

Name: Jay Patel, Heer Patel, Jatan Bhatt, Arth Modi

ID: 1001870971, 1001965102, 10011927386, 1001979538

CSE-5324 - 003

Assignment 4

11/15/2022

Split-Money

Project description: -

This program's purpose is to allow roommates, friends, group activities, and other parties to split costs. The simplest approach to quit stressing about "who owes who" when splitting expenses with friends and family is to use Split-Money. Anywhere in the world, Split-Money can be used to manage group spending on things like lodging, travel, and other items. When many people are splitting costs with one another, splitting the money can be a headache and make things difficult. Splitting expenditures in this circumstance will be made simpler with Split Money. Our goal is to reduce the stress and discomfort that money places on our most important relationships.

Functionalities which the app will provide: -

Add Expense: Split expenses between individuals or groups. Users can add the expenses in the app and divide that among users they wanted to.

Create Groups: Users can create groups in the application and split expenses among a group of people based on the percentage of expense each person owes.

Expense History: Users can keep track of all the expenses which took place in a period and keep track of incomplete settlements.

Record/Settle Payment: Record a payment and settle balances once the user has paid the expenses they owed.

Update Expense: Modify/Edit the expense shared with a group/individual. Users who added expense can Modify/Edit the expense.

Domain Model: -

To organize development items and automatically facilitate dependency discovery, a domain model is employed, offering a shared reference point. These relationships are used to calculate dependencies between development objects and are progressively enhanced to produce direct dependency linkages. Formally speaking, a domain model is a diagram that illustrates the connections between the different entities in the problem domain and other crucial information, such as the following:

Objects that fall under the domain's purview - things like expense, accounts, and transactions, for instance.

Behaviors such objects display while interacting with one another— In our system, for instance, you can add an expense and share them with a group. These interactions between the things in our domain happen frequently.

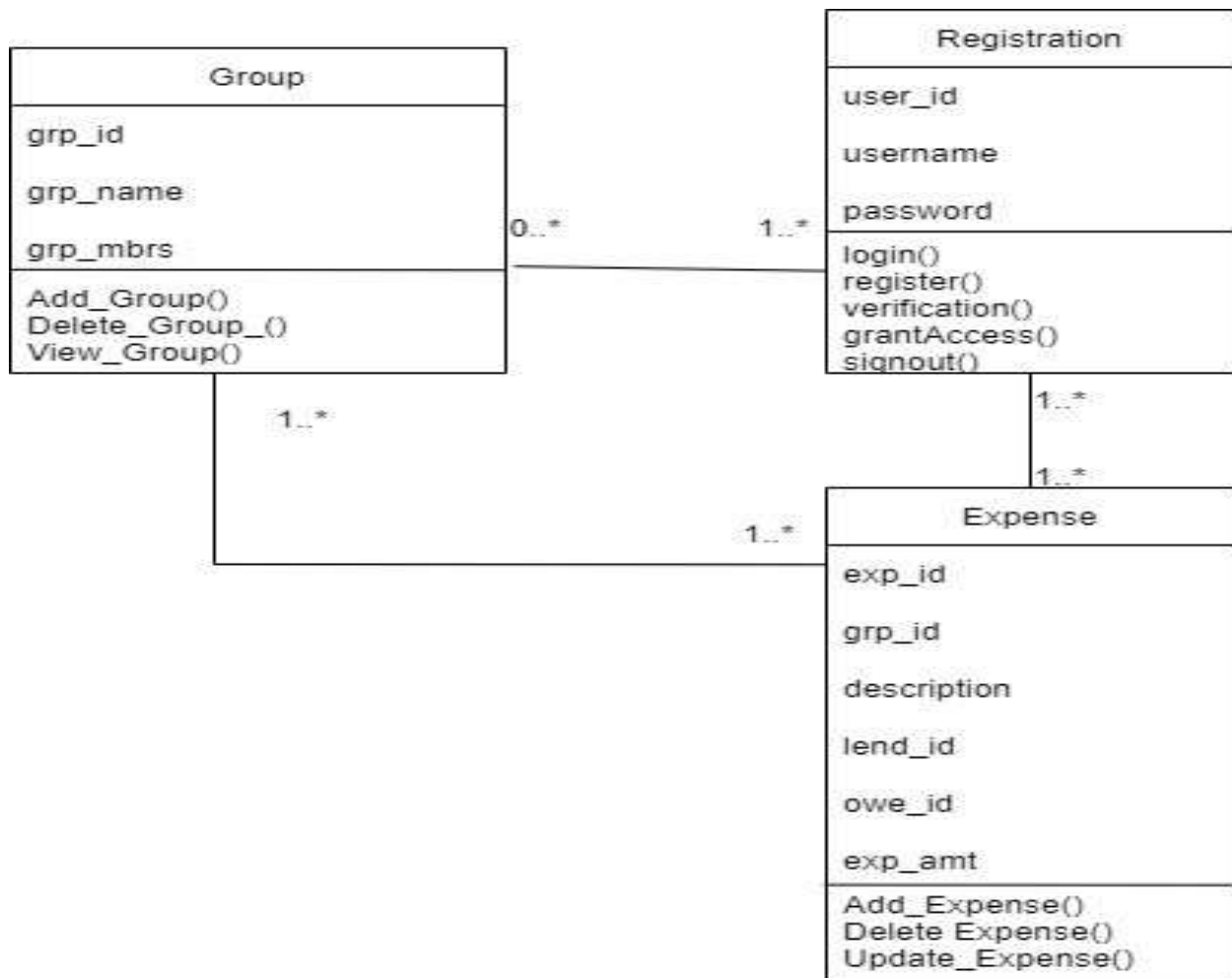
The environment in which the model runs—This refers to the collection of presumptions and restrictions that are pertinent to the issue domain and automatically apply to the software model you create. One of the presumptions that defines the context of our domain model for Split Money is that a new account can only be opened for a living person or entity.

In every model, every object (entity or value object) needs to have a distinct lifecycle pattern. Every one of the following events must have defined handling procedures for every type of object you have in your model:

Creation: The process is used to generate objects within the system.

Participation in behaviors: How the object is remembered during system interactions. In your system, you model an entity or a value object in this manner. A complex entity may be made up of both value objects and other entities. A Transaction entity, for instance, makes references to other entities like an Expense or other objects like Group or Account.

How the thing is kept in its persistent state is known as persistence. This covers matters like how to insert, update, remove, or query an object like an Expense if your persistent form is a relational database, as well as how to write the element to the persistent storage and get the details in response to system requests.



Let us examine the domain model's structural elements in more detail. Utilizing UML (Unified Modeling Language). The results of a brief structural study are as follows:

Numbers - This is an instance of plurality. It specifies the maximum number of instances of a class A that can be connected to a single instance of a class B. In other words, a relation like one-to-one or one-to-many is what a number denotes. These figures are on the "contains" connection for Expense, Transaction, and Groups in the example.

Actor System Interaction Model: -

Modeling and designing how the system interacts with the actors to carry out the use cases is known as actor-system interaction modeling.

The process of modeling actor-system interactions involves creating a two-column table that lists each interaction's actor input, actor action, and system response.

Sign up: -

Actor: User	System: App
	1. The web page displays a signup page.
2. User enters first name, last name, email id, details, username, and password and clicks Signup button.	3. Web page displays the welcome page.
4. User sees the welcome page.	

Login: -

Actor: User	System: App
	1. Web page displays login page.
2. User enters username and password and clicks login button.	3. Web page displays the welcome page.
4. User sees the welcome page.	

Create group: -

Actor: User	System: App
	1. Web page displays create group page.
2. User enters group name and usernames which user wants to add in that group.	3. Web page displays the group page.
4. User sees the group page.	

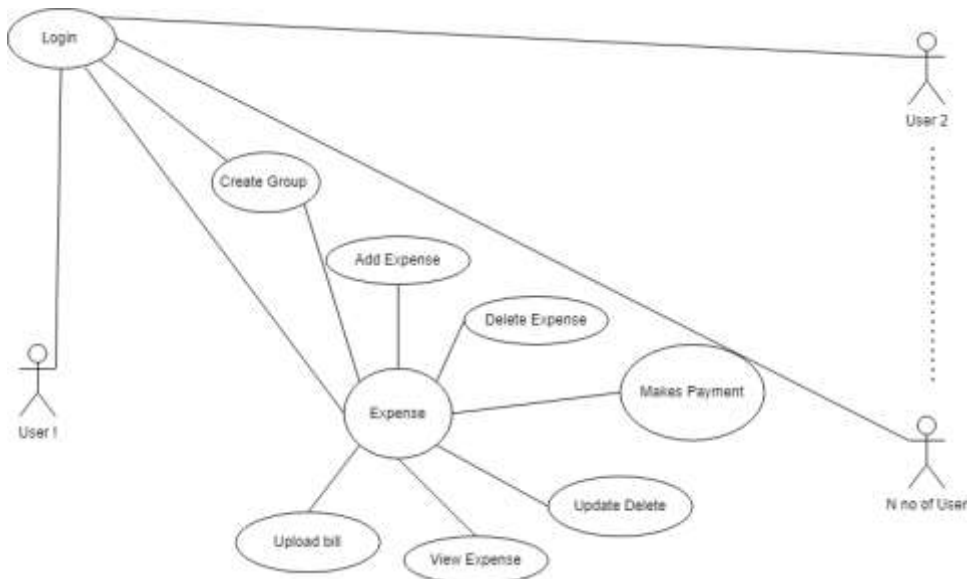
Added Expense: -

Actor: User	System: App
	1. Web page displays add expense page.
2. User enters username in which he/she wants to add expense and amount.	3. Web page displays the group page.
4. User sees the group page.	

Expense: -

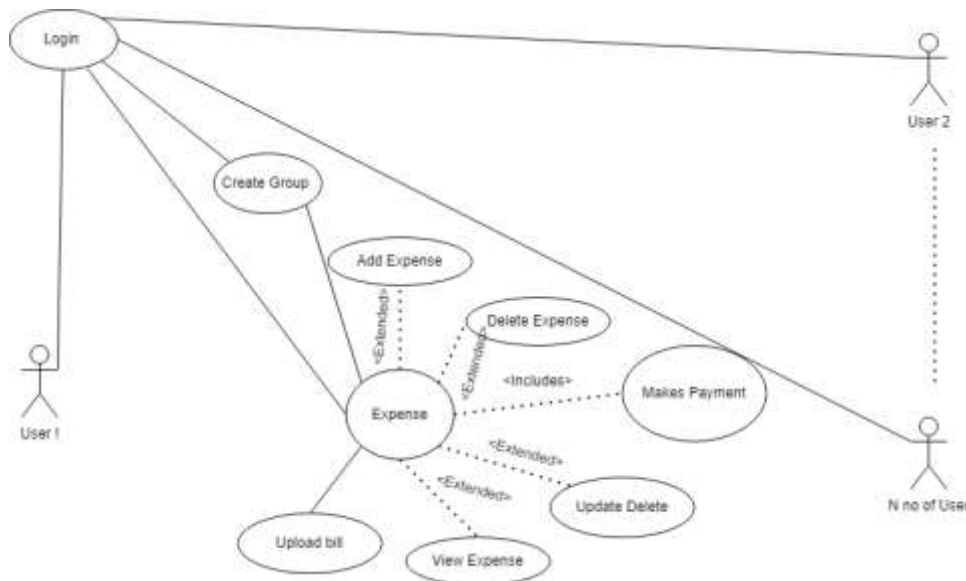
Actor: User	System: App
	1. Web page displays expense page.
2. User sees the expenses with the particular user.	

Use Case Diagram: -



The user has the option to split the cost with other users. Users can also form groups to split costs. Users collaborate to add, view, amend, and alter expenses. The expense that the other user created may be settled by other users. The user may also include a copy of the invoice to obtain proof of expense.

High Level use case: -



We do not have an extended use case

RUCTM: -

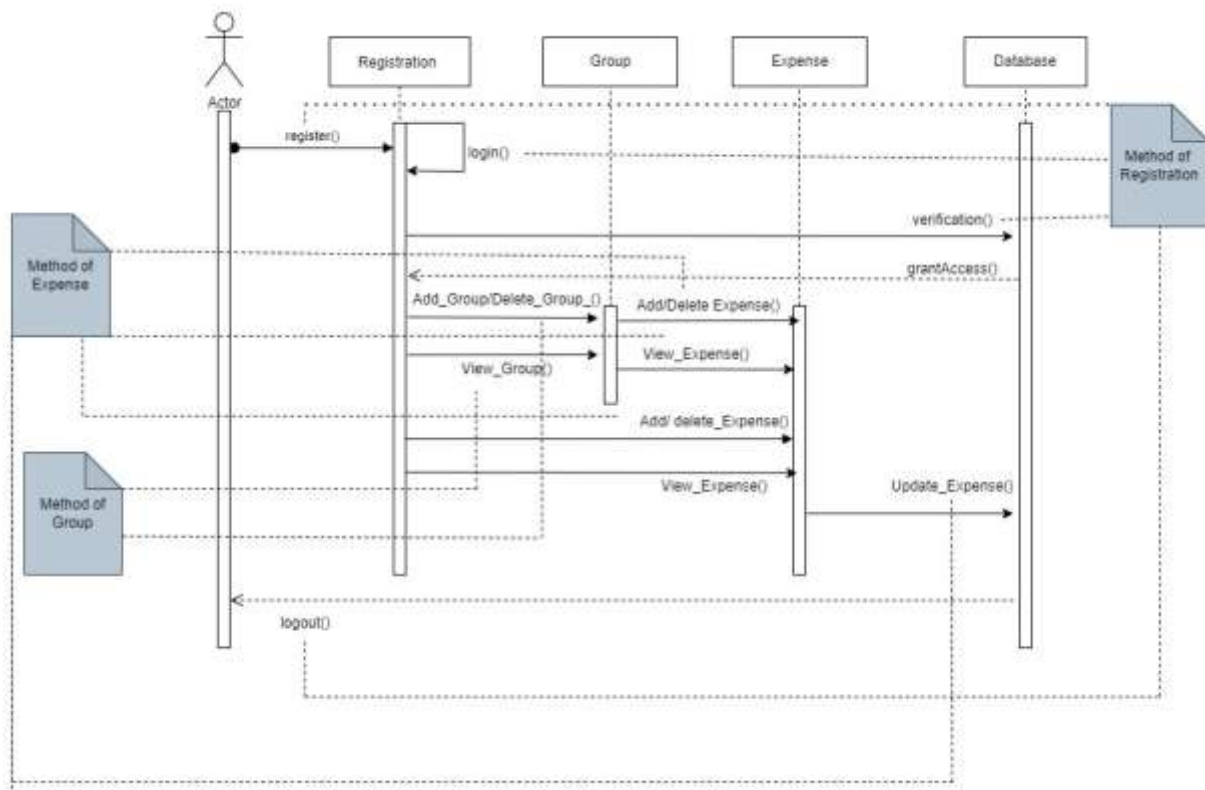
	Priority Weight	Login	Added Expense	View Expense	Modify Expense	Delete Expense	Upload Bill	Make Group
R1	1	X						
R2	4							X
R3	2		X	X	X	X		
R4	3						X	
UC Priority		1	2	2	2	2	3	4

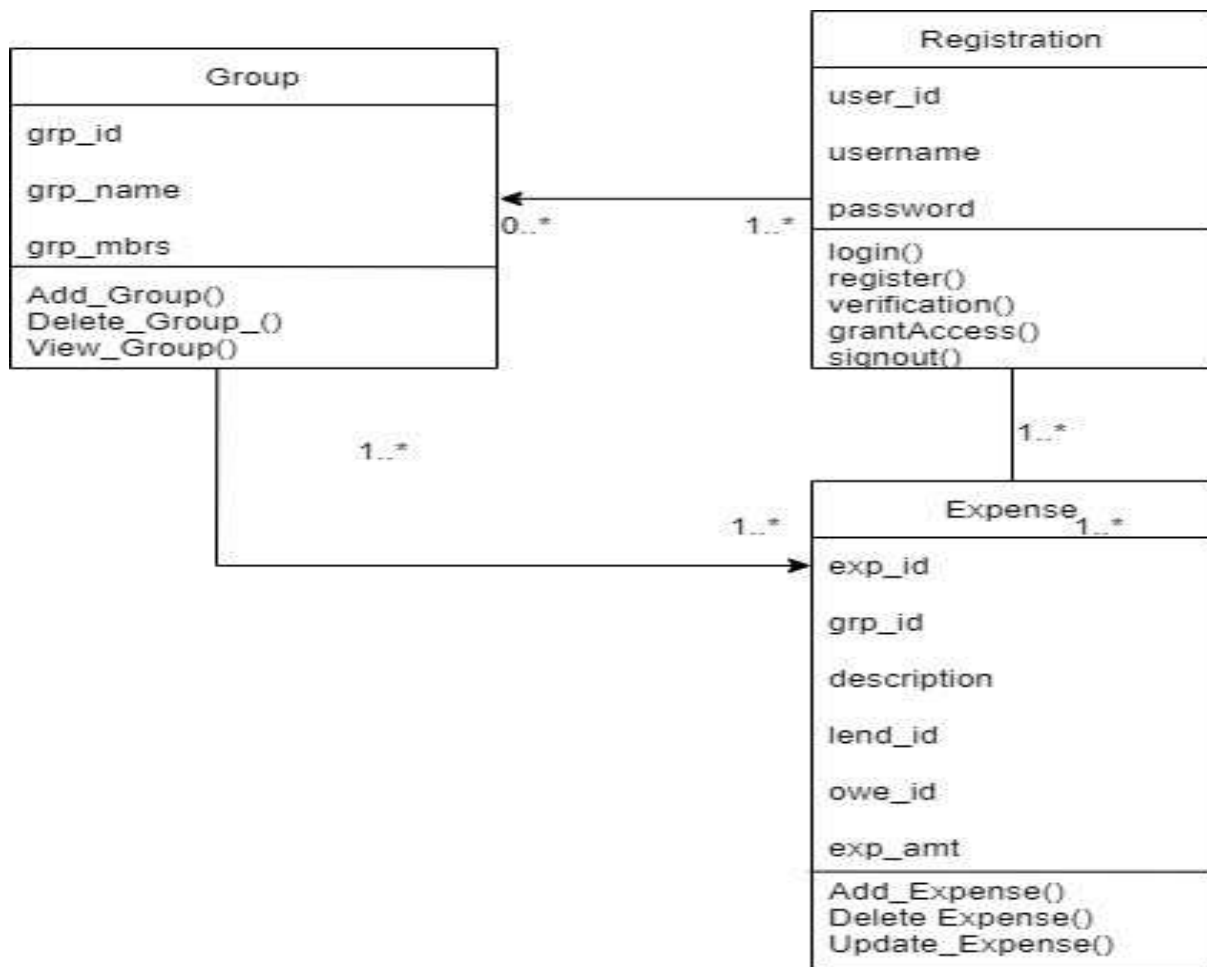
	Priority Weight	Effort per week	Depends on	Iteration 1	Iteration 2	Iteration 3
R1	1	5	-	1		
R2	4	5	R1			4
R3	2	5	R1	2		
R4	3	5	R3		3	
UC Priority		20		1	3	4

RUCTM for each iteration

Class Diagram: -

Class Diagram is a type of diagram that shows how system, attributes and methods interact among the objects. It is derived from domain and UML diagram. The main purpose is to show static structure of the system which helps developers and other team members to understand the system.



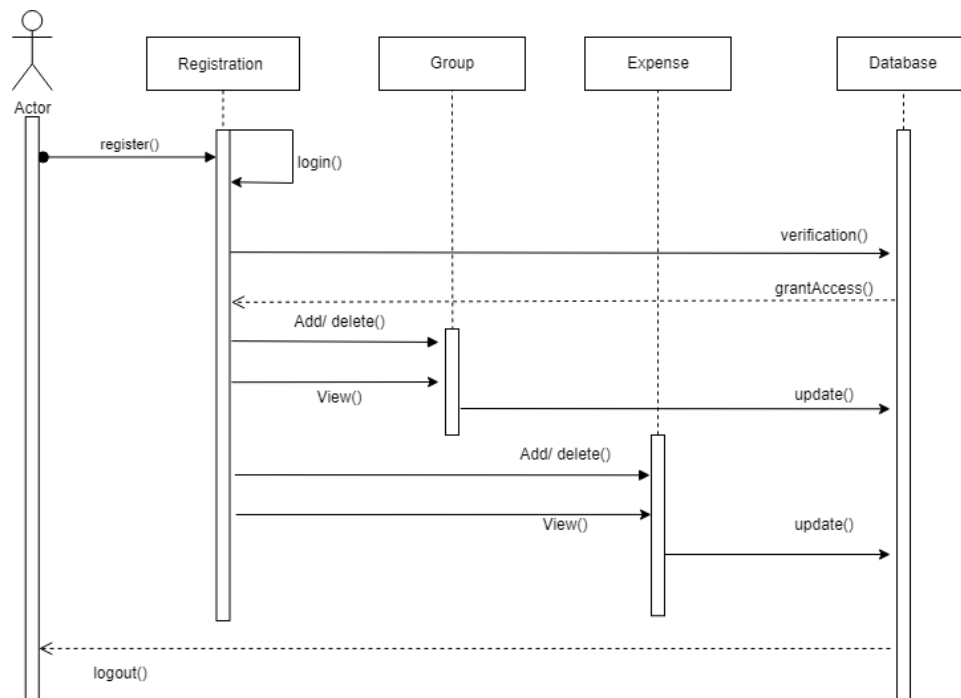


Sequence Diagram: -

UML Sequence Diagrams are interaction diagrams that describe the steps used to complete an operation. They depict how items interact within the framework of a cooperation. By using the vertical axis of the diagram to represent time and the messages that are transmitted and when, sequence diagrams, which have a time focus, can visually depict the order of an interaction.

Purpose of Sequence Diagram:

- Model the complex interactions of a system's active items.
- Model the communication between instances of an item inside a team that implements a use case.
- Model the interaction of the items in the operation-realizing collaboration.
- Either simulate general interactions (which would depict all possible routes through the interaction) or particular interactions (showing just one path through the interaction).

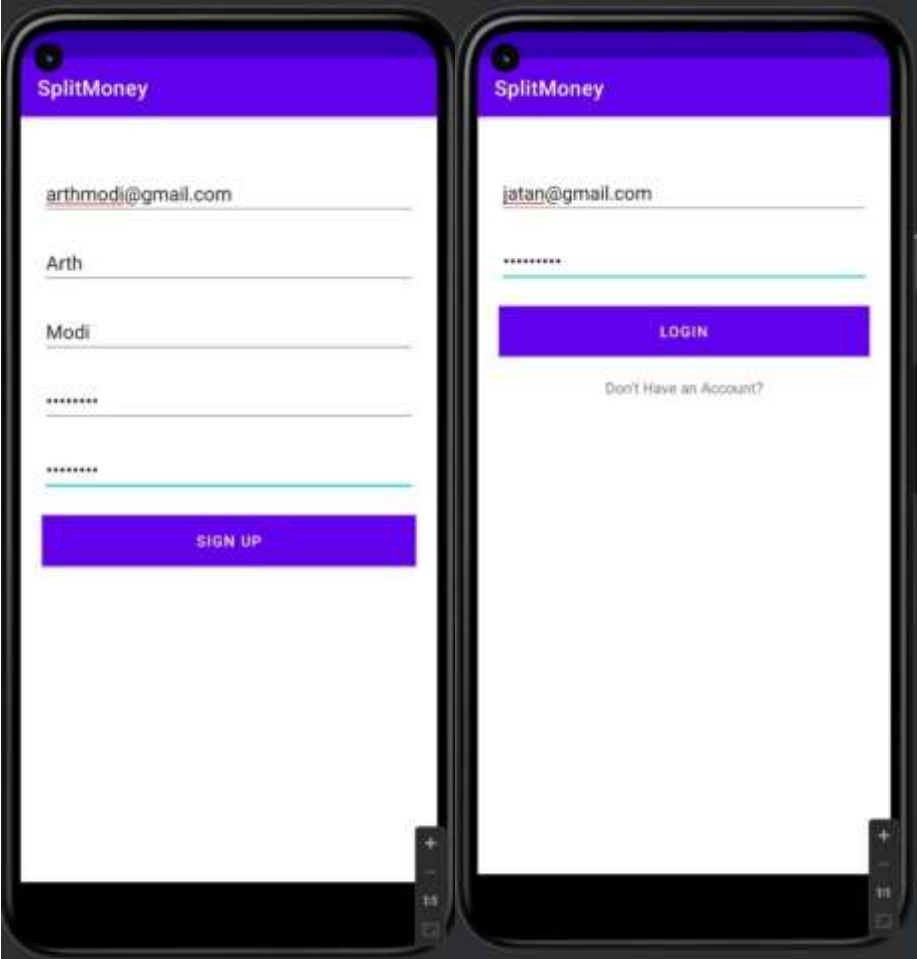


Screenshots of Implementation of code: -

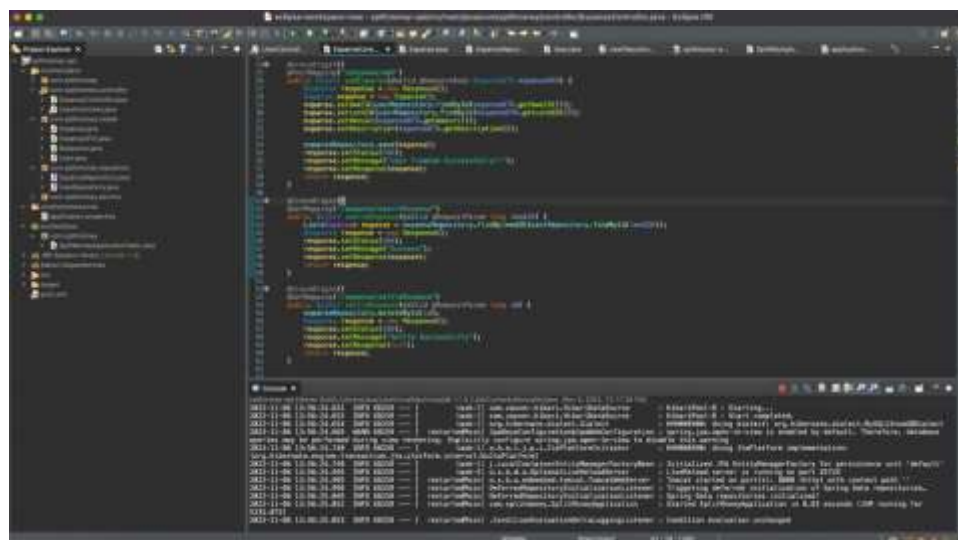
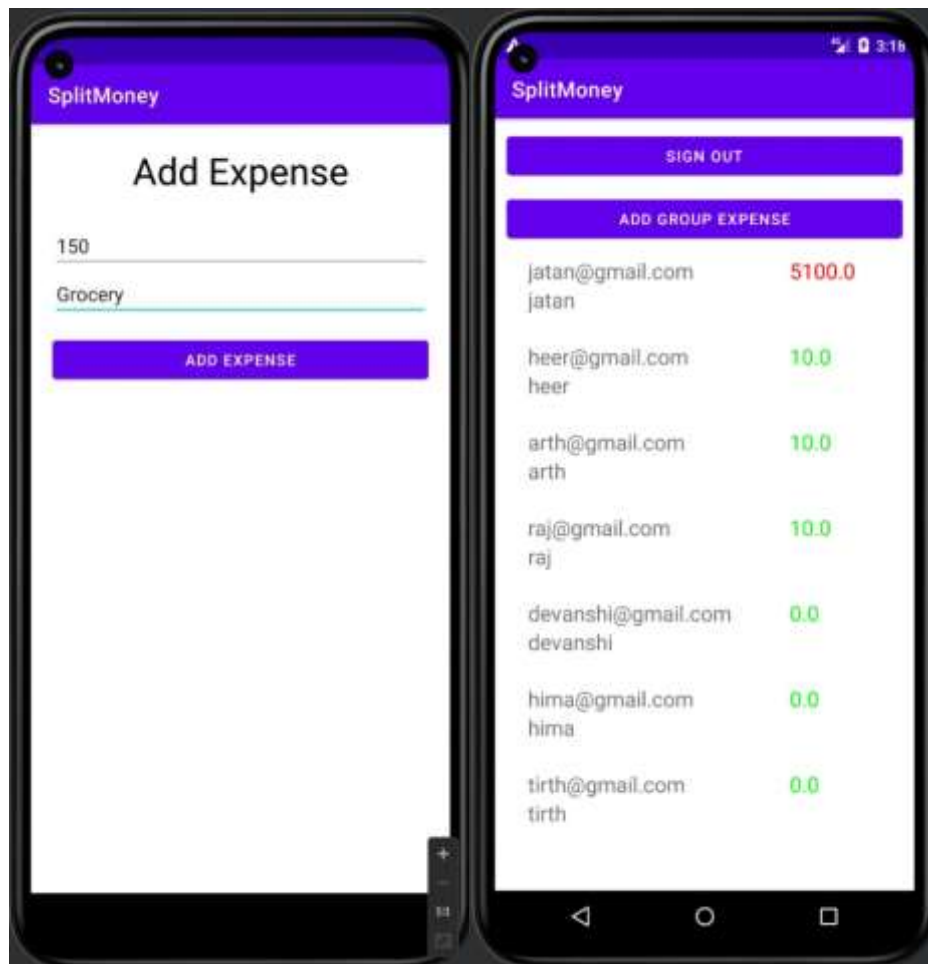
User login/register search API screenshot:

The screenshot displays a Java IDE with a REST API implementation. The code defines endpoints for user registration, login, and group management. The logs at the bottom show the application running successfully on a Spring Boot server, with messages indicating the start of the application and the successful execution of the API endpoints.

Login and Sign up screen shot



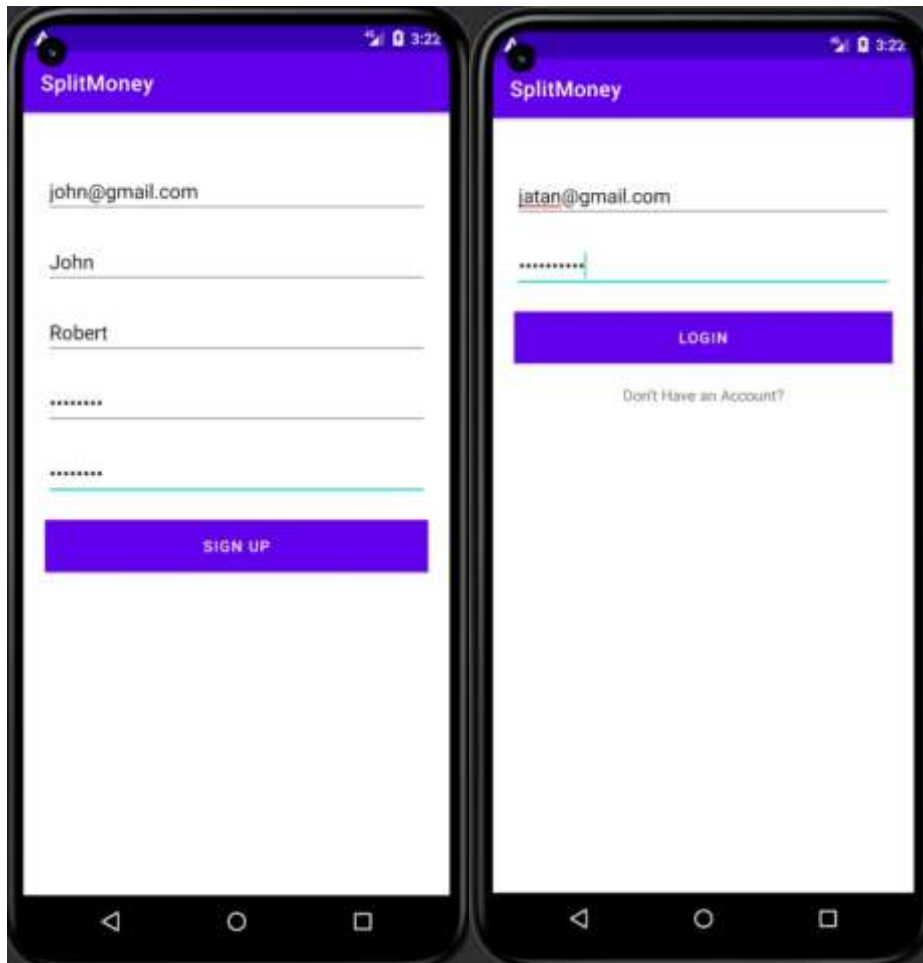
Expense management API screenshot to add Expense:



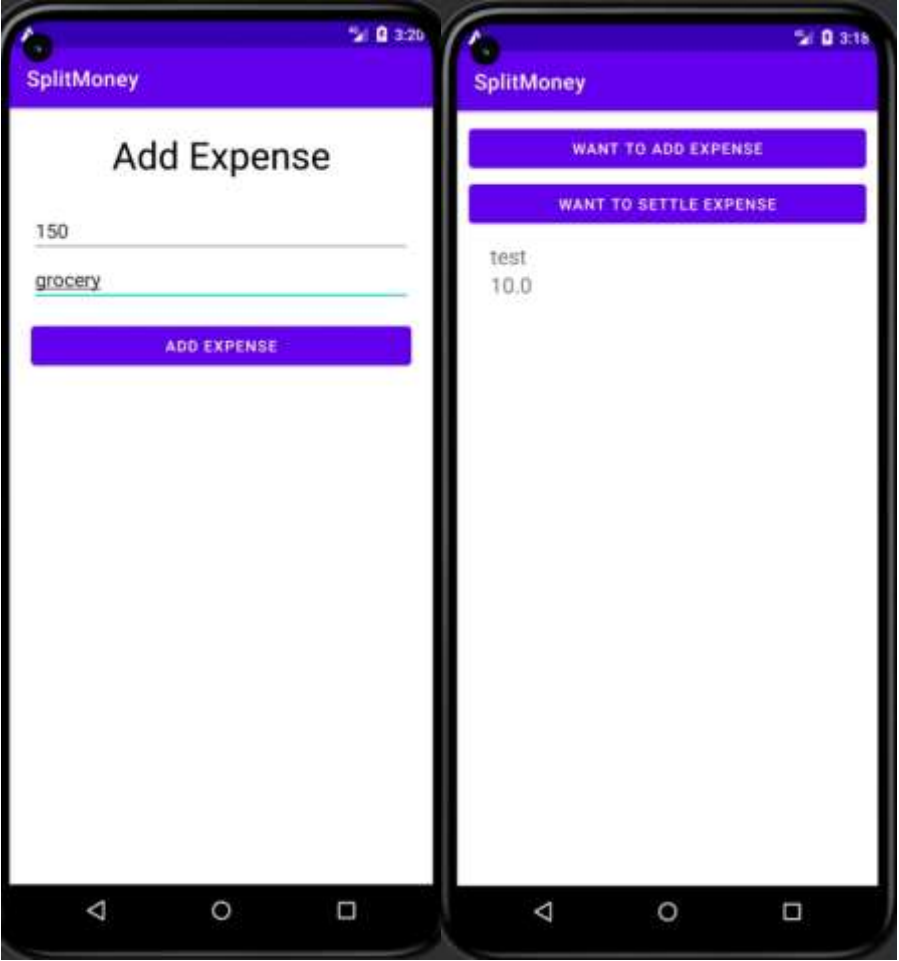
All new users must create an account; if they already have one, they can log in and access a list of all users to share expenses. From there, they can add expenses to any person they share costs with.

Screenshots for each functionalities: -

Sign Up & Login Page -



Add/Settle Expense – When we click on add Expense expense can be added and when we click on settle expense that particular expense gets settled.



Creating Group to split the Expense among group:



References: -

- Object-Oriented Software Engineering: An Agile Unified Methodology (Devid C. Kung 2014, 132)
- Object-Oriented Software Engineering: An Agile Unified Methodology (Devid C. Kung 2014, 183)
- Object-Oriented Software Engineering: An Agile Unified Methodology (Devid C. Kung 2014, 184)
- Object-Oriented Software Engineering: An Agile Unified Methodology (Devid C. Kung 2014, 187)
- Object-Oriented Software Engineering: An Agile Unified Methodology (Devid C. Kung 2014, 189)
- Object-Oriented Software Engineering: An Agile Unified Methodology (Devid C. Kung 2014, 190)
- Object-Oriented Software Engineering: An Agile Unified Methodology (Devid C. Kung 2014, 192)
- Object-Oriented Software Engineering: An Agile Unified Methodology (Devid C. Kung 2014, 195)
- Object-Oriented Software Engineering: An Agile Unified Methodology (Devid C. Kung 2014, 201)
- Object-Oriented Software Engineering: An Agile Unified Methodology (Devid C. Kung 2014, 209)
- Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. 3rd edition. Prentice Hall, Craig Larman, 2009

CSE 5324 : SFWR ENG-I ANALYSIS, DESIGN AND TESTING
Academic Integrity

In order for your Assignment/Homework/Project to be accepted you must read the following, sign this form and attach it to your papers (as the last page of your assignment).

Academic Integrity: Students enrolled in this course are expected to adhere to the UT Arlington Honor Code:

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

Student Signature: 
Name: Jay Patel

Student

Student Id Number: 1001870971

Date: 9/19/2022

CSE 5324 : SFWR ENG-I ANALYSIS, DESIGN AND TESTING
Academic Integrity

In order for your Assignment/Homework/Project to be accepted you must read the following, sign this form and attach it to your papers (as the last page of your assignment).

Academic Integrity: Students enrolled in this course are expected to adhere to the UT Arlington Honor Code:

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

Student Signature: 
Name: Heer Patel

Student

Student Id Number: 1001965102

Date: 9/19/2022


CSE 5324 : SFWR ENG-I ANALYSIS, DESIGN AND TESTING
Academic Integrity

In order for your Assignment/Homework/Project to be accepted you must read the following, sign this form and attach it to your papers (as the last page of your assignment).

Academic Integrity: Students enrolled in this course are expected to adhere to the UT Arlington Honor Code:

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

Student Signature: 
Name: Arth Modi

Student

Student Id Number: 1001979538

Date: 9/19/2022

CSE 5324 : SFWR ENG-I ANALYSIS, DESIGN AND TESTING
Academic Integrity

In order for your Assignment/Homework/Project to be accepted you must read the following, sign this form and attach it to your papers (as the last page of your assignment).

Academic Integrity: Students enrolled in this course are expected to adhere to the UT Arlington Honor Code:

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

Student Signature: 
Name: Jatan Bhatt

Student

Student Id Number: 1001927386

Date: 9/19/2022