Cairo University
Faculty of Computers and Artificial Intelligence

CS251

Introduction to Software Engineering

Money Minds

Software Requirements Specifications

Version 1.0

Loai Hataba   20230553

Abdullah Mohamed   20230231

Hossam Abdelaziz   20230121

CS251: Phase 1 – HoodRatz
Project: Money Minds

Software Requirements Specifications

March 2025

Contents

CS251: Phase 1 – HoodRatz
Project: Money Minds

Software Requirements Specifications

Team

ID	Name	Email	Mobile
20230553	Loai Hataba	LoaiHataba@gmail.com	01275397858
20230231	Abdullah Mohamed	abdallamohammmed649@gmail.com	01157756931
20230121	Hossam Abdelaziz	hossamabdelaziz2295@gmail.com	01094723054

Document Purpose and Audience

Purpose:

The purpose of this paper is to make the requirements for this software more understandable by outlining the features that the software must have in order to reduce development time and expenses.

Audience:

- Software Development Team
- Stakeholders

Introduction

Software Purpose

The purpose of the Money Minds Software is to help users set and achieve financial goals while providing guidance for better financial planning and expense management.

Software Scope

Money Minds Software implements a budget tracking system for individual users or even organizations, offering various functionalities for expense tracking, income management, setting financial goals, spending analysis, and budget planning. The app helps users monitor their financial

Software Requirements Specifications

activities, generate reports, and gain insights into their spending habits to make informed financial decisions.

Definitions, acronyms, and abbreviations

Phrase	Definition
Budget	A financial plan written down to decide how much money would be spent each month on what.
Income Sources	Something that provides a regular supply of money, such as employment, investments, or real estate
Expenses	The money spent on goods, services, and obligations such as rent, bills, and groceries.
Transactions	Records of money moving in and out, including income and expenses.
Financial Goals	Goals set by users to save or spend within specific limits over a period of time.
Uptime	The percentage of time the system is operational and accessible to users.
Downtime	The period when the system is unavailable due to maintenance or issues.
Scalability	The ability of the system to handle increased numbers of users and transactions without performance degradation.
Usability	How simple it is for people to understand and use the application
OWASP	Open Web Application Security Project, a framework that defines best practices for web security.
Portability	Portability defines how a system or its element can be launched on one environment or another.

Software Requirements Specifications

Compatibility	Compatibility defines how a system can co-exist with another system in the same environment.
Reliability	Reliability measures the ability of a piece of equipment to perform its intended function for a specific interval without failure.
Maintainability	The simplicity with which a software system or component can be updated to fix bugs, enhance functionality or other features, or adjust to a changing environment.
Availability	Availability measures the ability of a piece of equipment to be operated if needed.

Requirements

Functional Requirements

Requirement ID	Requirement State
FR01	As soon as the user launches the application, a login/sign-up screen will appear.
FR02	If the user chooses to log in, the application will provide a box for them to enter their username and password.
FR03	If a user chooses to register, the app will provide a signup page where they can enter their personal information, including first name, last name, password, gender, country, and email address. And asking if the user would like to sync with their bank account. This registration process will enable personalized features and secure access to their financial data.

Software Requirements Specifications

FR04	The application shall provide users with three primary tabs for managing their finances:

	1. **"My Budget"** tab where users can:
	  o Create a new budget by specifying income sources, expense categories, and budget limits.
	  o Edit or delete existing budgets.
	  o View a summary of their financial status, including total income, total expenses, and remaining budget.
	2. **"Transactions"** tab where users can:
	  o Add new transactions by entering details such as category, amount, date, and payment method.
	  o Edit or delete recorded transactions.
	  o Filter transactions by date, category, or amount.
	3. "Goals" tab where users can:
	  o View their financial goals
	  o Set new financial goals by specifying target savings, timeframes, and desired spending limits.
	  o Track progress toward their financial goals based on their recorded transactions.

FR05	In addition to the three primary tabs, the application should provide a Reports panel where users can: 1. View graphical representations (charts, graphs) of their spending and savings trends. 2. Generate summary reports of monthly or yearly income and expenses. 3. Export reports in PDF or Excel format for external use.
FR06	The application should provide notifications to help users stay on track with their financial goals: 1. Budget Alerts: Notify users when they are approaching or exceeding their budget. 2. Bill Reminders: Send reminders for upcoming bills and due payments. 3. Goal Progress Notifications: Notify users of their progress toward financial goals.

Software Requirements Specifications

Non-Functional Requirements

| Measure	Details
Performance	• The sign-in/sign-up screen should load within 3 seconds under normal network conditions. • All screens should be loaded in under 7 seconds.
Scalability	• The system must handle up to 3,000 concurrent users without performance degradation. • It should be designed to scale dynamically as the user base grows.
Maintainability	• Critical failures must be resolved within 72 hours in at least 80% of cases. • High-priority bugs should have a 75% resolution rate within three days
Reliability & Availability	• The system must have a monthly uptime of at least 90% and a monthly downtime of no more than 72 hours.
Portability & Compatibility	• The system should be cross-platform, Android, and iOS devices. • The app should be optimized for both low-end and high-end mobile devices, ensuring smooth performance. • The system should be compatible with the latest and previous two major OS for Android and iOS. • The system should operate in both online and offline modes, enabling users to access essential features such as viewing budgets and transactions offline while requiring internet access for synchronization.
Security	• Users must only access their own financial data, with role-based access control (RBAC) or permission-based restrictions in place. • Unauthorized users must not access restricted pages or perform unauthorized actions. • The system should follow OWASP security standards to prevent common web vulnerabilities. • Data must be encrypted at rest and in transit to protect user privacy to ensure privacy and security.
Usability	• The interface should be user-friendly, intuitive, and require minimal training.

Software Requirements Specifications

	• The system should follow UX best practices for accessibility and ease of navigation. • The interface should provide clear navigation, consistent UI elements, and responsive design for an optimal user experience across different screen sizes.

System Models

Use Case Model
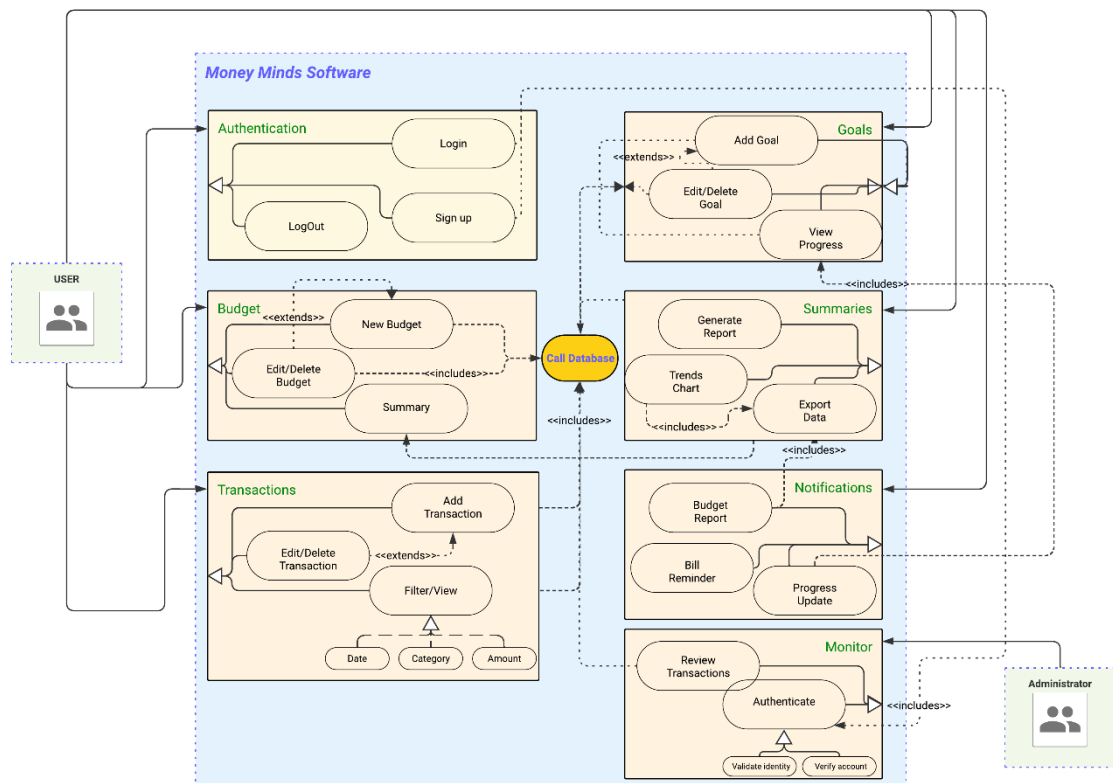
Software Requirements Specifications

Enriched User Stories

- **User Story #1**

User Story ID	US #1
User Story Name	User Sign-up
Actors	User
Description	**As** a user I **like** to be able to sign-up to the application **So** that I can sign in and use its services
Per condition	
Post condition	
Acceptance Criteria	**Given** I'm a user and I'm on the sign-up/log-in page **When** I click sign-up and fill in all the fields with my information and click submit **Then** the system signs me up

- **Scenarios**

Normal Scenario

| Actor Action	System Response
1- User clicks on sign-up	
	2- Redirect to sign-up page 3- System displays boxes to enter information
3- User Fills in the boxes 4- User chooses whether they want to sync with their bank account 5- User clicks submit	
	6- System verifies the user information - If user chooses to sync system adds more features to user package - Else system doesn't add certain features to user's account 7- System saves user info in the database

Software Requirements Specifications

Exceptional Scenario

Actor Action	System Response
1- User clicks on sign-up	
	2- Redirect to sign-up page 3- System displays boxes to enter information
3- User Fills in the boxes 6- User clicks submit	
	6- System verifies the user information 7- System finds input information invalid 8- System displays "Please enter valid information." Message

- **Screen Design**
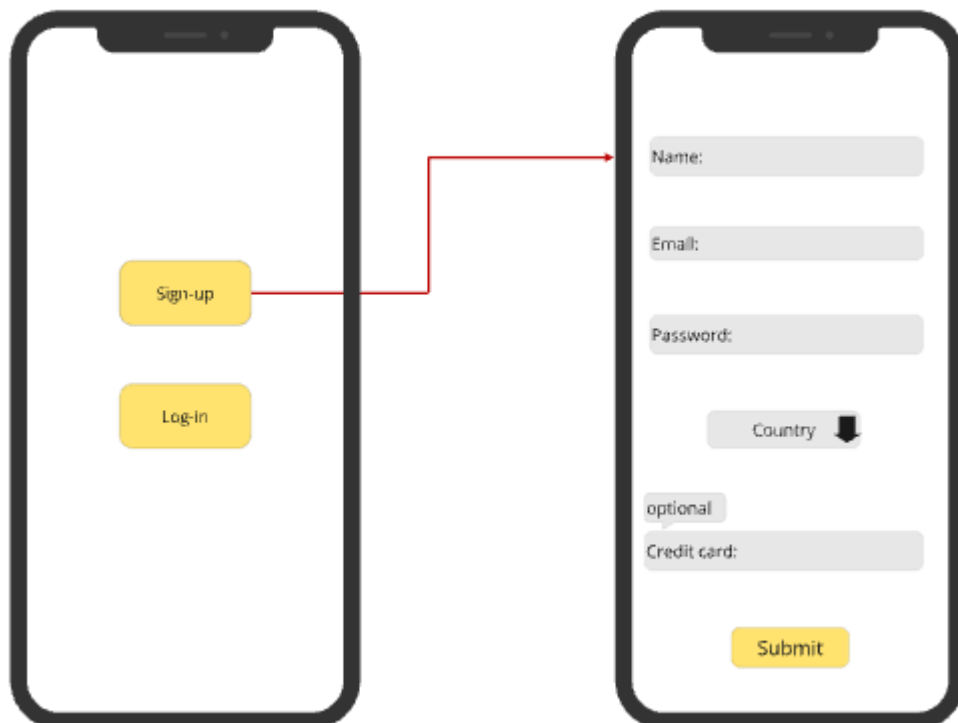
CS251: Phase 1 – HoodRatz
Project: Money Minds

Software Requirements Specifications

- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Name field	Text < 50 characters	Structured Text
Email field	Text < 50 characters	Structured Text
Password field	Alphanumeric < 50 Characters	Password confirmation and must include an uppercase letter, numbers and/or special characters
Country field	A-Z list	Middle eastern countries only??
Credit card field	Integer < 25 digits	

- **User Story #2**

User Story ID	US #2
User Story Name	User log-in
Actors	User
Description	**As** a User **I like** to be able to log in to the application **So** that I can use its services
Per condition	
Post condition	
Acceptance Criteria	**Given** I'm a logged-out user on the log-in page **When** I fill in the "Username" and "Password" fields with my authentication credentials and I click the log-In button **Then** the system logs me in to my account

Software Requirements Specifications

- **Scenarios**

Normal Scenario

Actor Action	System Response
1- User clicks on log-in	
	2- Redirect to log-in page 3- System displays boxes to enter username and password
4- User Fills in the boxes 5- User clicks log-in	
	6- System verifies the user information
	7- System accesses user to their account

Exceptional Scenario

| Actor Action	System Response
1- User clicks on log-in	
	2- Redirect to log-in page 3- System displays boxes to enter username and password
4- User Fills in the boxes 5- User clicks log-in	
	6- System verifies the user information 7- System finds input information invalid 8- System displays "Please enter valid username/password." Message

Software Requirements Specifications

- **Screen Design**
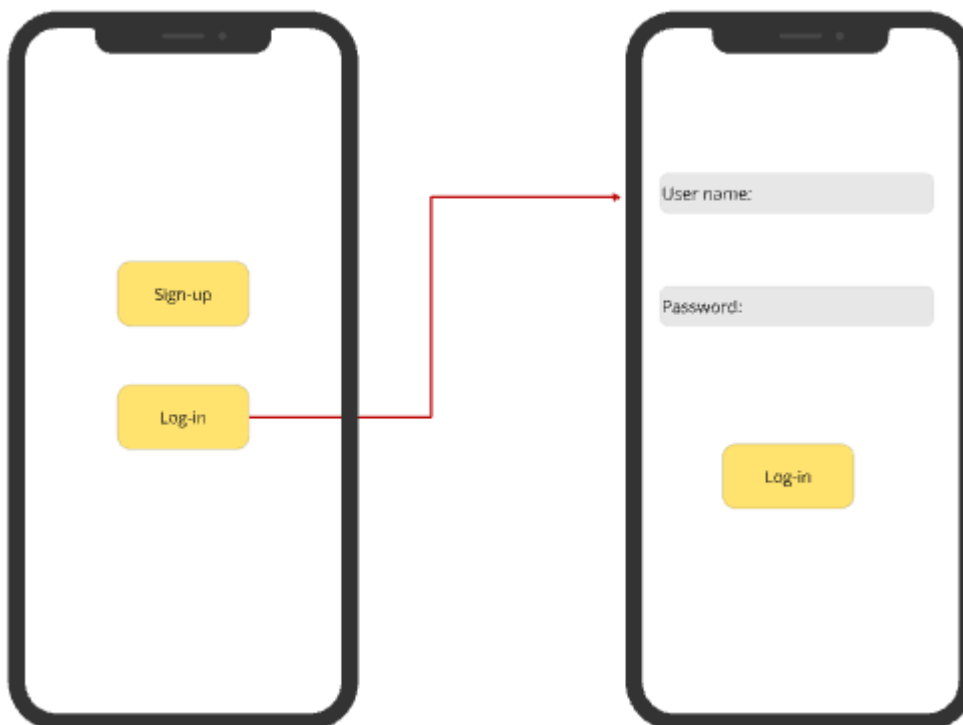
- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Name field	Text < 50 characters	Structured Text
Password field	Text < 50 characters	Structured Text

Software Requirements Specifications

- **User Story #3**

User Story ID	US #3
User Story Name	Display Budgets
Actors	User
Description	**As** a User **I like** to be able to see my budgets list **So** that I can access a certain budget of mine
Per condition	
Post condition	
Acceptance Criteria	**Given** I'm a logged-in user and I'm on the home screen **When** I click on "MyBudgets" button **Then** the system redirects me to the Budgets list

- **Scenarios**

Normal Scenario

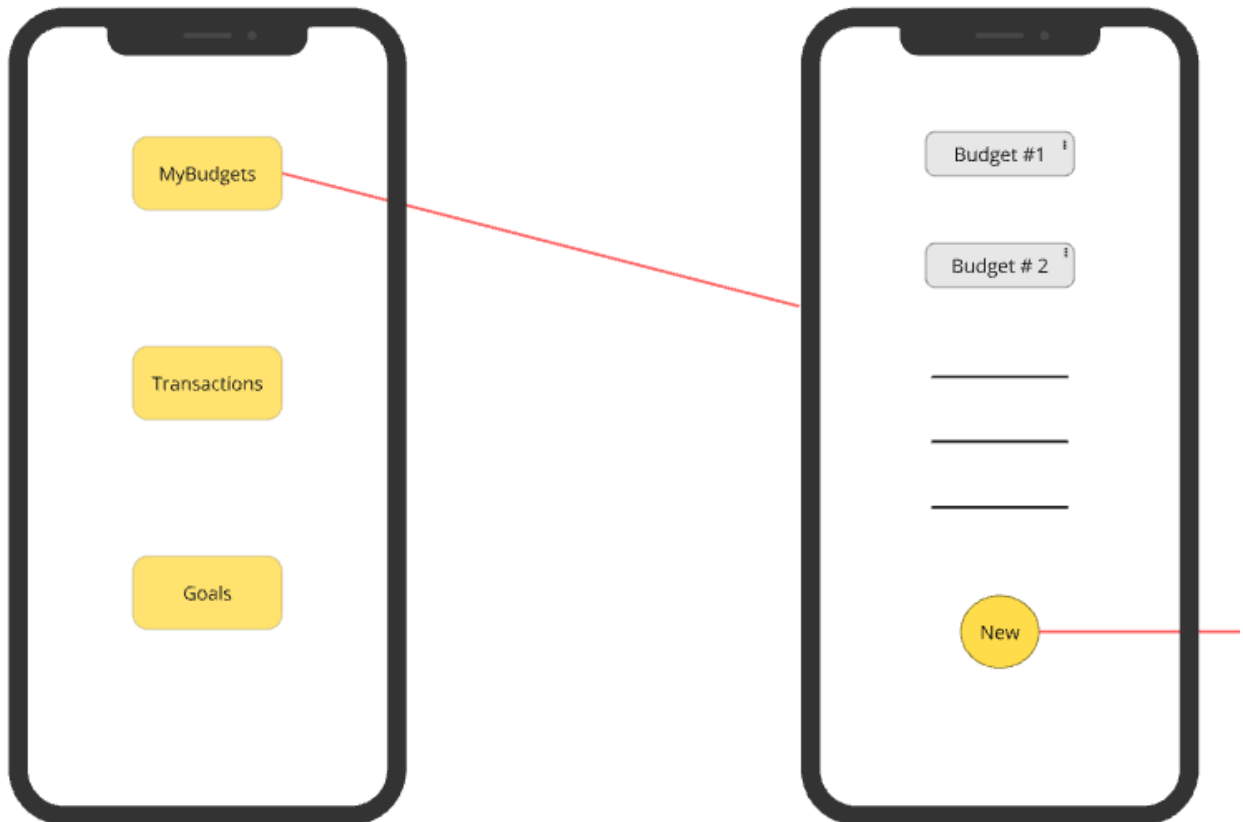| Actor Action	System Response
1- User clicks on "MyBudgets"	
	2- Redirect to Budgets page 3- System displays all of user's Budgets and add budget button • If user has no budgets only display add button

Software Requirements Specifications

- **Screen Design**

- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Budget name fields	Text < 100 characters	Structured Text

Software Requirements Specifications

- **User Story #4**

User Story ID	US #4
User Story Name	Add/Edit a Budget
Actors	User
Description	**As** a User **I like** to be able to add or edit a budget in my budget list **So** that I can better manage my finances
Per condition	
Post condition	
Acceptance Criteria	**Given** I'm a user in the "MyBudgets" page **When** I click on the "New" button or the 3 dots on any budget then click "Edit" and fill in the budget information fields **Then** the system adds a budget or applies the changes to the existing budget on my budget list

- **Scenarios**

Normal Scenario

Actor Action	System Response
1- User clicks on "New" or clicks on the 3 dots then "Edit"	
3- User clicks on "Done" button	2 - System displays several boxes for input, including Budget name, Category, Budget income, Expenses, Budget limit, and Time frame 4- System adds the budget to the database or applies the changes to the existing budget

Software Requirements Specifications

Exceptional Scenario

Actor Action	System Response
1- User clicks on the 3 dots then "Delete"	
	2- System deletes the budget from the database and from the budgets list

- **Screen Design**
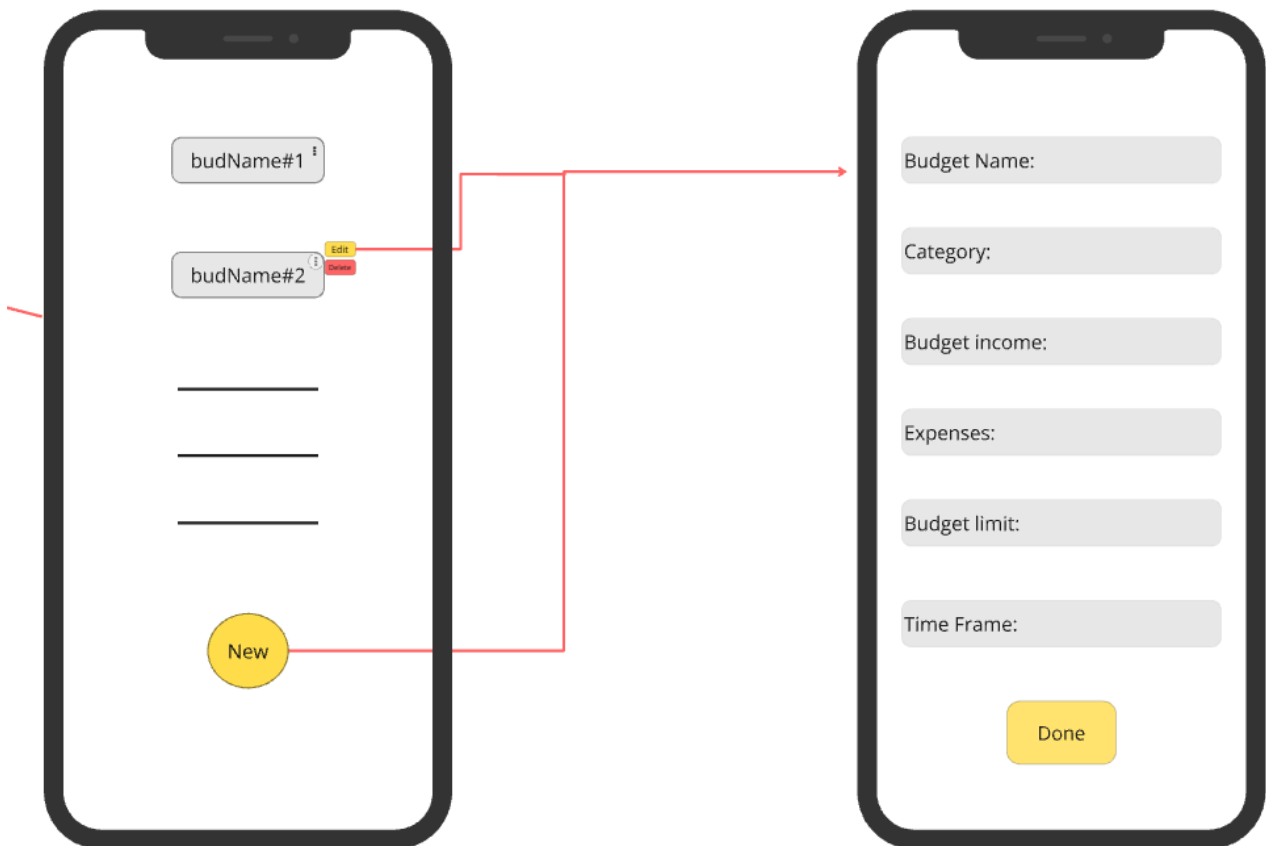
Software Requirements Specifications

- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Budget name field	Text < 100 characters	Structured Text
Category field	Text < 50 characters	Structured Text
Budget income field	Float < 10 digits	Only 2 digits after decimal point and >= 0
Expenses field	Float < 10 digits	Only 2 digits after decimal point and >= 0
Budget limit field	Float < 10 digits	Only 2 digits after decimal point and >= 0
Time frame field	Date/Time < 15 Characters	DD/MM/YY 24-hour format

- **User Story #5**

User Story ID	US #5
User Story Name	Viewing a Budget
Actors	User
Description	**As** a user I **like** to be able to view my budget's details **So** that I can track my income, expenses and remaining balance
Per condition	
Post condition	
Acceptance Criteria	**Given** I'm a user and I already created a budget **When** I click on my budget title form the budget list **Then** the system opens a detailed page of info about the budget

Software Requirements Specifications

- **Scenarios**

  Normal Scenario

Actor Action	System Response
1- User clicks on one of their budgets on the budget list	
	2- Redirect to that budget page 3- System displays information about the budget including summary, expenses breakdown, leftover money

- **Screen Design**
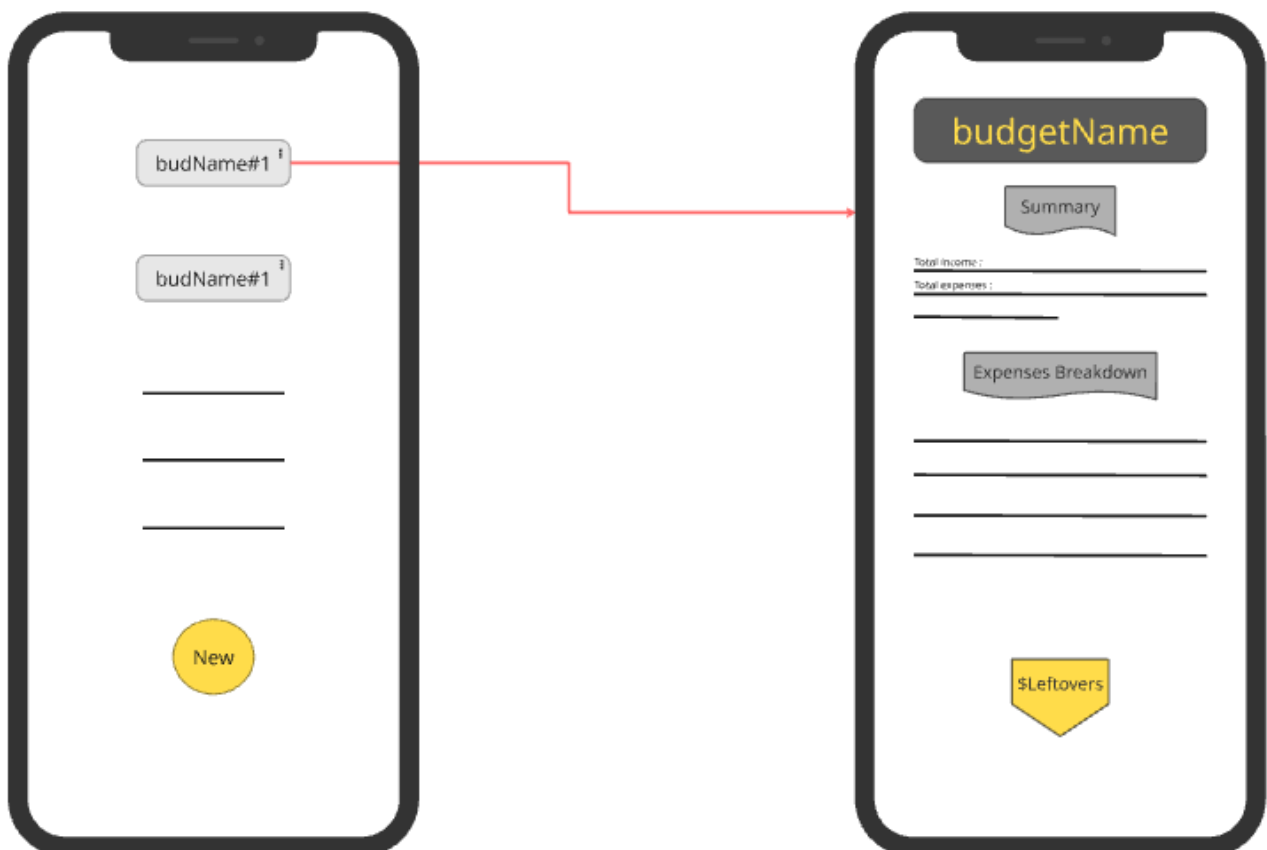
Software Requirements Specifications

- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Budget name field	Text < 100 characters	Structured Text
Summary information	Text < 1000 characters	Structured Text
Expenses information	Text < 1000 characters	Structured Text
Leftovers field	Float < 10 digits	Only 2 digits after decimal point and >= 0

- **User Story #6**

User Story ID	US #6
User Story Name	Add/Edit a transaction
Actors	User
Description	**As** a user I **like** to be able to see my transactions history **So** that I can track my income and add or edit individual transactions
Per condition	
Post condition	
Acceptance Criteria	**Given** I'm a user and I'm on the home screen and I go to the 'Transactions' page **When** I click on "New" or the 3 dots then "edit" and fill the boxes **Then** the system adds a new transaction or applies the changes to the existing transaction

Software Requirements Specifications

- **Scenarios**

Normal Scenario

Actor Action	System Response
1- User clicks on "Transactions"	
	2- Redirect to the "Transactions" page 3- System displays the history of all transactions the user has ever made and some details about them
4- User clicks on "New" or "Edit"	
	5- System displays several boxes of input including Transaction name , Type, Category, Amount, Date and Payment Method
6- User fills the boxes and clicks "Done"	
	7- System verifies transaction information 8- System adds the transaction or saves the changes to the existing transaction in the database

Exceptional Scenario

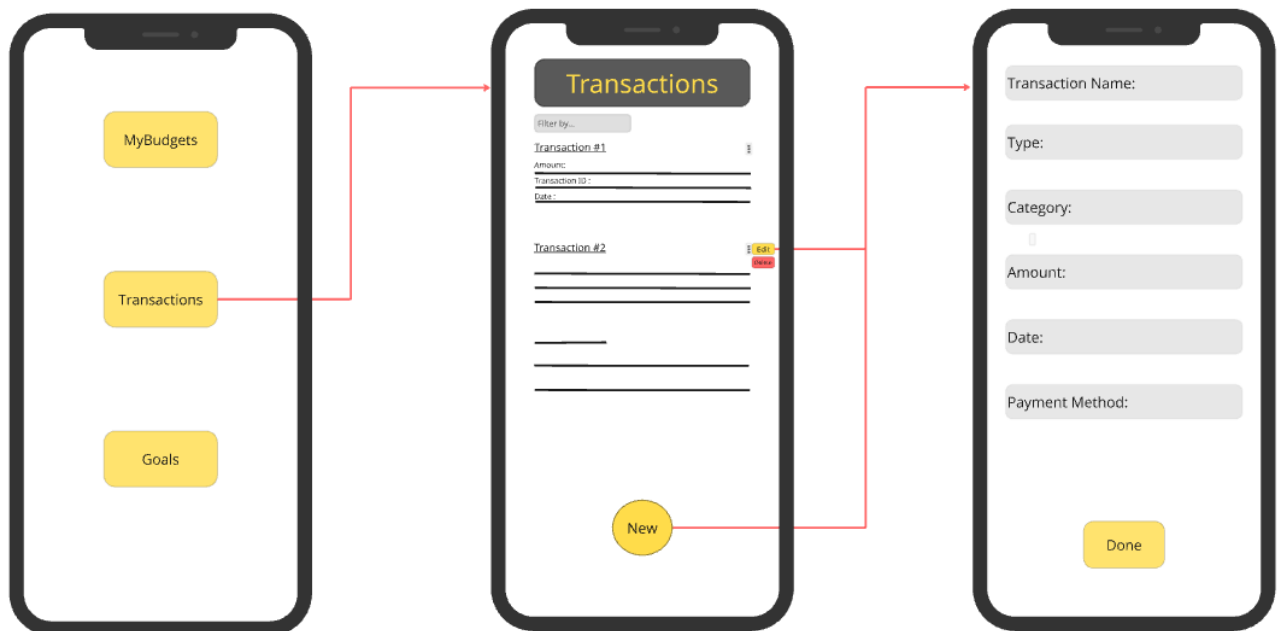| Actor Action	System Response
1- User fills the boxes and clicks "Done"	
	2- System verifies transaction information 3- System finds input information invalid 4- System displays "Please enter valid information" message

Software Requirements Specifications

- **Screen Design**

- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Transaction name field	Text < 100 characters	Structured Text
Category field	Text < 50 characters	Structured Text
Amount field	Float < 10 digits	Only 2 digits after decimal point
Type field	List = 2 strings	Expense or income
Date field	Date/Time < 15 Characters	DD/MM/YY 24-hour format
Payment Method field	List < 10 strings	Payment companies supported by the application whether its card or a payment app

Software Requirements Specifications

- **User Story #7**

User Story ID	US #7
User Story Name	Add/Edit a Goal
Actors	User
Description	**As** a user I **like** to be able to see and create financial goals **So** that I can manage my savings effectively and add or edit individual Goals
Per condition	
Post condition	
Acceptance Criteria	**Given** I'm a user and I'm on the home screen and I go to the 'Goals' page **When** I click on "New" or the 3 dots then "edit" and fill the boxes **Then** the system adds a new Goal or applies the changes to the existing Goal

- **Scenarios**

Normal Scenario

| Actor Action	System Response
1- User clicks on "Goals"	
	2- Redirect to the "Goals" page 3- System displays all the financial goals that was set the user with information about them and a progress bar
4- User clicks on "New" or "Edit"	
	5- System displays several boxes of input including Goal name, Target Amount, Time Frame
6- User fills the boxes and clicks "Done"	
	8- System adds the Goal or saves the changes to the existing Goal in the database

CS251: Phase 1 – HoodRatz
Project: Money Minds

Software Requirements Specifications

- **Screen Design**
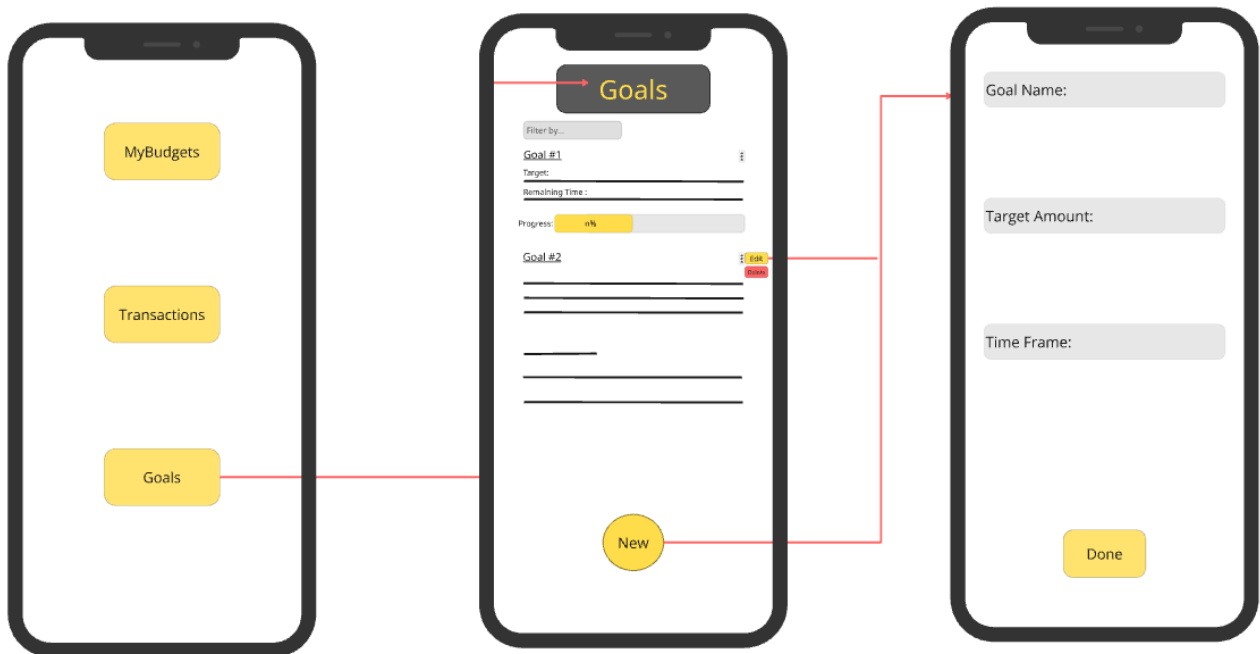
- **Data Dictionary:**

Element Label	Type/Length	Data Validation / Business Rule
Goal name field	Text < 100 characters	Structured Text
Target Amount field	Float < 10 digits	Only 2 digits after decimal point
Time Frame field	Date/Time < 15 Characters	DD/MM/YY 24-hour format

System Navigation Map

- **Draw a navigation map that show how the screens are related (See example at https://stuff.mit.edu/afs/sipb/project/android/docs/training/design-navigation/wireframing.html)**

Tools

- **Write a list of all tools used to develop the design (e.g., ArgoUML, Visual-Paradigm, mocqus, etc.)**

Software Requirements Specifications

Ownership Report

Student	Items he created
Loai Hataba	**Part of Use Case Model, Non-Functional Requirements, and User Stories #1 and #2.**
Abdallah Mohamed	Part of Introduction, Functional and Non-Functional Requirements
Hossam Mohamed	Part of User Stories #1 – 10 and Navigation map