Group 9

DebtSetGo – Personal Financer

Software Engineering Project

Names:

Jigyasa Jha
Mohsin Hussain
Zihan Tang
Fikru Gedeon
Prithvi Kurmi

Semester: Fall 2025

Group Number: 9

Coordinator: Fikru Gedeon

Name of the Guide: Dr. Tushara Sadasivuni

Date: (10/20/2025)

Table of Contents

1.0 INTRODUCTION..... 3

1.1 Software Engineers' information..... 3

1.2 Planning and Scheduling..... 3

1.3 Teamwork Basics..... 4

1.4 Problem Statement..... 4

1.5 System Requirements..... 5

1.5.1 Context Diagram..... 5

1.5.2 Activity Diagram..... 6

2.0 REQUIREMENTS..... 7

2.1 Use Cases..... 7

2.2 Requirements..... 10

2.3 Use Case Diagrams..... 13

3.0 DATABASE TABLES..... 15

4.0 CLASS DIAGRAM..... 16

5.0 BEHAVIORAL MODELING..... 16

6.0 IMPLEMENTATION..... 16

7.0 GITHUB..... 16

References..... 16

1.0 INTRODUCTION

1.1 Software Engineers' information

Jigyasa Jha: Senior computer science and political science double major

Mohsin Hussain:

Zihan Tang: First year CS PhD student, research area is VR, AR, security, privacy

Fikru Gedeon: Senior computer science major

Prithvi Kurmi: Junior Computer Science major, STEM Tutor CS, SI Leader Calc II, Backend Developer

1.2 Planning and Scheduling

Assignee Name	Email Address	Task	Duration (hours)	Dependency	Due Date	Evaluation
Jigyasa Jha	jjha1@student.gsu.edu	Implement frontend and logic	20	None	10/21/2025	
Mohsin Hussain	mhussain11@student.gsu.edu	Implement database	20	None	10/21/2025	
Zihan Tang	ztang15@student.gsu.edu	Class diagram Behavior modeling	20	None	10/21/2025	
Fikru Gedeon	gfikru@student.gsu.edu	Implement database	20	None	10/21/2025	
Prithvi Kurmi	pkurmi@student.gsu.edu					

1.3 Teamwork Basics

Ground rules:

- Each team member should contribute to the project to the best of their abilities.
- Each team member should take part in group meetings, if any team member is unable to take part in a group meeting, they should let the rest of their group members know ahead of time.
- Team members are expected to listen to opinions shared by other team members respectfully and without interrupting.
- Team members are expected to take ownership of assigned tasks and inform the group of any obstacles that arise while working on their assigned task.
- Team members are expected to follow project guidelines.
- When facing difficult behaviors from other team members. The rest of the team will discuss what the source of the problem is and decide how to proceed

When facing group problems with the project, team members can call for a meeting or directly contact any specific team member to resolve the issue.

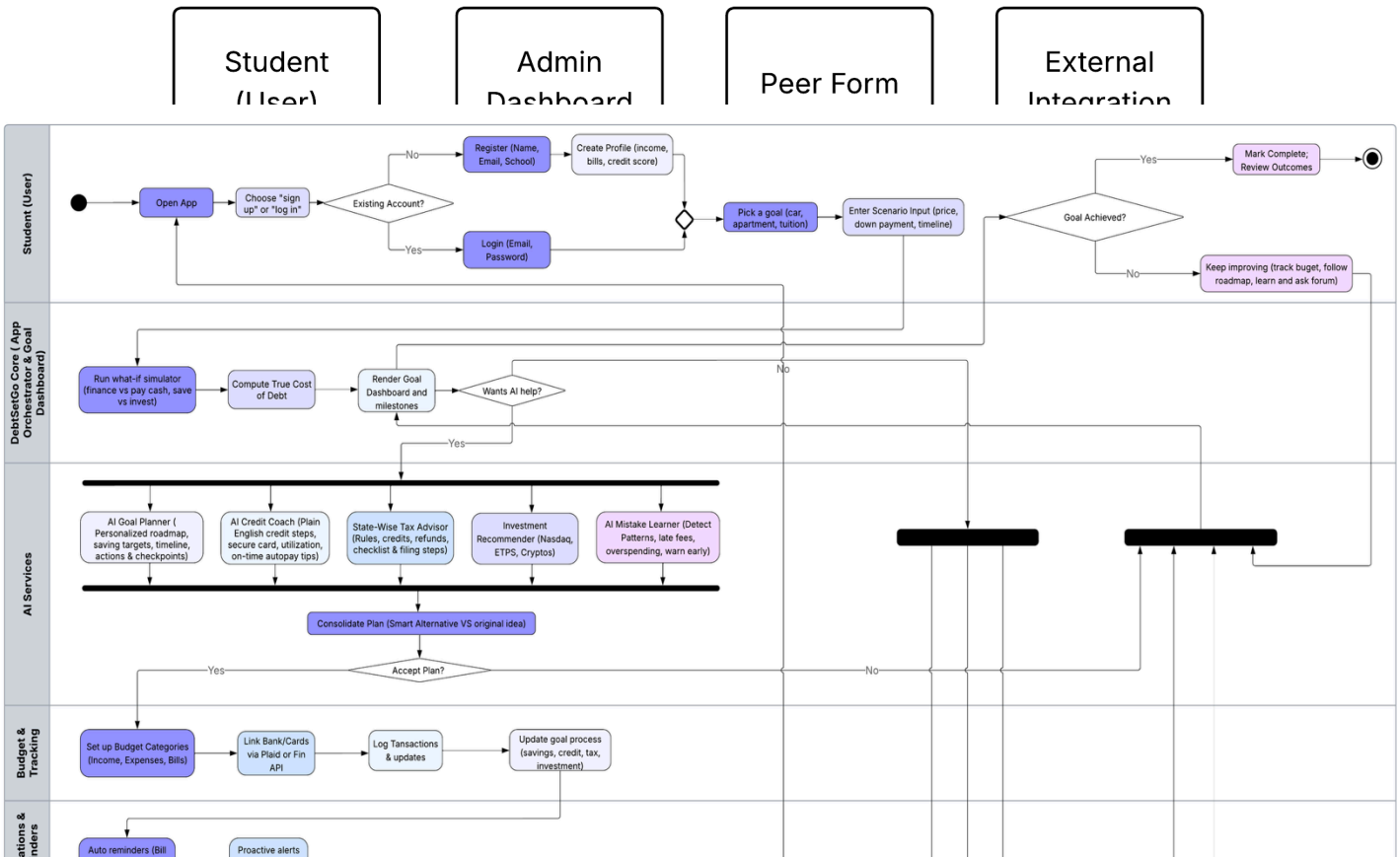
1.4 Problem Statement

- DebtSetGo is an AI-powered personal finance platform designed for students to manage credit, taxes, savings, and investments in one place.
- Features include:
    - **User Management**: Handles login, signup, and profile personalization.
    - **AI Goal Planner**: Generates step-by-step financial roadmaps.
    - **AI Credit Coach**: Provides credit-building tips and reminders.
    - **State-Wise Tax Advisor**: Offers state-specific tax rules and refund estimates.
    - **Investment Recommender**: Suggests safe, student-friendly investment options.
    - **What-If Simulator**: Compares scenarios like saving vs. investing.
    - **AI Mistake-Learner**: Learns user patterns and prevents repeat mistakes.
    - **Budget Tracker**: Tracks income, expenses, and bills.
    - **Goal Dashboard**: Displays progress visually with charts.
    - **Notifications & Reminders**: Sends alerts for bills, deadlines, and goals.
    - **Educational Library**: Provides simple guides on financial topics.
    - **Peer Forum + Admin Dashboard**: Community space with moderation.
    - **Achievement Cards**: Auto-generates social media-friendly success cards.
    - **Smart Payment Suggestions**: Helps maximize credit card rewards.
- **Who is DebtSetGo for?** College students and young adults who are managing finances for the first time.
- **What problem does it solve?** It prevents costly financial mistakes (e.g., bad loans, missed payments) and teaches financial literacy in an actionable, student-friendly way
- **What alternatives are available?** Mint, Credit Karma, NerdWallet…

- **Why is this project compelling and worth developing?** It addresses a unique pain point for students who lack personalized financial guidance and combines education, AI-driven insights, and motivation.
- **Top-level objectives & differentiators:**
  - Objectives: Help students build assets, not debt; provide personalized planning; increase financial literacy.
  - Differentiators: Student-first design, AI-driven goal setting, plain-English explanations, social and motivational features.
- **Target Customers:** U.S. college students and young adults (ages 18–25).
- **Scope:** Mobile-friendly web app (future mobile app option) supports login for students and admin dashboard.
- **Competitors and novel approach:**
  - Competitors: Mint, Credit Karma, NerdWallet
  - Novelty: AI-based goal planner, proactive mistake prevention, integrated tax and credit help, student community.
- **Interesting technical points:**
  - AI-driven personalized financial planning
  - Predictive mistake prevention
  - Interactive financial simulations and visualizations
- **Login features:** Yes, it includes Client Login

1.5 System Requirements

1.5.1 Context Diagram



1.5.2 Activity Diagram

2.0 REQUIREMENTS

2.1 Use Cases

Use Case no.: 01
Use Case Name: User Registration and Login
Actors: Student (User)
Description: A new user signs up with email and password, verifies email, and logs in. Returning users authenticate and reach the dashboard.
Alternate Path: Social login (school SSO) if enabled.
Exception Path: Invalid credentials; locked account after N failed attempts; email not verified.
Pre-condition: System reachable; user not authenticated.
Post-condition: Auth session created; user profile stub stored.

Use Case no.: 02
Use Case Name: Profile Personalization
Actors: Student (User)
Description: User sets name, school and state, income sources, credit card ownership, financial goals preference.
Alternate Path: Skip non mandatory fields; import from previous survey.
Exception Path: Invalid state code; network error while saving.
Pre-condition: User authenticated.
Post-condition: Profile updated; personalization saved.

Use Case no.: 03
Use Case Name: AI Goal Planner
Actors: Student (User)
Description: User provides financial goals such as saving, loan repayment, or building credit. The system's AI Goal Planner analyzes income and expenses to generate a personalized saving plan with milestones.
Alternate Path: User imports a goal template or links previous budget data.
Exception Path: Invalid inputs or missing income data.
Pre-condition: User authenticated, and profile exists.
Post-condition: Goal data and monthly targets stored in database.

Use Case no.: 04
Use Case Name: Budget Analytics & Visualization
Actors: Student (User)
Description: The system retrieves budget and transaction data, categorizes expenses, and displays charts showing planned vs. actual spending. Alerts are generated for overspending.
Alternate Path: User changes time frame or export report to CSV.
Exception Path: Missing transaction data or database error.
Pre-condition: Budget data exists.

Post-condition: Graphs and alerts visible on dashboard.

Use Case no.: 05
Use Case Name: AI Mistake Learner
Actors: Student (User)
Description: System analyzes past financial behaviors and detects repeat mistakes such as late payments. It then sends preventive tips to avoid recurrence.
Alternate Path: User snoozes alert for specific patterns.
Exception Path: Insufficient historical data to train AI model.
Pre-condition: Minimum three months of activity available.
Post-condition: Recommendations recorded and displayed under “Insights”.

Use Case no.: 06
Use Case Name: Financial Progress Report Generator
Actors: Student (User)
Description: The system compiles a detailed monthly report showing income, expenses, savings progress, and goal achievements. It highlights improvements and provides personalized suggestions for better money management.
Alternate Path: User selects a custom date range or exports the report as a PDF/CSV.
Exception Path: Missing data for selected period or report generation timeout.
Pre-condition: User must have at least one month of recorded transactions and goals.
Post-condition: Report generated, saved in user history, and available for download.

Use Case no.: 07
Use Case Name: AI Financial Advisor Chatbot
Actors: Student (User)
Description: An AI-powered chatbot provides real-time financial advice, answers questions about budgeting, goals, and spending habits, and suggests smart actions.
Alternate Path: User asks generic financial questions or requests detailed trend analysis.
Exception Path: AI service unavailable or insufficient user data for personalized responses.
Pre-condition: User authenticated; chatbot feature enabled.
Post-condition: Advice and recommendations displayed in chat.

Use Case no.: 08
Use Case Name: Investment Tracker and Advisor
Actors: Student (User)
Description: The system allows users to log their small investments (savings accounts, ETFs, or crypto). It tracks performance, calculates gains/losses, and provides AI-based suggestions for diversification or risk reduction.
Alternate Path: User imports investment data from a connected bank or portfolio app.
Exception Path: Invalid investment type, incorrect balance data, or failed third-party API.
Pre-condition: User authenticated, and investment tracking feature enabled.
Post-condition: Investment portfolio updated, and analytics shown in the “Wealth Insights” section.

Use Case no.: 09

Use Case Name: AI Credit Coach
Actors: Student (User)
Description: The system provides personalized credit-building advice, detects factors lowering credit scores, and suggests actions like paying bills on time or reducing utilization.
Alternate Path: User imports credit report or data manually.
Exception Path: User don't have access to credit report
Pre-condition: credit data available.
Post-condition: Credit score insights and improvement tips displayed on dashboard.

Use Case no.: 10
Use Case Name: State-Wise Tax Advisor
Actors: Student (User)
Description: The system provides state-specific tax guidance, refund estimations, and filing deadlines based on the user's location and income.
Alternate Path: User manually inputs state or uploads tax document.
Exception Path: Invalid or unsupported state code; missing income data.
Pre-condition: User profile completed with state info.
Post-condition: Personalized tax summary and recommendations displayed.

Use Case no.: 11
Use Case Name: Educational Library & Quiz Module
Actors: Student (User)
Description: Users access short articles, videos, and quizzes on financial literacy topics (budgeting, credit, investing).
Alternate Path: User downloads resources or saves for later.
Exception Path: Missing resource files or connection timeout.
Pre-condition: User authenticated.
Post-condition: Learning progress saved; quiz scores recorded.

Use Case no.: 12
Use Case Name: Bank Account Linking & Data Import
Actors: Student (User), External Financial Institution (via API/CSV)
Description: User connects bank(s) or credit accounts to import transaction history and account balances for automatic categorization and reconciliation.
Alternate Path: Manual CSV upload or manual transaction entry if API is unavailable.
Exception Path: Failed third-party API auth; stale or inconsistent CSV formats.
Pre-condition: User authenticated and consents to data linking.
Post-condition: Transactions imported, matched, and available for categorization.

Use Case no.: 13
Use Case Name: Notification & Communication System
Actors: Student (User), Admin
Description: informs students about key financial activities such as upcoming bills, deadlines, investment updates, tax reminders, and achievement milestones.
It also enables two-way communication between users and the platform.
Alternate Path: The user modifies settings to receive specific types of notifications.

Exception Path: No internet connection, App fails to deliver real-time notifications; Invalid email or app permission disabled, Notification cannot be delivered externally.
Pre-condition: User is successfully logged in; User profile includes verified contact information (email, app notifications enabled).
Post-condition: Notifications are delivered successfully to the user via selected channels.

Use Case no.: 14
Use Case Name: Fraud Detection & Dispute Workflow
Actors: Student (User), Fraud Team/Bank (System)
Description: System flags unusual transactions and allows user to review/confirm or start a dispute workflow that notifies the linked bank or internal fraud team.
Alternate Path: User marks transaction as safe; system learns to reduce false positives.
Exception Path: Insufficient transaction metadata to evaluate; user unreachable for confirmation.
Pre-condition: Connected accounts and transaction data present.
Post-condition: Transaction flagged status saved; dispute ticket created and user notified.

Use Case no.: 15
Use Case Name: Admin Dashboard & System Monitoring
Actors: Admin, Support Agent
Description: Admin can view system health, user activity metrics, manage content (educational resources), handle user support tickets, and perform escalations.
Alternate Path: Read-only analytics role for product management; escalate to engineering for incidents.
Exception Path: Admin action attempted without sufficient privileges.
Pre-condition: Admin authenticated with RBAC role.
Post-condition: Admin actions recorded in audit logs; support ticket state updated.

2.2 Requirements

Requirement number: 01
Use Case number: 01
Introduction: Secure account creation and login.
Inputs: Email, password, optional SSO token.
Requirements Description: Must hash passwords; Email verification before full access; Rate-limit login attempts; lockout after N tries.
Outputs: Auth session; verification email sent.

Requirement number: 02
Use Case number: 02
Introduction: Store and update user profile data.
Inputs: name, state_code, school, preferences.
Requirements Description: Validate state_code against allowed list; Partial updates supported; Audit profile changes.
Outputs: Updated profile record.

Requirement number: 03
Use Case number: 03
Introduction: AI algorithm computes monthly savings plan and expected completion timeline.
Inputs: Goal type, target amount, time frame, income.
Requirements Description: Goal data must auto-update monthly; secure storage required.
Outputs: Goal plan and visual progress chart.

Requirement number: 04
Use Case number: 04
Introduction: Auto-categorize expenses, compute variance, generate graphs.
Inputs: User's transactions and budgets.
Requirements Description: Charts refresh daily; data stored in MySQL tables.
Outputs: Pie charts, trend lines, and overspending alerts.

Requirement number: 05
Use Case number: 05
Introduction: Identify repetitive mistakes and recommend preventive steps.
Inputs: User transaction history and payment logs.
Requirements Description: AI model must retrain weekly as new data arrives.
Outputs: Personalized alerts and reminder notifications.

Requirement number: 06
Use Case number: 06
Introduction: The system must automatically generate a monthly financial report summarizing the user's income, expenses, savings progress, and goal achievements.
Inputs: User's transaction history categorized spending data, and goal completion records.
Requirements Description: The system shall aggregate the user's financial data, compute monthly performance metrics, and highlight key insights such as total savings rate, expense distribution, and improvement areas.
Outputs: A downloadable, visualized monthly financial report (PDF/CSV) stored in the user's account history.

Requirement number: 07
Use Case number: 07
Introduction: The chatbot deliver personalized, AI-driven financial advice in real time.
Inputs: User queries, recent spending activity, and goal data.
Requirement Description: The AI Financial Advisor shall use natural language understanding and user profile analytics to generate tailored financial advice, such as saving strategies, spending control tips, or budget adjustments.
Outputs: Interactive chat response with personalized recommendations to the user.

Requirement number: 08
Use Case number: 08
Introduction: The system must track investment performance and provide AI-based portfolio recommendations.

Inputs: User-entered or imported investment data (amount, type, balance, source).
Requirements Description: The Investment Tracker shall calculate profit/loss, update portfolio value in real time, and use AI analytics to suggest diversification or risk reduction.
Outputs: Updated dashboard displaying portfolio performance and recommendations.

Requirement number: 09
Use Case number: 09
Introduction: AI analyzes patterns, generates personalized credit-improvement tips.
Inputs: User's credit score, payment history, and credit utilization.
Requirements Description: Credit data must refresh monthly; comply with privacy laws.
Outputs: Credit tips and milestone tracker.

Requirement number: 10
Use Case number: 10
Introduction: Retrieve state tax data, calculate estimated tax/refund.
Inputs: User income, location, deductions.
Requirements Description: Must update tax rules annually; support all U.S. states.
Outputs: Tax summary and filing suggestions.

Requirement number: 11
Use Case number: 11
Introduction: Retrieve content, track user progress, compute quiz results.
Inputs: Topic selection or keyword search.
Requirements Description: Content must be mobile-friendly; store user activity logs for insights.
Outputs: Progress badge and learning report.

Requirement number: 12
Use Case number: 12
Introduction: Importing transaction/account data from external institutions.
Inputs: OAuth token, secure aggregator token or CSV file upload.
Requirements Description: Use industry-standard aggregators (or OAuth) for account linking; validate CSV schema and provide mapping UI; token refresh handling; store only needed metadata and encrypted tokens.
Outputs: Imported transactions and account balances mapped to user accounts.

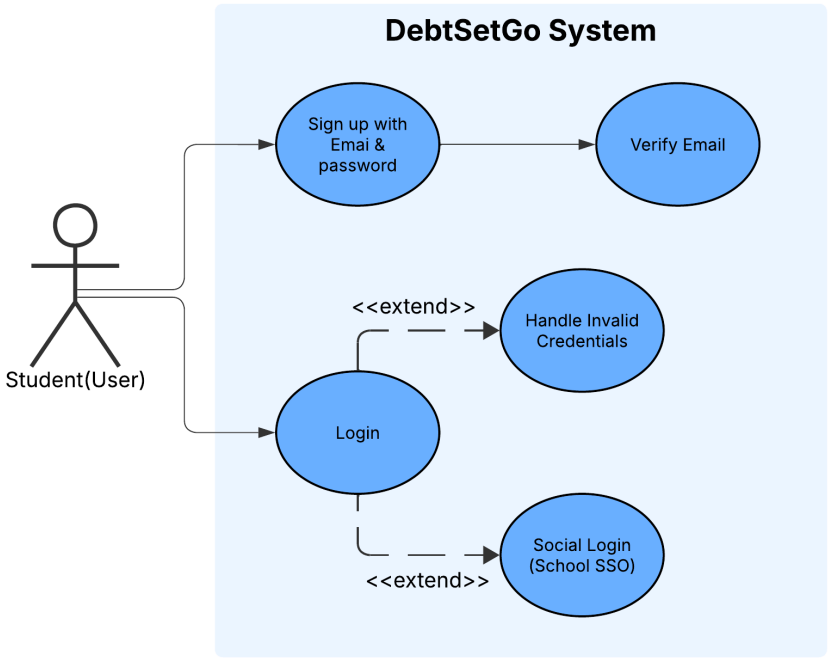
Requirement number: 13
Use Case number: 13
Introduction: Deliver push, email, and SMS messages for reminders, alerts, and marketing. Queue message, apply user preferences and throttling, send via provider.
Inputs: Notification trigger (due date, overspend, fraud flag), user contact preferences.
Requirements Description: Rate-limit outbound messages per user; comply with CAN-SPAM, TCPA and local privacy laws; support opt-out and preference center.
Outputs: Notification record, delivery receipt, retry logs.

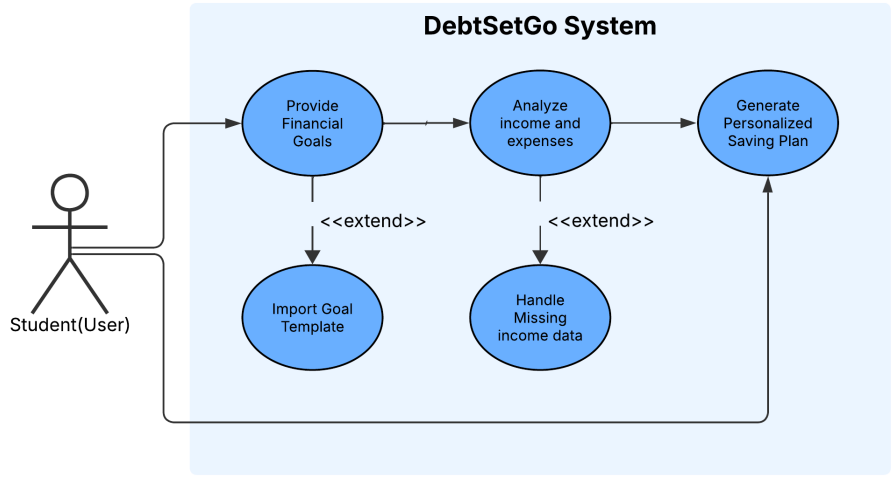
Requirement number: 14

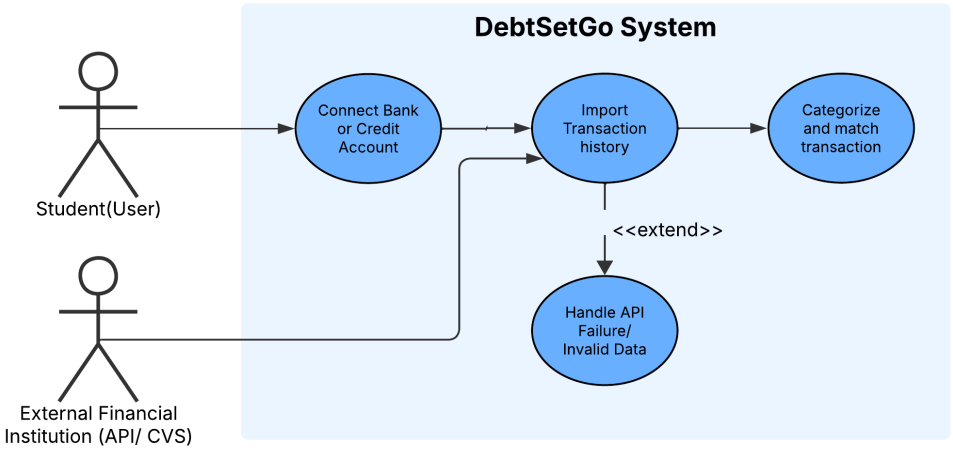
Use Case number: 14
Introduction: Detect suspicious activity and enable dispute resolution. Rule-based and ML scoring to flag anomalies, prompt user verification, allow dispute ticket creation routed to bank or support.
Inputs: Transaction feed, user behavioral baselines, device/location signals.
Requirements Description: Flagged incidents must be handled within SLA; store related logs for at least 12 months; protect PII and comply with applicable financial regulations.
Outputs: Flagged transaction status, dispute ticket, resolution timeline.

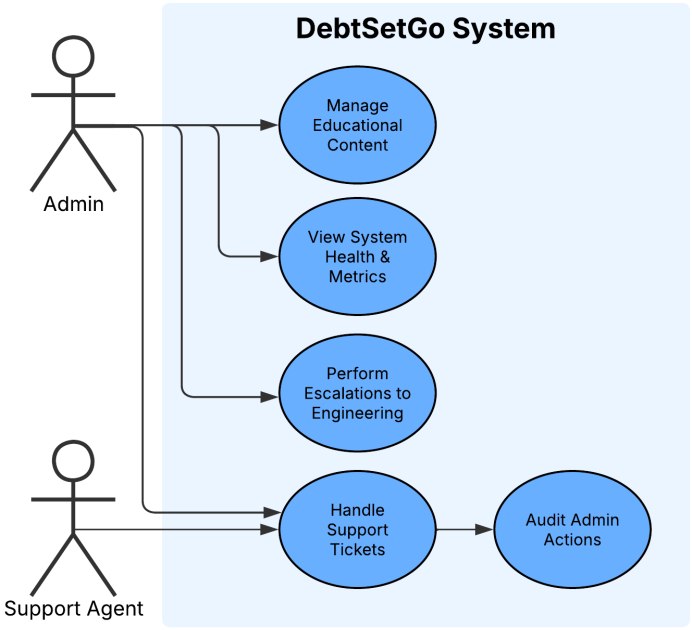
Requirement number: 15
Use Case number: 15
Introduction: Admin capability to manage platform safely and transparently. Enforce role-based access control (RBAC), require MFA for sensitive operations, and write every admin action to immutable audit logs.
Inputs: Admin actions (user suspension, content edits, report generation).
Requirements Description: Audit logs retained per policy; only super admins can change RBAC definitions; admin UI must mask PII unless explicitly authorized.
Outputs: Admin action logs, activity reports, and role change records.

2.3 Use Case Diagrams

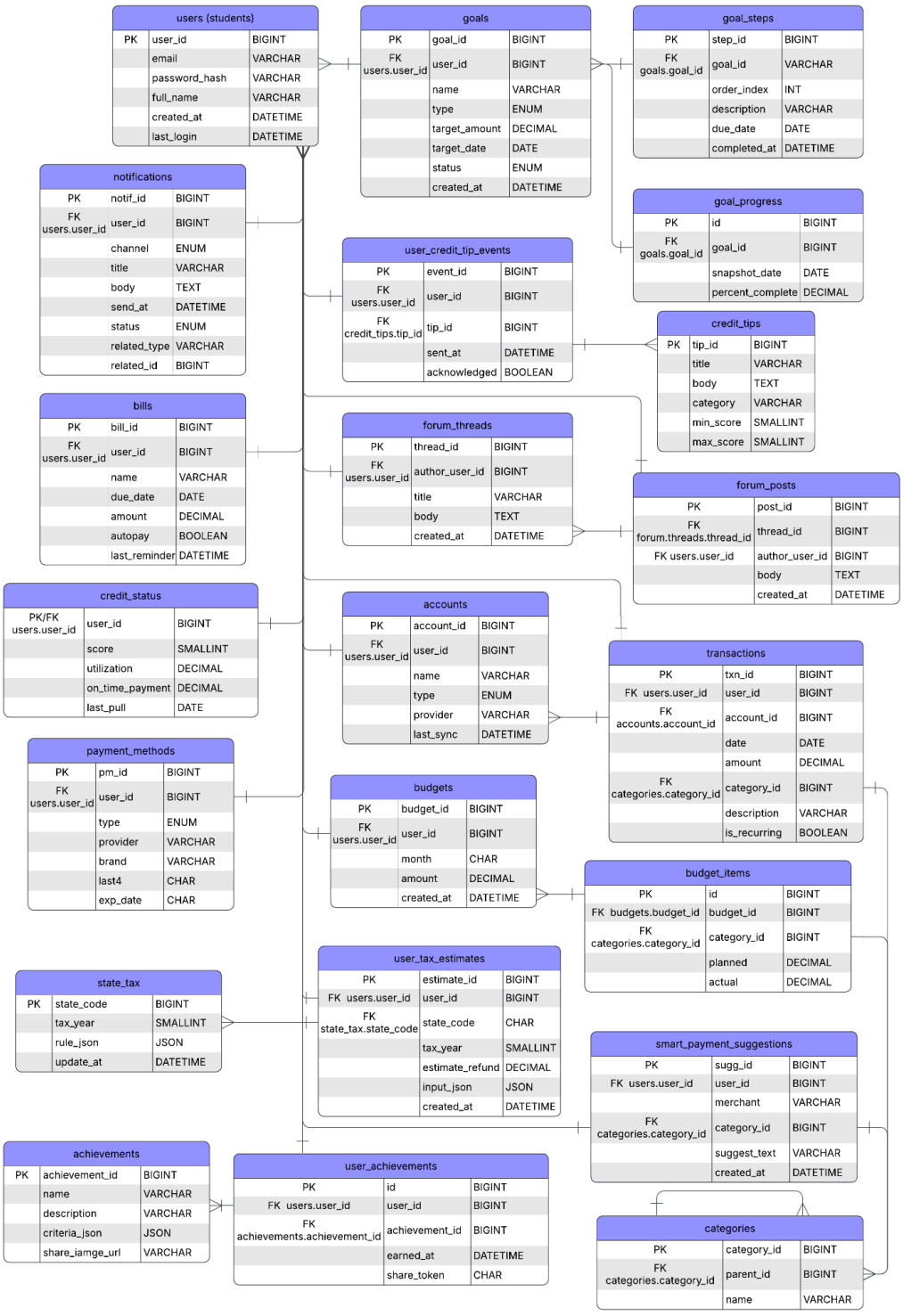



DebtSetGo System
Provide Financial Goals
Analyze income and expenses
Generate Personalized Saving Plan
<<extend>>
<<extend>>
Import Goal Template
Handle Missing income data
Student(User)


DebtSetGo System
Connect Bank or Credit Account
Import Transaction history
Categorize and match transaction
<<extend>>
Handle API Failure/ Invalid Data
Student(User)
External Financial Institution (API/ CVS)



3.0 DATABASE TABLES



We will use MySQL in our projects

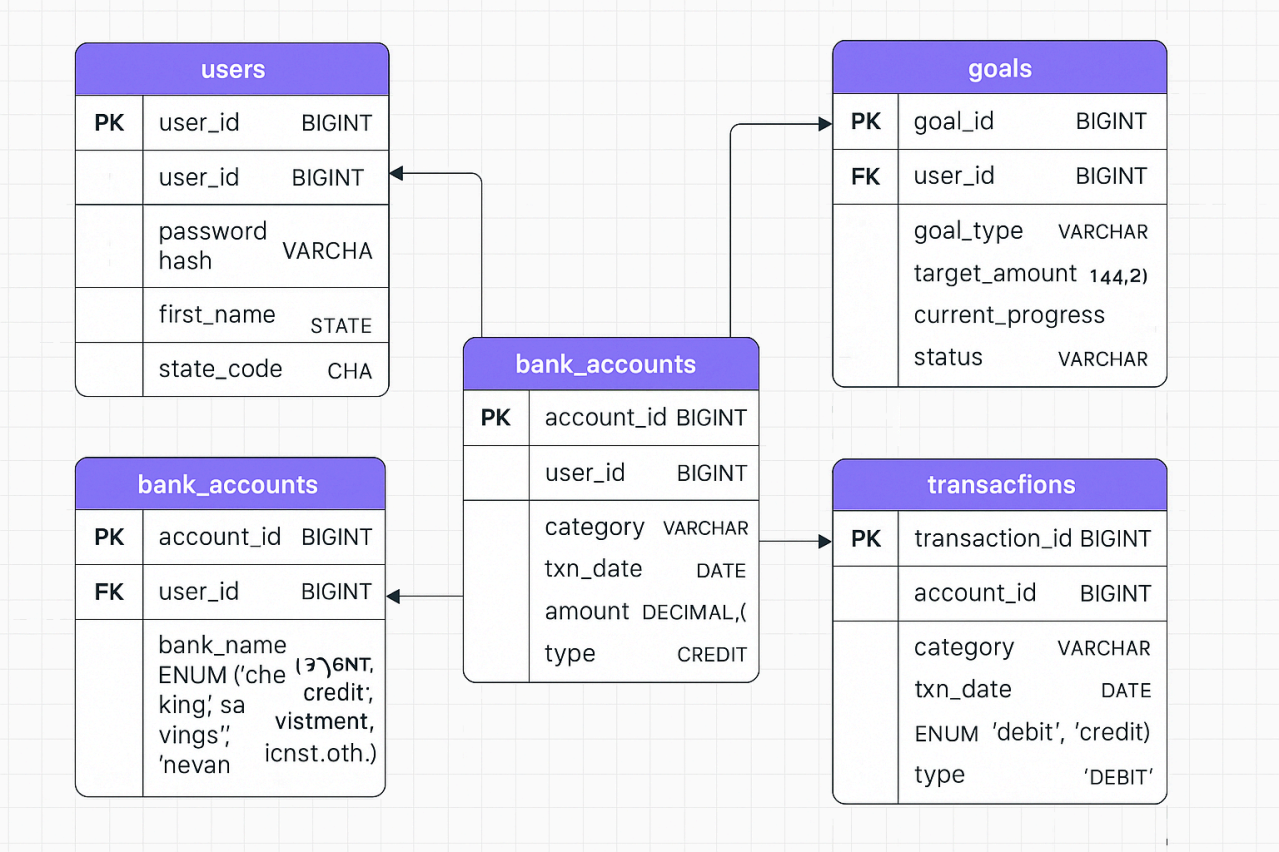


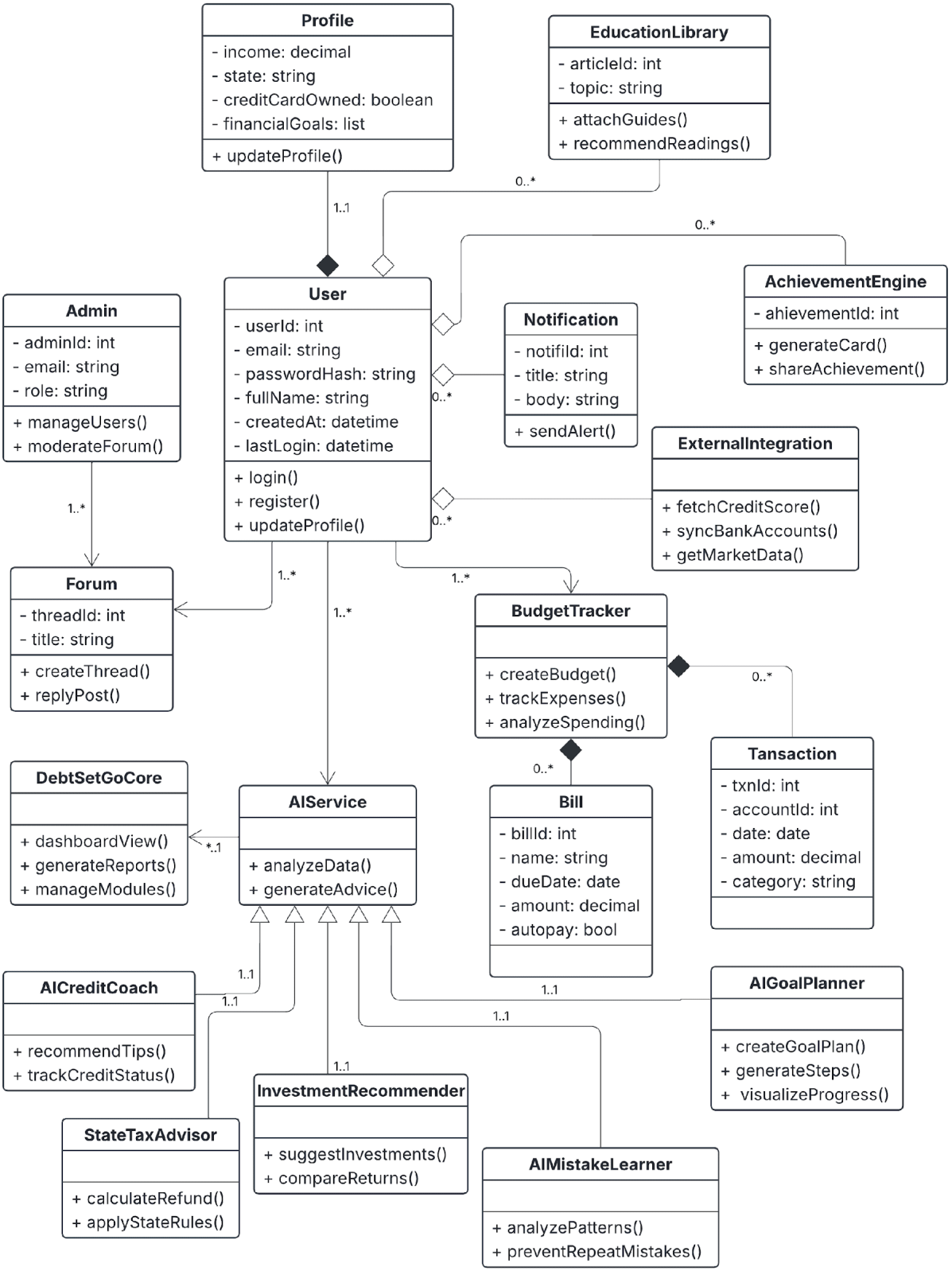
Figure 3.1: ER Diagram of DebtSetGo Database (MySQL Workbench – Sprint 3)

Table	Description	Key Fields
users	Stores login and basic profile information for each student user.	user_id (PK), email, password_hash, first_name, state_code
bank_accounts	Contains information about the user's connected financial accounts.	account_id (PK), user_id (FK), bank_name, account_type, balance
transactions	Records all income and expense data for each account.	transaction_id (PK), account_id (FK), category, txn_date, amount, type
goals	Saves user financial goals and progress.	goal_id (PK), user_id (FK), goal_type, target_amount, current_progress, status

These tables are related as follows:

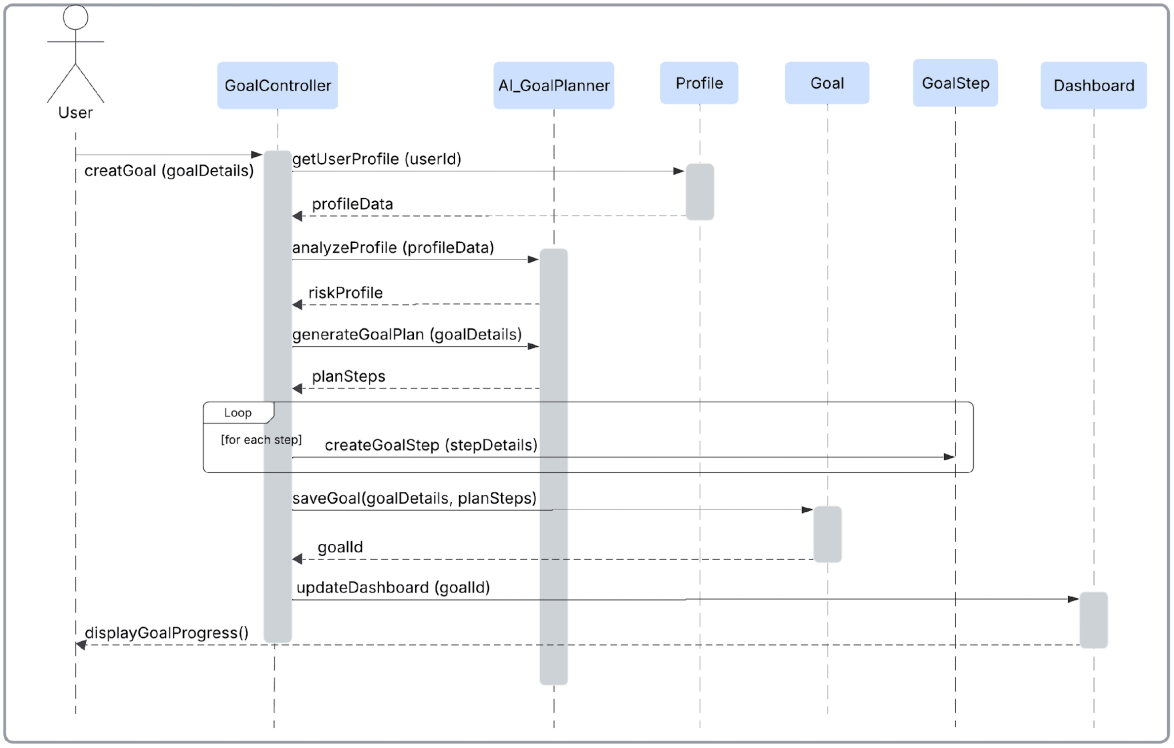
- One user can have **many bank accounts**.
- One bank account can have **many transactions**.
- One user can have **multiple goals**.

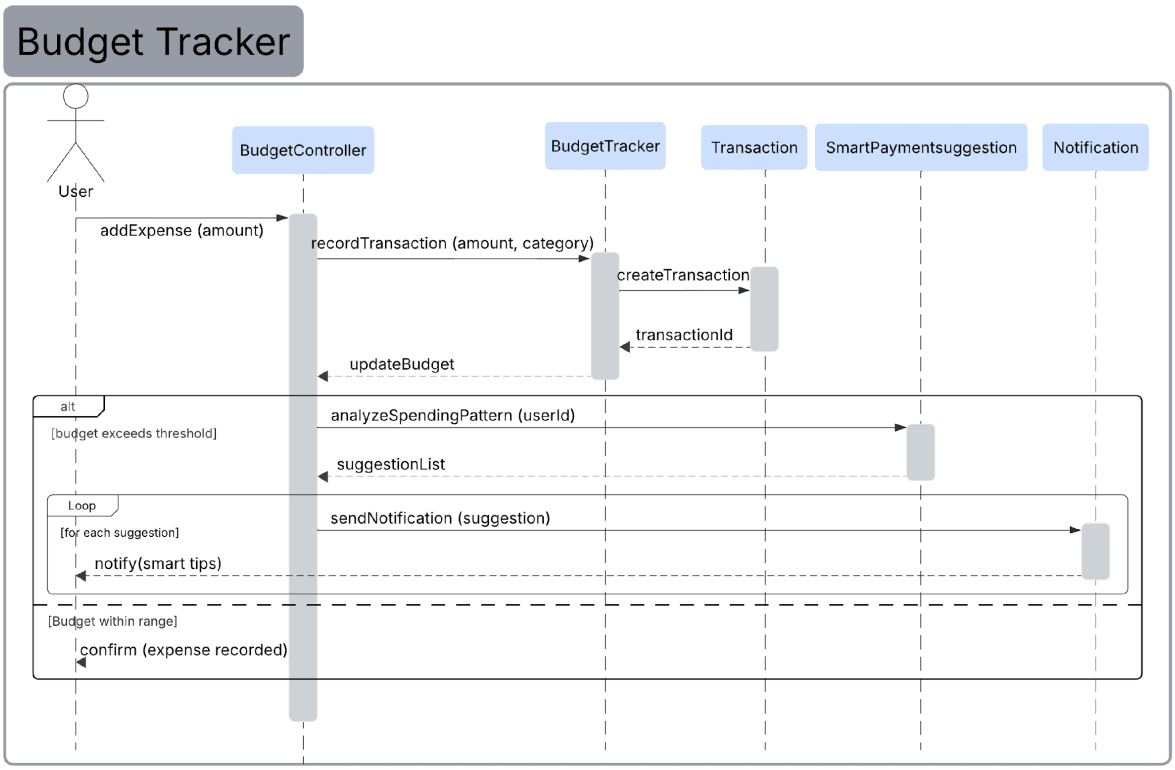
4.0 CLASS DIAGRAM


Profile
- income: decimal
- state: string
- creditCardOwned: boolean
- financialGoals: list
+ updateProfile()
EducationLibrary
- articleId: int
- topic: string
+ attachGuides()
+ recommendReadings()
0..*
1..1
0..*
AchievementEngine
- ahievementId: int
+ generateCard()
+ shareAchievement()
Admin
- adminId: int
- email: string
- role: string
+ manageUsers()
+ moderateForum()
User
- userId: int
- email: string
- passwordHash: string
- fullName: string
- createdAt: datetime
- lastLogin: datetime
+ login()
+ register()
+ updateProfile()
Notification
- notifiId: int
- title: string
- body: string
+ sendAlert()
0..*
0..*
ExternalIntegration
+ fetchCreditScore()
+ syncBankAccounts()
+ getMarketData()
1..*
0..*
Forum
- threadId: int
- title: string
+ createThread()
+ replyPost()
1..*
1..*
1..*
BudgetTracker
+ createBudget()
+ trackExpenses()
+ analyzeSpending()
0..*
Tansaction
- txnId: int
- accountId: int
- date: date
- amount: decimal
- category: string
DebtSetGoCore
+ dashboardView()
+ generateReports()
+ manageModules()
*..1
AIService
+ analyzeData()
+ generateAdvice()
0..*
Bill
- billId: int
- name: string
- dueDate: date
- amount: decimal
- autopay: bool
1..1
1..1
AICreditCoach
+ recommendTips()
+ trackCreditStatus()
1..1
1..1
AIGoalPlanner
+ createGoalPlan()
+ generateSteps()
+ visualizeProgress()
1..1
InvestmentRecommender
+ suggestInvestments()
+ compareReturns()
StateTaxAdvisor
+ calculateRefund()
+ applyStateRules()
AIMistakeLearner
+ analyzePatterns()
+ preventRepeatMistakes()

5.0 BEHAVIORAL MODELING

AI Goal Planner





6.0 IMPLEMENTATION

Database Implementation (MySQL) – Mohsin Hussain
For Sprint 3, I created the **DebtSetGo MySQL database** using four main tables: users, bank_accounts, transactions, and goals.
I executed the SQL script debtsetgo_sprint2_core_mysql.sql to create all tables and relationships.
After the tables were created, I ran several queries (SHOW TABLES, DESCRIBE, SELECT) to verify that the database structure was correct.
The database now supports the core functionality of user management, transaction tracking, and goal planning.

Frontend Implementation (React + Logic) – Jigyasa Jha

For Sprint 3, I developed the interactive web frontend for *DebtSetGo*, a personal-finance platform for college students. The interface allows users to enter income and expenses, see their total spending and balance, and view a category-wise spending chart. The goal was to make the site interactive, responsive, and easy to use without requiring a backend for demo purposes.

The frontend was implemented using **React (Vite)** with **HTML, CSS, and JavaScript.** Reusable components were created for the expense form, summary cards, expenses table, and category chart (using Chart.js). State is managed with React hooks, and data are persisted locally using localStorage so entries remain after refresh.

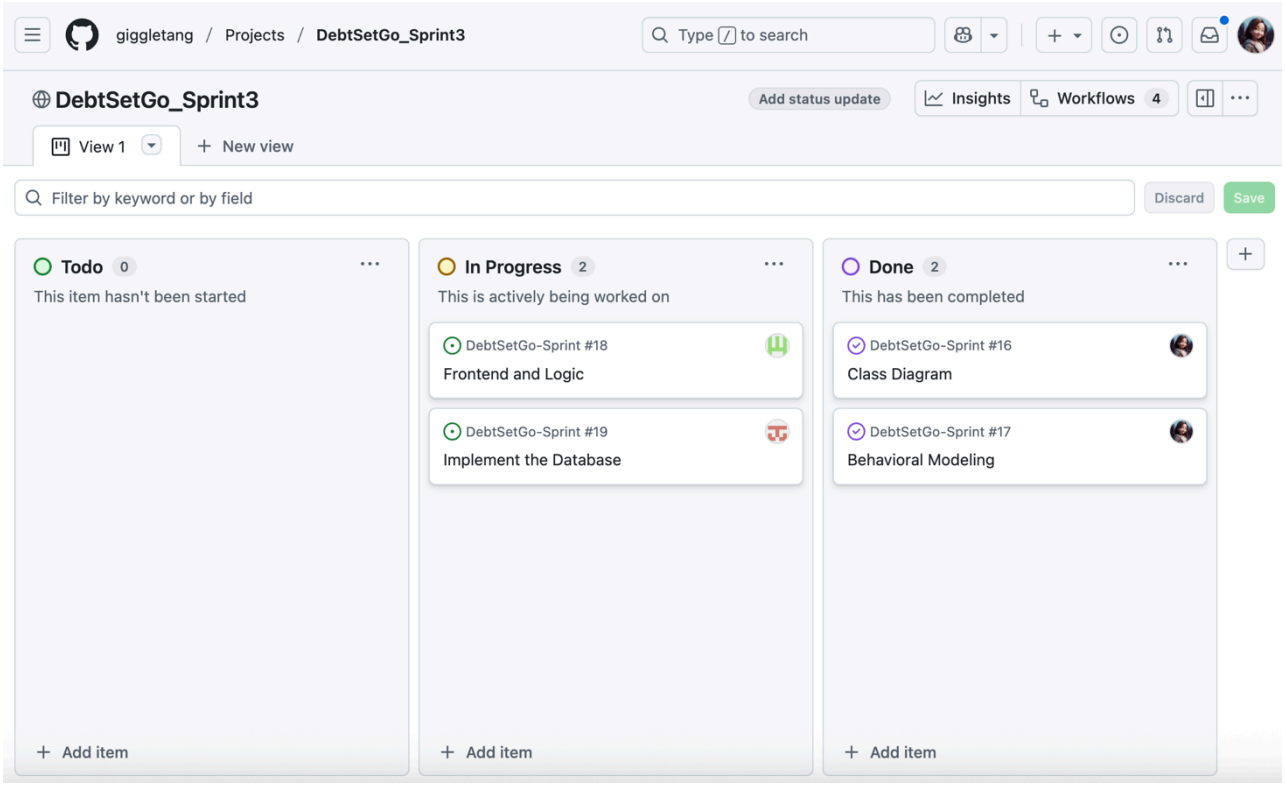
Steps to Run:

1. Install dependencies → npm install
2. Start the development server → npm run dev
3. Open the displayed local URL in a browser.

Testing and Verification:
– Validated form inputs and confirmed live updates to summary cards and charts.
– Tested data persistence via localStorage.
– Checked responsive layout on desktop and mobile screens.

7.0 GITHUB

https://github.com/giggletang/DebtSetGo-Sprint



References

- Bhatia, P., & Bhatia, R. (2021). Artificial intelligence in personal finance: Opportunities and challenges. International Journal of Computer Applications, 183(29), 12-18.

https://doi.org/10.5120/ijca2021921621

- National Endowment for Financial Education (NEFE). (2023). Financial Education Resources for College Students. Retrieved from https://www.nefe.org
- Mint by Intuit. (n.d.). Budget Tracker & Planner. Retrieved September 10, 2025, from https://mint.intuit.com