National University
Of Computer and Emerging Sciences

Project Deliverable 03

In partial fulfillment
of the requirement for the course of

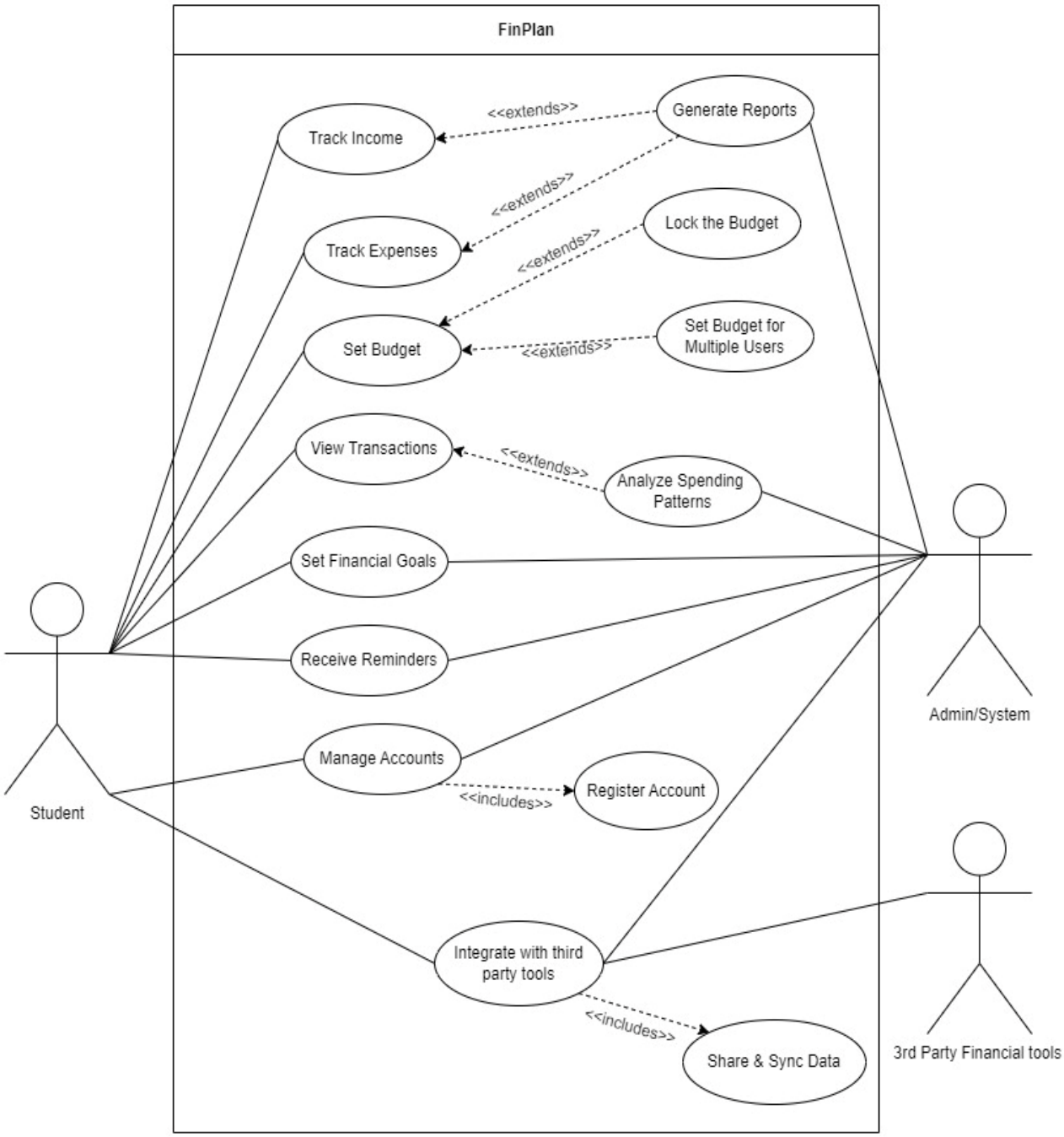
Software Design and Analysis

By

Raima Tariq | 22i-1145
Rimsha Azam | 22i-1129
Umama Bajwa | 22i-1133

BSCS-C

USE CASE DIAGRAM



USE CASE 1: TRACK INCOME (RIMSHA AZAM)

Use Case Name:	Track Income	
Scope:	FinPlan – A Personal Finance and Budget Management System	
Level:	User Goal	
Primary Actor:	Student	
Stakeholders and Interests:	• **Student**: Wants to track income sources and ensure accurate financial management. • **System**: Needs to store income records securely and provide easy access for future reference	
Pre-conditions:	• The student must be logged into the system. • The system must have access to store financial data.	
Post-conditions:	• The income data is saved and available for future reference. • The system updates the overall financial status of the student.	
Main Success Scenario:	**Actor Action:**	**System Response:**
	1. The student navigates to the "Track Income" section. **2.** The student inputs income details (e.g., source, amount, date).	**3.** The system validates the input data. **4.** The system records the income data in the student's profile. **5.** The system confirms that the income has been successfully added.

Extensions:	• **3a: Invalid Input**: If any of the fields (amount or date) are incorrect or missing, the system prompts the student to correct them. • **4a: Data Storage Failure**: If the system fails to store the data, it notifies the student and suggests retrying or contacting support.

USE CASE 2: TRACK EXPENSES (RIMSHA AZAM)

Use Case Name:	Track Expenses	
Scope:	FinPlan – A Personal Finance and Budget Management System	
Level:	User Goal	
Primary Actor:	Student	
Stakeholders and Interests:	• **Student**: Wants to categorize and track their spending to manage finances better. • **System**: Needs to accurately store expenses and categorize them for budget analysis.	
Pre-conditions:	• The student must be logged into the system. • The system must be able to categorize expenses.	
Post-conditions:	• The expense data is saved and categorized. • The system updates the budget and spending reports.	
Main Success Scenario:	**Actor Action:**	**System Response:**

	1. The student navigates to the "Track Expenses" section. **2.** The student selects a category (e.g., food, transport) and enters the expense details (e.g., amount, date).	**3.** The system validates the entered data. **4.** The system records the expense and updates the corresponding budget category. **5.** The system confirms the expense has been successfully added.
Extensions:	• **3a: Invalid Input**: If any required data is missing or incorrect, the system requests corrections. • **4a: Exceeded Budget** : If adding the expense exceeds the category budget, the system warns the student.	

USE CASE 3: SET BUDGET (RIMSHA AZAM)

Use Case Name:	Set Budget
Scope:	FinPlan – A Personal Finance and Budget Management System
Level:	User Goal
Primary Actor:	Student
Stakeholders and Interests:	**Student**: Wants to set limits on spending to manage finances effectively. **System**: Needs to enforce the set budget during expense tracking.
Pre-conditions:	• The student must be logged into the system. • The student should have set up categories for expenses.

Post-conditions:	● The budget is saved for each category. ● The system tracks the budget against the student's actual expenses.	
Main Success Scenario:	**Actor Action:**	**System Response:**
	1. The student navigates to the "Set Budget" section. **2.** The student selects categories (e.g., food, transport) and assigns budget limits to each.	**3.** The system validates the budget limits. **4.** The system records the budgets for the selected categories. **5.** The system confirms that the budgets have been successfully set.
Extensions:	● **3a: Invalid Input**: If a budget limit is unreasonable (e.g., zero or excessively high), the system prompts the student for a valid input. ● **4a: Existing Budget:** If a budget already exists for a category, the system asks if the student wants to overwrite it.	

USE CASE 4: VIEW TRANSACTIONS (RIMSHA AZAM)

Use Case Name:	View Transactions
Scope:	FinPlan – A Personal Finance and Budget Management System
Level:	User Goal
Primary Actor:	Student

Stakeholders and Interests:	**Student**: Wants to view a list of past transactions to review their financial activity. **System**: Needs to retrieve and display transaction data accurately.	
Pre-conditions:	● The student must be logged into the system. ● The system must have stored transaction data.	
Post-conditions:	The student can view, sort, and filter past transactions.	
Main Success Scenario:	**Actor Action:**	**System Response:**
	1. The student navigates to the "View Transactions" section. **3.** The student can filter transactions by category, amount, or date. **5.** The student reviews their transaction history.	**2.** The system displays a list of all transactions, sorted by date. **4.** The system displays the filtered transactions as requested.
Extensions:	● **2a: No Transactions Found**: If no transactions exist, the system informs the student that no data is available. ● **4a: Incorrect Filter** : If the filters yield no results, the system suggests adjusting the filters.	

USE CASE 5: SET FINANCIAL GOALS (RIMSHA AZAM)

Use Case Name:	Set Financial Goals
Scope:	FinPlan – A Personal Finance and Budget Management System
Level:	User Goal

Primary Actor:	Student	
Stakeholders and Interests:	• **Student**: Wants to set specific savings or debt repayment goals to stay on track financially. • **System**: Needs to monitor progress and notify the student of goal milestones.	
Pre-conditions:	• The student must be logged into the system. • The student should have income and expense data available.	
Post-conditions:	The financial goal is saved and progress is tracked over time.	
Main Success Scenario:	**Actor Action:**	**System Response:**
	1. The student navigates to the "Set Financial Goals" section. **2.** The student enters the goal details (e.g., goal type, target amount, deadline).	**3.** The system validates the goal details. **4.** The system records the goal and begins tracking progress. **5.** The system confirms that the goal has been successfully set.
Extensions:	• **3a: Invalid Input**: If any goal detail is missing or incorrect (e.g., deadline in the past), the system requests corrections. • **4a: Unrealistic Goal** : If the goal is too ambitious based on the student's income/expense patterns, the system suggests revising the goal.	

USE CASE 6: Generate Reports (RAIMA TARIQ)

Use Case Name:	Generate Reports	
Scope:	FinPlan – A Personal Finance and Budget Management System	
Level:	User Goal	
Primary Actor:	Student User	
Stakeholders and Interests:	● **Student:** Needs to visualize their financial data to track expenses, income, and savings, aiding in financial decisions. ● **System Administrator:** Ensures smooth, error-free operation of the system, including report generation.	
Pre-conditions:	● The user must be logged into the FinPlan system. ● The system must have recorded data on the user's income, expenses, and budgets.	
Post-conditions:	● The requested report is generated and presented to the user. ● The report is optionally saved or exported in a specified format.	
Main Success Scenario:	**Actor Action:**	**System Response:**
	1. The user selects "Generate Reports." **3.** The user selects a report type and inputs the date range. **5.** The user views the report.	**2.** The system displays available report types. **4.** The system processes the request and generates the report. **6.** The system provides options to save or export the report.

Extensions:	• **2a.** If no report type is selected, the system prompts the user to choose one. • **4a.** If no data is available for the selected date range, the system informs the user and requests a new range. • **6a.** If there's an error during saving or exporting, the system notifies the user with instructions to resolve the issue.

USE CASE 7: Receive Reminders (RAIMA TARIQ)

Use Case Name:	Receive Reminders	
Scope:	FinPlan – A Personal Finance and Budget Management System	
Level:	User goal	
Primary Actor:	Student User	
Stakeholders and Interests:	• **Student:** Wants to stay informed about upcoming financial obligations (e.g., bills, subscriptions) to avoid missed payments. • **System Administrator:** Ensures the timely and accurate sending of reminders.	
Pre-conditions:	• The user is logged in. • The system has recorded due dates for bills, subscriptions, or financial commitments.	
Post-conditions:	• The user receives a reminder notification. • The system logs that the reminder was sent.	
Main Success Scenario:	**Actor Action:**	**System Response:**

	1. The user enters financial commitments (e.g., bills) with due dates. **3.** The system sends a notification to the user for upcoming due dates (e.g., within 7 days). **5.** The user opens the notification and views the details (e.g., bill type, due date, amount). **7.** The user decides to act on the reminder (e.g., mark as paid, snooze, or dismiss).	**2.** The system stores the commitments and schedules reminders based on the due dates. **4.** The user receives the reminder notification. **6.** The system displays the reminder details to the user. **8.** The system updates the status based on the user's choice (e.g., stops further notifications if marked as paid).
Extensions:	• **2a.** If there are no upcoming due dates, the system skips sending reminders. • **7a.** If the reminder fails to send, the system retries and logs the failure.	

USE CASE 8: Analyze Spending Patterns (RAIMA TARIQ)

Use Case Name:	Analyze Spending Patterns
Scope:	FinPlan – A Personal Finance and Budget Management System
Level:	User goal
Primary Actor:	Student User
Stakeholders and Interests:	• **Student:** Wants insights into their spending behavior to make better financial decisions • **System Administrator:** Ensures the data is accurately processed and displayed.

Pre-conditions:	• The user is logged in. • The user has entered expense data into the system (daily, weekly, or monthly).	
Post-conditions:	• The system displays a visual analysis (graph/chart) of spending patterns. • The user can identify areas of overspending or adherence to budget goals.	
Main Success Scenario:	**Actor Action:**	**System Response:**
	1. The user selects the "Analyze Spending Patterns" option from the menu. 3. The user selects the time period for analysis (e.g., "Last Month"). 5. The user confirms the selection. 7. The user reviews the spending patterns.	2. The system displays options to choose a time period 4. The system retrieves the user's expense data for the selected time period. 6. The system generates a visual representation of spending categories.
Extensions:	• **3a.** If no expense data is available for the selected time period, the system displays a message: "No data available for this period." • **6a.** If the user requests a detailed analysis, the system provides category breakdowns with further insights.	

USE CASE 9: Integrate with Other Financial Tools (RAIMA TARIQ)

Use Case Name:	Integrate with Other Financial Tools
Scope:	FinPlan – A Personal Finance and Budget Management System

Level:	User goal	
Primary Actor:	Student User	
Stakeholders and Interests:	● **Student User:** Wants to simplify tracking their financial transactions by linking bank accounts to automate data input. ● **Banking Institutions:** Facilitate secure data access via APIs while ensuring user privacy and financial security. ● **System:** Ensures the integration process is secure and reliable.	
Pre-conditions:	● The user is logged into the system. ● The external financial tool supports integration with third-party platforms. ● The user has necessary login credentials and permissions for the external tool.	
Post-conditions:	● The system successfully integrates with the external financial tool. ● Financial data is synced between FinPlan and the external tool.	
Main Success Scenario:	**Actor Action:**	**System Response:**
	1. The user selects the "Integrate with External Financial Tools" option. **3.** The user selects an external tool (e.g., bank account) to integrate. **5.** The user provides the necessary credentials and permissions **7.** The user receives confirmation that the integration was successful.	**2.** The system presents available tools (e.g., bank accounts, investment platforms) for integration. **4.** The system prompts the user for credentials and any required permissions for the external tool. **6**. The system connects to the external tool and verifies the connection. **8.** The system begins syncing financial data between FinPlan and the external tool.
Extensions:	● **4a**. If the connection fails due to incorrect credentials, the system prompts the user to re-enter valid login details. ● **6a.** If the external tool does not support integration, the system informs the user that the integration is not possible.	

USE CASE 10: Create Budgets for Multiple Users (UMAMA BAJWA)

Use Case Name:	Create Budgets for Multiple Users	
Scope:	FinPlan – A Personal Finance and Budget Management System	
Level:	User-goal level	
Primary Actor:	Student User	
Stakeholders and Interests:	• **Students:** Want to manage shared financial obligations efficiently and ensure everyone contributes fairly. • **System:** Want to provide a seamless experience that supports collaboration without over-complicating the interface.	
Pre-conditions:	• Users must be registered in the system and logged in. • Users must have already set up individual profiles with financial data**.**	
Post-conditions:	A new shared budget is created, and all associated users have access to view and update it.	
Main Success Scenario:	**Actor Action:**	**System Response:**
	1. User navigates to the "Create Budget" section. **3.** The user enters budget details including categories, amounts, and user assignments and submits the form. **5**. The user can view and manage the created budget	**2.** System displays a form for creating a new budget. **4.** System validates the input and creates the new budget. **6.** The system updates the budget list and makes it available for tracking.

Extensions:	• **4a. Invalid Data:** If the system detects invalid data (e.g., missing fields, incorrect formats), it prompts the user to correct the errors before proceeding. • **6a. Save Failure:** If the system fails to save the budget due to a technical issue, it notifies the user and suggests retrying.

USE CASE 11: Lock the Budget (UMAMA BAJWA)

Use Case Name:	Lock the budget	
Scope:	FinPlan – A Personal Finance and Budget Management System	
Level:	User goal	
Primary Actor:	Student User	
Stakeholders and Interests:	• **Student User**: Wants to lock the budget to prevent further changes and ensure financial discipline. • **System**: Ensures the budget is securely locked and prevents unauthorized modifications.	
Pre-conditions:	• The user must be logged into the FinPlan system. • The user must have an existing budget created.	
Post-conditions:	The student user is notified when a budget limit is reached and the category is locked	
Main Success Scenario:	**Actor Action:**	**System Response:**

	1. The user logs into FinPlan and selects the option to lock a budget. **3**. The user selects the budget to lock. **5.**The user confirms the action.	**2.** The system displays a list of available budgets. **4.** The system prompts the user to confirm the locking action. **6.** The system locks the budget, preventing further modifications, and logs the activity.
Extensions:	● **4a. Budget Already Locked**: If the selected budget is already locked, the system notifies the user and suggests unlocking it first if changes are needed. ● **6a. Locking Failure**: If the system fails to lock the budget due to a technical issue, it notifies the user and suggests retrying..	

USE CASE 12: Manage Accounts (UMAMA BAJWA)

Use Case Name:	Manage Accounts
Scope:	FinPlan – A Personal Finance and Budget Management System
Level:	User-goal level
Primary Actor:	System
Stakeholders and Interests:	● **Student User**: Wants their account to be secure and their data to be managed properly. ● **System**: Ensures data integrity, security, and smooth operation of the account management features.
Pre-conditions:	● The user must be logged into the FinPlan system. ● The system must have user accounts created.

Post-conditions:	• User accounts are created, updated, or deleted as needed. • The system logs all account management activities for audit purposes.	
Main Success Scenario:	**Actor Action:**	**System Response:**
	1. The user logs into FinPlan and selects the option to manage their account. **3.** The user chooses to update their account information. **5.** The user updates the necessary account details and submits the changes. **7.** The user confirms the changes.	**2.** The system verifies the user's credentials and grants access to account management features. **4.** The system displays the current account details. **6.** The system validates the input data and updates the account information. **8.** The system saves the updated account details and logs the activity for audit purposes.
Extensions:	• **6a. Invalid Information:** If the system detects invalid information (e.g., missing fields, incorrect formats), it prompts the user to correct the errors before proceeding. • **8a. Processing Failure:** If the system fails to process the request due to a technical issue, it notifies the user and suggests retrying	

USE CASE 13: Sync and Share data (UMAMA BAJWA)

Use Case Name:	Sync and share data
Scope:	FinPlan – A Personal Finance and Budget Management System
Level:	User-goal level

Primary Actor:	Student User	
Stakeholders and Interests:	• **Student User**: Wants to ensure their financial data is up-to-date across devices and share data with others for collaborative financial planning. • **System**: Ensures data integrity, security, and smooth operation of the sync and share features.	
Pre-conditions:	• The user must be logged into the FinPlan system. • The system must have user data available for syncing and sharing.	
Post-conditions:	• The user's data is synced across devices. • The data is shared with specified users..	
Main Success Scenario:	**Actor Action:**	**System Response:**
	1. The user logs into FinPlan and selects the option to sync data. **3.** The user initiates the sync process. **5**. The user selects the option to share data. **7**. The user enters recipient details and confirms sharing. notifications, and logs the sync and share activities.	**2.** The system verifies the user's credentials and prepares data for syncing. **4.** The system syncs the data across devices and confirms the completion. **6**. The system prompts for recipient details and permissions. **8**. The system validates, sends notifications, and logs the sync and share activities.
Extensions:	• **4a. Sync Failure**: If the system fails to sync data due to a technical issue, it notifies the user and suggests retrying. • **6a. Invalid Recipient Details**: If the system detects invalid recipient details (e.g., non-existent email addresses), it prompts the user to correct the errors before proceeding. • **8a. Notification Failure**: If the system fails to send notifications due to a technical issue, it notifies the user and suggests retrying.	

USE CASE 14: Register User (UMAMA BAJWA)

Use Case Name:	Register User	
Scope:	FinPlan – A Personal Finance and Budget Management System	
Level:	User-goal level	
Primary Actor:	Student User	
Stakeholders and Interests:	• **Student User**: Wants to create an account to access the FinPlan system and manage personal finances. • **System**: Ensures the registration process is secure and user data is stored correctly.	
Pre-conditions:	• The user must have access to the FinPlan system. • The system must be online and available..	
Post-conditions:	The user account is created and stored in the system.	
Main Success Scenario:	**Actor Action:**	**System Response:**
	1. The user accesses the FinPlan system and selects the option to register. **3.** The user fills in the required details (e.g., name, email, password) and submits the form. **5**. The user confirms the registration	**2.** The system displays the registration form. **4.** The system validates the input data. **6**. The system creates the user account, stores the data, and confirms the registration.

Extensions:	• **4a. Invalid Data**: If the system detects invalid data (e.g., missing fields, incorrect formats), it prompts the user to correct the errors before proceeding. • **6a. Registration Failure**: If the system fails to create the account due to a technical issue, it notifies the user and suggests retrying.