

**Nile University**

**School of Information Technology and Computer Science**

**Program of** Choose an item.

—Write the Title of the Project here——

Choose an item. **Senior Project I**

**Submitted in Partial Fulfilment of the Requirements**

**For the Bachelor’s Degree in Information Technology and Computer Science**

Choose an item.

**Submitted by**

—List your names and IDs here—

**Supervised by**

—Write your supervisors here—

**Giza – Egypt**

**Fall 2025**

**Project Summary**

Keywords:

Table of **Contents**

**No table of contents entries found.**

List of **Figures**

**No table of figures entries found.**

List of **Tables**

**No table of figures entries found.**

List of **Abbreviations**

**Chapter 1**

**Introduction**

* 1. **Background:**
* Provide context for the project by discussing the existing state of the field or problem.
* Explain why the project is important and relevant.
  1. **Motivation:**
* Describe the reasons and factors that motivated the selection of this particular project.
* Discuss any real-world problems or applications that the project aims to address.
  1. **Objectives:**
* Clearly state the specific goals and objectives of the project.
* Outline what the project intends to achieve.
  1. **Scope:**
* Define the boundaries and limitations of the project.
* Specify what is included and excluded from the project scope.
  1. **Significance of the Study:**
* Explain the potential impact or contributions of the project to the field.
* Discuss any potential benefits or applications.
  1. **Outline the structure of the report:**

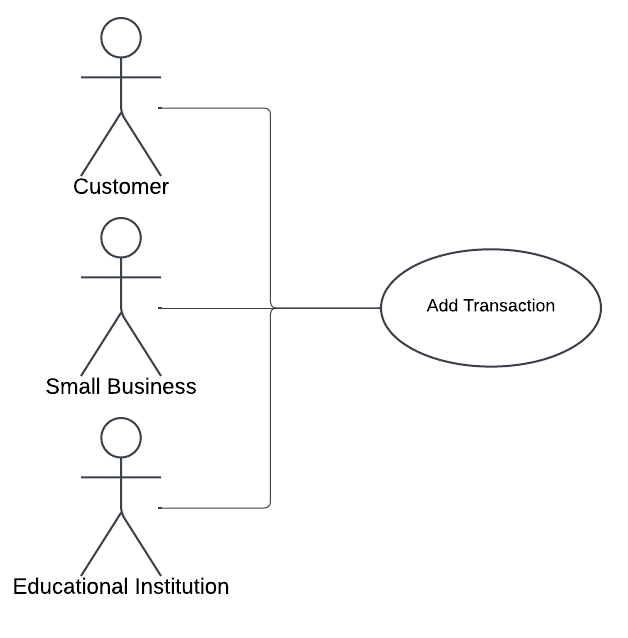
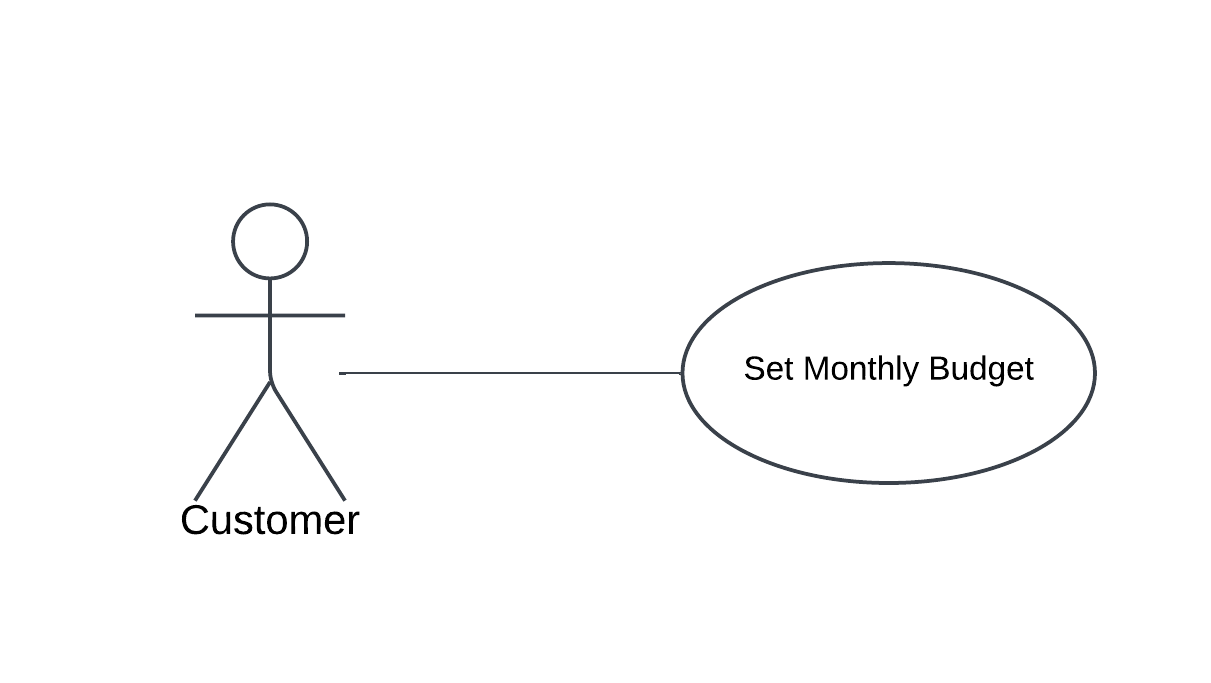
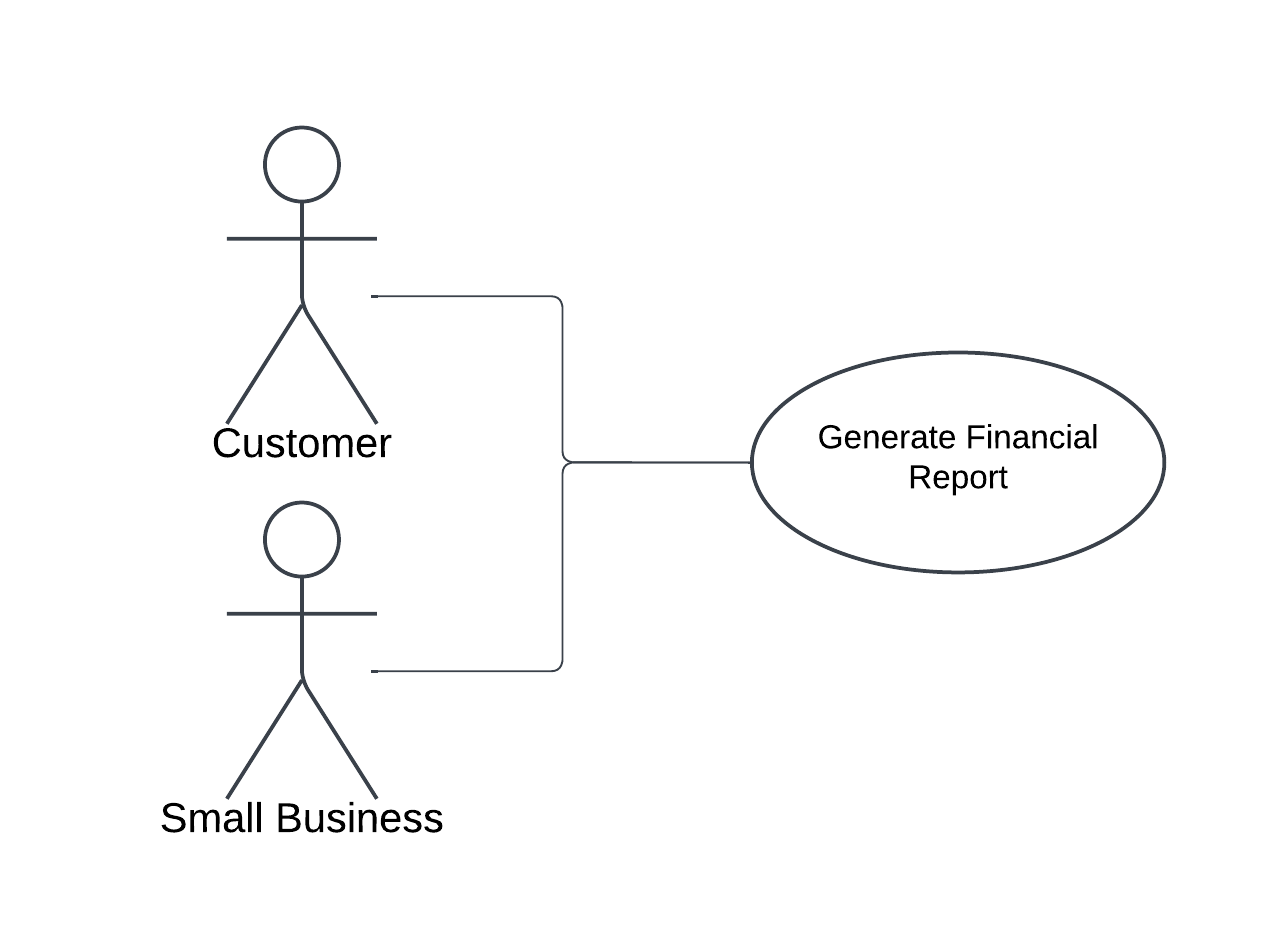
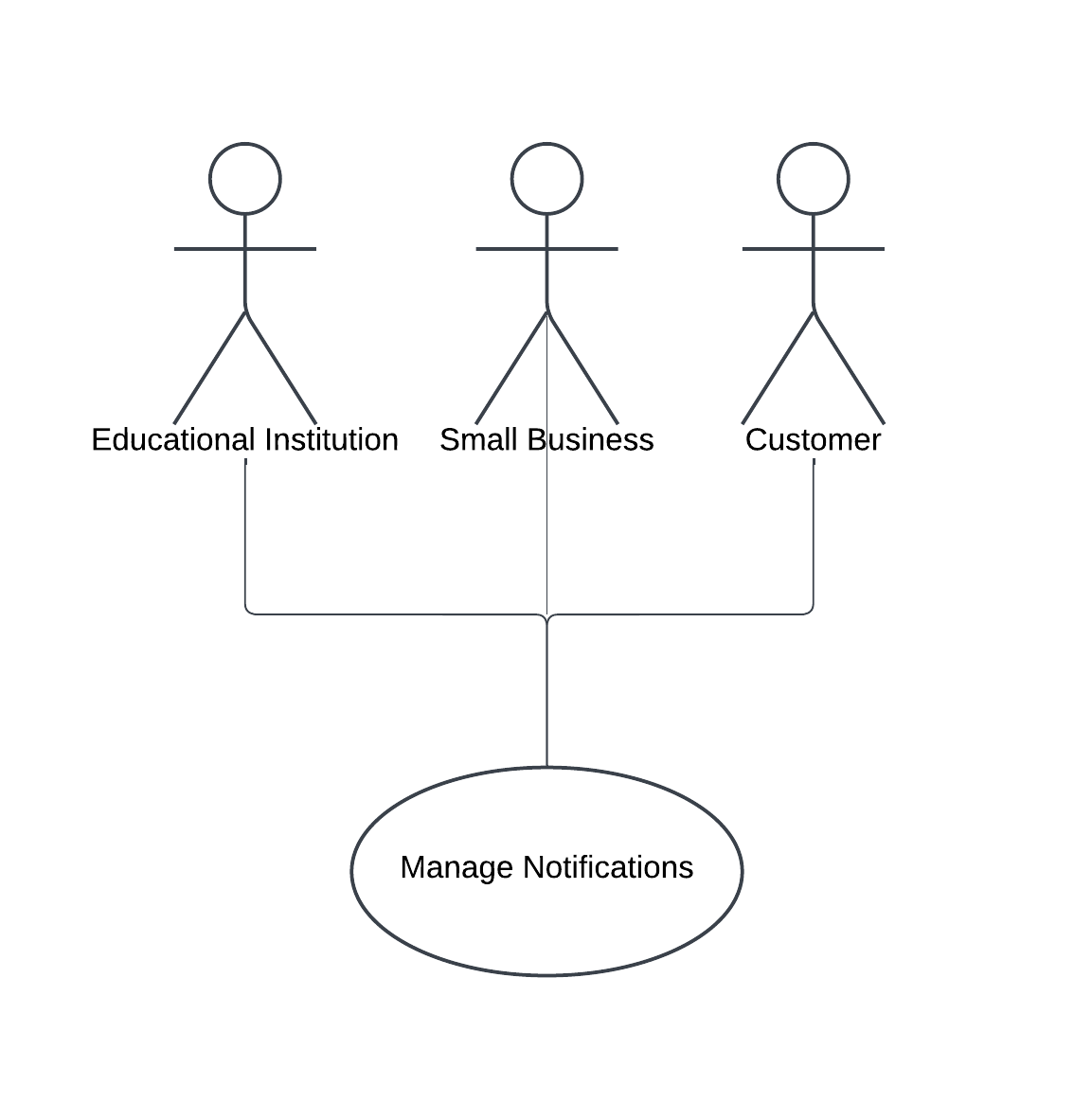
**Chapter 2**

**Related Work**

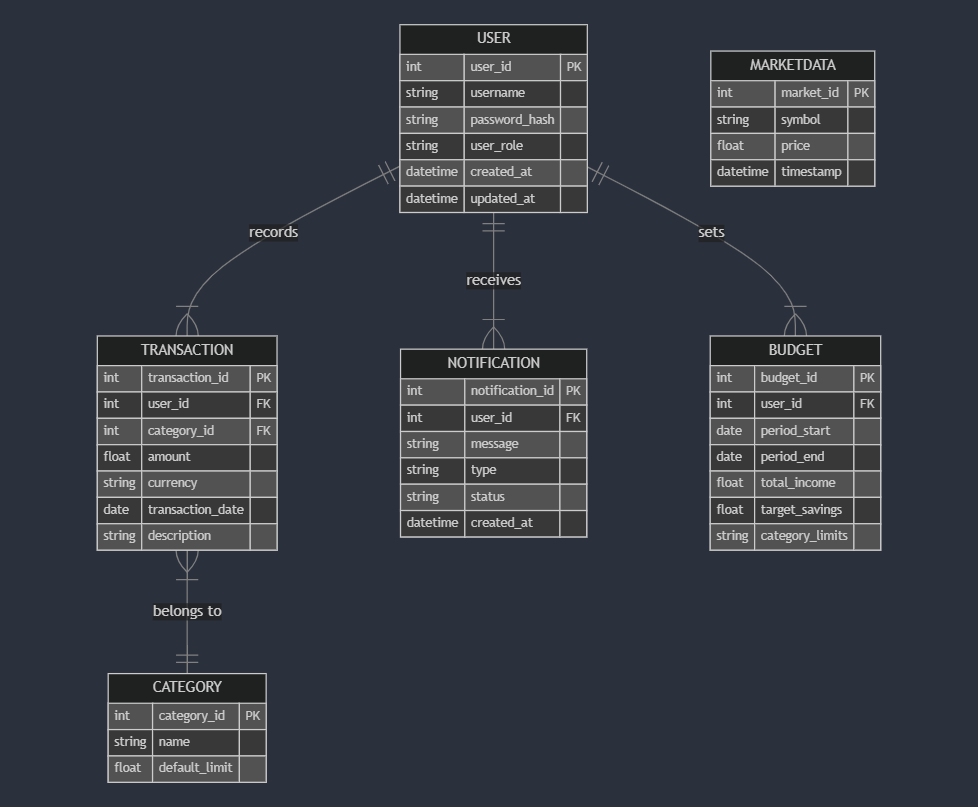
* 1. **Introduction to Literature Review:**
* Briefly explain the purpose of the literature review.
* Outline the structure of the literature review chapter.
  1. **Historical Perspective:**
* Discuss any historical developments or milestones related to the project.
* Provide context for the evolution of the project's subject matter.
  1. **Theoretical Framework:**
* Present theories or models relevant to the project.
* Discuss how these theories contribute to the understanding of the project.
  1. **Previous Research and Studies:**
* Summarize key findings from previous research related to the project.
* Identify gaps in the existing knowledge
  1. **Current State of the Field:**
* Discuss the current state of the field, including recent advancements or trends.
* Highlight any challenges or unresolved issues.

**Chapter 3**

**Materials and Methods**

* 1. **System Description:**
* FinFlow is a software system designed to help users (individuals, small businesses, educational institutions, and financial advisors) manage their finances more effectively. The application allows users to record daily expenses, set budgets and savings goals, and obtain insights or recommendations for better financial decision-making.
* **Stakeholders and Their Goals**
  + **Customers (Individuals)**
    - Track daily expenses (income & spending)
    - View and analyze personal financial summary
    - Receive investment or saving recommendations
  + **Small Businesses**
    - Monitor operational expenses and revenues
    - Generate basic financial statements and reports
    - Avoid high costs of enterprise-level financial tools
  + **Educational Institutions**
    - Use the app as a teaching tool in finance, business, or computer science courses
    - Illustrate practical application of data management, AI (for recommendations), and user experience design
  + **Financial Advisors**
    - Track and review client expenses to provide better advice
    - Recommend the tool as a supplementary method for clients to manage their personal or small business finances
* **External Interactions**
  + **Investment Market Data**: For providing real-time data and updates on investment opportunities and market changes.
  + **Authentication Services**: (e.g., OAuth, Single Sign-On) for external login or social login using Google, Apple, etc..
  + **Notification Services**: used to send alerts, reminders, and confirmations to users.
* **User Objectives/Requirements**
  + **Easy Data Entry and Management**
    - **Multiple Input Methods:** Users can add financial data through various channels (manual entry, scanning receipts, etc..).
    - **Categorization:** Automatic or user-defined labeling of expenses (e.g., groceries, utilities, entertainment, work(user-defined), etc..).
    - **Multi-Currency Support:** Allows users to work with multiple currencies within the application.
  + **Real-Time Dashboards and Visualizations**
    - **Clear Dashboard**: An intuitive dashboard that summarizes financial status, including total balance, expenses, income, and savings in a single glance.
    - **Goal Tracking**: Simple visual indicators showing progress toward savings goals or investment targets.
  + **Budgeting and Alerts**
    - **Monthly/Weekly Budget Setup**: Users can define spending caps and targets, with the ability to break them down by category or project.
    - **Spending Alerts**: Push notifications or emails when spending approaches or exceeds budget limits.
    - **Flexible Budgeting Periods**: Some users might want to track finances weekly, bi-weekly, or quarterly instead of monthly.
  + **Savings and Investment Guidance**
    - **Personalized Recommendations**: AI-driven or rules-based suggestions on which categories to cut back on or which investments to consider.
    - **Integration with Market Data**: Real-time updates for stocks, mutual funds, or other investment options.
  + **Small Businesses Features**
    - **Cash Flow Projections**: Automated forecasting based on current spending and income trends.
    - **Invoice and Payment Tracking**: Basic invoicing functionalities or integration with third-party invoicing platforms.
  + **Educational Tools**
    - **Demo Accounts**: Allow students to practice financial tracking with simulated data.
    - **Interactive Tutorials**: Step-by-step guides on budgeting, investing, or analyzing data.
    - **Course Integrations**: Options for instructors to set up class projects using the app (e.g., shared data sets, group work features).
  + **Notifications and Reminders**
    - **Bill Reminders**: Alerts for upcoming recurring expenses or bills.
    - **Milestone Notifications**: Updates when a user reaches a particular saving/investment goal.
    - **Custom Schedules**: Users choose frequency and method (push, email, text) of notifications.
  1. **System Requirements:**
* **Functional Requirements**
  + The user should be able to create an account and authenticate with secure credentials
  + The user should be able to add financial data using multiple methods
  + The system shall display a dashboard with up-to-date summaries of expenses, income, and savings goals
  + The system shall generate financial reports like monthly expenses, profit/loss statements
  + The system shall integrate with external market data APIs to offer basic investment or savings recommendations.
  + The system shall send notifications for nearing budget limits, bill reminders, etc.. via email, SMS, or push notifications
  + The system shall provide user roles (e.g., individual user, small business, financial advisor)
* **Non-Functional Requirements**
  + **Security**
    - All data must be encrypted at the database level and in transferring the data using HTTPS
  + **Performance**
    - Key dashboard pages should load within 2 seconds under normal usage.
  + **Portability**
    - The system shall run on major operating systems like Android and IOS
* **Use Cases**
  + **UC1: Add Daily Transaction**
    - **Actors**: Customer, Small Business, Educational Institution
    - **Description**: Actors record income or expenses, either manually or via integrated bank/payment APIs.
  + **UC2: Set Monthly Budget** 
    - **Actors**: Customer
    - **Description**: Individual users specify budget limits and savings goals for a given period.
  + **UC3: Generate Financial Report** 
    - **Actors**: Small Business, Customer
    - **Description**: Small businesses and individuals create detailed statements (monthly expenses, profit/loss).
  + **UC4: Manage Notifications**
    - **Actors**: Customer, Small Business, Educational Institution
    - **Description**: All roles may receive or configure alerts related to spending limits, upcoming bills, or system updates.
  + **UC5: Manage User Accounts**
    - **Actors:** Customer, Small Business, Financial Advisor, Educational Institution
    - **Description:** Basic account operations (registration, login, profile management). In some systems, certain administrative tasks might be restricted to an admin role.
* **Software Interfaces**
  + **Database Interface**
    - **Technology:** Relational database (e.g., MySQL or PostgreSQL)
    - **Access Method:** SQL queries
    - **Data Storage:** User profile information, transactions, budget settings, notifications.
    - **Verification:** Database queries, stored procedures, integrity constraints.
  + **External Market Data APIs**
    - **Input:** Current market data for stocks, funds, or other investments.
    - **Output:** HTTP GET requests for relevant financial instruments.
    - **Protocol:** RESTful endpoints with JSON responses.
    - **Verification:** Rate-limiting checks, API key usage.
  + **User Interface (UI)**
    - **Input:** Touch gestures (mobile)
    - **Output:** Graphical dashboards, charts, forms, and reports.
    - **Verification:** UI testing (usability, functionality, accessibility).
  + **Notification Services**
    - **Channels:** Push notifications.
    - **Verification:** Track delivery status, handle errors (e.g., bounced emails).
* **Hardware Interfaces**
  + **Mobile Device Hardware**
    - **Camera**: For scanning receipts.
    - **Biometric Sensors**: For secure login (fingerprint, facial recognition).
    - **Verification**: Ensuring the app has permission to use the camera or biometric sensors
* **Network Interfaces**
  + **Protocol**:
    - The application shall use HTTPS for all data transmissions to ensure confidentiality and integrity.
    - WebSocket connections or SSE (Server-Sent Events) may be used for real-time dashboards and notifications.
  + **Ports**:
    - Port **443** for secure HTTPS.
  + **Firewall and Security**:
    - The system must comply with organizational/network security policies, ensuring that inbound and outbound traffic is restricted to necessary endpoints only.
  1. **Design Constraints:**

The application’s design must be visually appealing and easy to navigate to motivate users and decrease resistance and overhead for the user to input his/her daily transactions.

* 1. **Research Design:**
* **Requirements Gathering**: We began by collecting user needs and system requirements from stakeholders (individual users, small businesses, educational institutions, financial advisors).
* **Design & Prototyping**: We created an initial prototype for the user interface.
* **Implementation**: We’re developing the application incrementally, focusing on core features like expense tracking and budgeting, followed by more advanced modules like investment advice and AI-driven recommendations.
* **Testing & Evaluation**: Then we’re going to conduct iterative testing like unit tests, integration tests, and user acceptance testing on our close network to ensure functionality, usability, and reliability.
* **Deployment & Review**: Finally, deploying the final version of the application for real-world use and gathering feedback for future improvements.
  1. **Data Design (Entities and Relationships):**
* **User**
  + **user\_id (PK):** Unique identifier for each user.
  + **username, password\_hash:** Login credentials.
  + **user\_role:** Role within the system (e.g., “individual,” “small\_business,” “advisor,” “institution”).
  + **Relationships:**
    - One User can have many Transactions.
    - One User can have one or many Budgets (depending on whether you allow multiple budgets per user).
    - One User can have many Notifications.
* **Transaction**
  + **transaction\_id (PK):** Unique identifier for each transaction.
  + **user\_id (FK):** Points to the User who created/owns the transaction.
  + **category\_id (FK):** Points to a Category, such as “Food,” “Rent,” etc.
  + **amount, currency, transaction\_date, description:** Core details of each financial record.
  + **Relationships:**
    - Many Transactions belong to one User.
    - Many Transactions can be linked to one Category.
* **Category**
  + **category\_id (PK):** Unique identifier for each category (e.g., “Groceries,” “Utilities”).
  + **name:** Human-readable category label.
  + **default\_limit (optional):** A default budget limit for that category (if applicable).
  + **Relationships:**
    - One Category can be associated with many Transactions.
* **Budget**
  + **budget\_id (PK):** Unique identifier for each budget.
  + **user\_id (FK):** Points to the User who owns this budget.
  + **period\_start, period\_end:** Defines the date range for the budget.
  + **total\_income, target\_savings:** High-level user financial goals for that period.
  + **category\_limits (optional):** stored in a separate table or JSON field to map individual categories to spending limits.
  + **Relationships:**
    - One User can have one or multiple Budgets (project-specific decision).
    - A Budget typically references the same categories used in Transactions.
* **Notification**
  + **notification\_id (PK):** Unique identifier for each notification.
  + **user\_id (FK):** Points to the User who receives the notification.
  + **message, type, status, created\_at:** Details about the notification content and state (e.g., read/unread).
  + **Relationships:**
    - Many Notifications belong to one User.
* **Market Data**
  + **market\_id (PK):** Unique identifier for each market data record.
  + **symbol, price, timestamp:** Basic fields for tracking stocks, cryptocurrencies, or other financial instruments.
  + **Relationships:**
    - no direct foreign key to User
    - The application periodically fetches and store market data for analytics or investment suggestions.
  1. **Interaction Design (if applicable):**
* **Login and Registration**
  + **Process**: Users enter credentials (username/password or social login).
  + **Design Choice**:
    - Clean, minimalistic login screen.
    - Visual cues for password requirements (strength meter, checklist).
  + **Feedback**: Immediate error messages for invalid credentials; success message upon login.
* **Main Dashboard**
  + **Process**: After logging in, users land on a dashboard that summarizes key financial data (account balances, budgets, recent transactions).
  + **Design Choice**:
    - Card-based layout to display different financial metrics.
    - Color-coded status indicators and categories
  + **Feedback**: Real-time updates as data is added or modified.
* **Transactions**
  + **Process**: Add, edit, and view transaction details.
  + **Design Choice**:
    - Wizard-style form for adding a transaction step by step (date, category, amount).
    - Inline editing for quick updates.
  + **Feedback**: Confirmation dialog before saving (verification), success/failure messages post-save.
* **Budget Management**
  + **Process**: Users create or adjust monthly/weekly budgets per category.
  + **Design Choice**:
    - Progress bars or circular graphs indicate percentage spent.
    - Auto-calculation of remaining budget based on real-time transaction data.
  + **Feedback**: Alerts when nearing budget limit (“You’ve spent 80% of your Food budget!”).
* **Reports**
  + **Process**: Generate financial reports (monthly expenditure, budget remaining, etc.)
  + **Design Choice**:
    - Tabular data with the option to export (PDF, CSV).
    - Interactive charts (clickable segments in a pie chart to drill down)
  + **Feedback**: Loading indicator while generating reports, success message when the report is ready.
* **Investment/Advisor Mode**
  + **Process**: Users or financial advisors explore investment options or review advice.
  + **Design Choice**:
    - Recommendation cards listing potential investments with risk levels.
    - Overall ratings for investment options
  + **Feedback**: Real-time market data updates.
* **Notification Center**
  + **Process**: Users receive notifications for approaching budget limits or new advisor messages.
  + **Design Choice**:
    - Bell icon in the top navigation.
    - Badges showing unread messages/alerts.
  + **Feedback**: Clicking on a notification displays details; dismiss or snooze notifications as needed.
  1. **Data Flow:**
* **Input Stage**: User inputs data using one of the methods provided like manual transaction entries or monthly budget setup. External APIs like market data feed the system with raw data.
* **Processing Stage**: The system’s sub-processes (Transactions, Budgets, Reports, etc.) validate, transform, and store data.
* **Output Stage**: The application provides dashboards, reports, and alerts/notifications to end-users. Reports can be viewed in-app or exported (PDF, CSV), while alerts are sent via push notifications.
  1. **Integration with External Systems:**

To provide automatic transaction imports, real-time market data, and extended functionality (notifications, identity management), FinFlow integrates with various external systems. These include:

* + **Market Data Services:** for investment and financial market information.
  + **Notification Providers:** for sending emails, SMS, or push notifications.
  + **Authentication Services:** for secure user logins.
* **Market Data Integration**
  + **Purpose**
    - Fetch real-time or historical prices for stocks, mutual funds, investing options.
    - Provide users with up-to-date information for investment decision-making or portfolio tracking.
  + **Data Exchange Format**
    - JSON responses.
  + **Protocols and Security**
    - API Keys provided by the market data vendor, included in request headers or query parameters.
    - HTTPS for secure connections, ensuring integrity of market data.
  + **Workflow**
    - **API Request**: The system sends a GET request with the valid API key.
    - **Response Parsing**: The system receives and parses JSON or XML data (current price, 24h change, etc.).
    - **Caching and Storage**: Frequently used data are cached to reduce repeated API calls.
    - **Display and Analytics**: Users see market updates on dashboards or in the investment advice module.
* **Notification Services**
  + **Purpose**
    - Send out alerts and notifications (budget overspending alerts, transaction confirmations, investment updates).
  + **Data Exchange Format**
    - **Push Notifications**: Platform-specific APIs (Firebase Cloud Messaging, Apple Push Notification service).
  + **Protocols and Security**
    - REST with HTTPS for push services.
    - API Keys or OAuth tokens to authenticate with notification providers.
  + **Workflow**
    - **Trigger**: A relevant event occurs (user’s expense crosses 80% of the budget limit).
    - **API Call**: The system formats a message and sends it to the notification provider.
    - **Delivery**: The provider handles the actual sending of push notifications to the user.
    - **Logging**: The system records the notification in its database for auditing purposes

**Chapter 4**

**Implementation and Preliminary  
Results**

* 1. **Programming Languages and Tools:**
* Specify the programming languages, frameworks, and tools used for implementation.
* Justify the choice of these technologies.
  1. **Code Structure:**
* Provide an overview of the structure of the codebase.
* Explain how the code is organized and modularized.
  1. **Data Structures and Databases:**
* Discuss the data structures used in the project.
* Detail the database schema and data storage mechanisms.
  1. **Quantitative Results:**
* Present numerical data, statistics, or quantitative findings.
* Use tables, graphs, or charts for clarity.
  1. **Qualitative Results:**
* Discuss any qualitative findings, observations, or insights.
* Include quotes or examples where relevant.

.

**Chapter 5**

**Discussion and Conclusion**

* 1. **Interpretation of Results:**
* Analyze the results in relation to the project objectives.
* Discuss any patterns, trends, or correlations.
  1. **Comparison with Previous Studies:**
* Compare your findings with those of previous research.
* Highlight similarities, differences, or advancements.
  1. **Limitations:**
* Acknowledge any limitations or constraints encountered during the project.
* Discuss how these limitations may have affected the results.
  1. **Summary of Findings:**
* Summarize the main findings of the project.
* Provide a concise overview of the project's accomplishments.
  1. **Future Work:**
* Suggest areas for future research or improvements.
* Discuss potential extensions or developments of the project.

**References**

**Appendices**