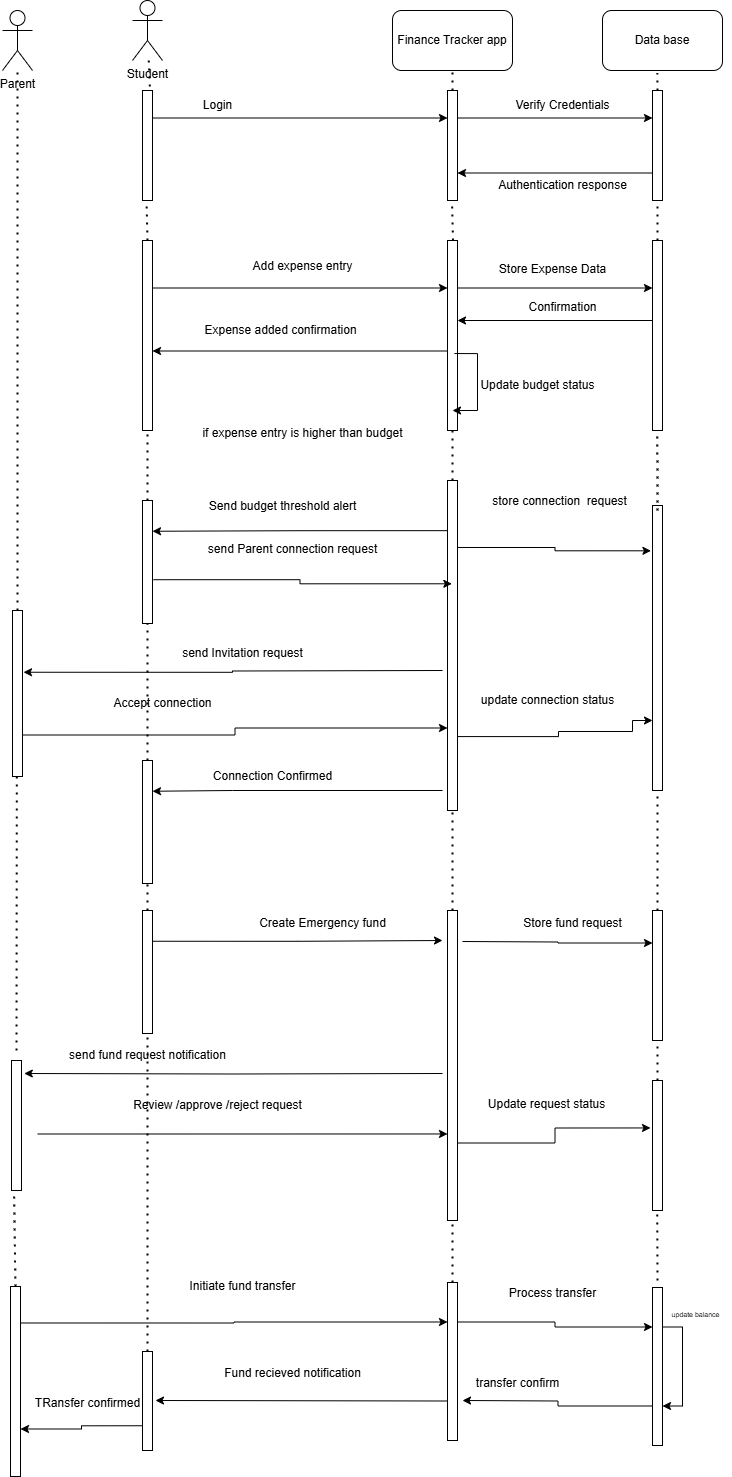
DFD Diagram:

Level 0:

Level 1:



SSD:



Class Diagram:

