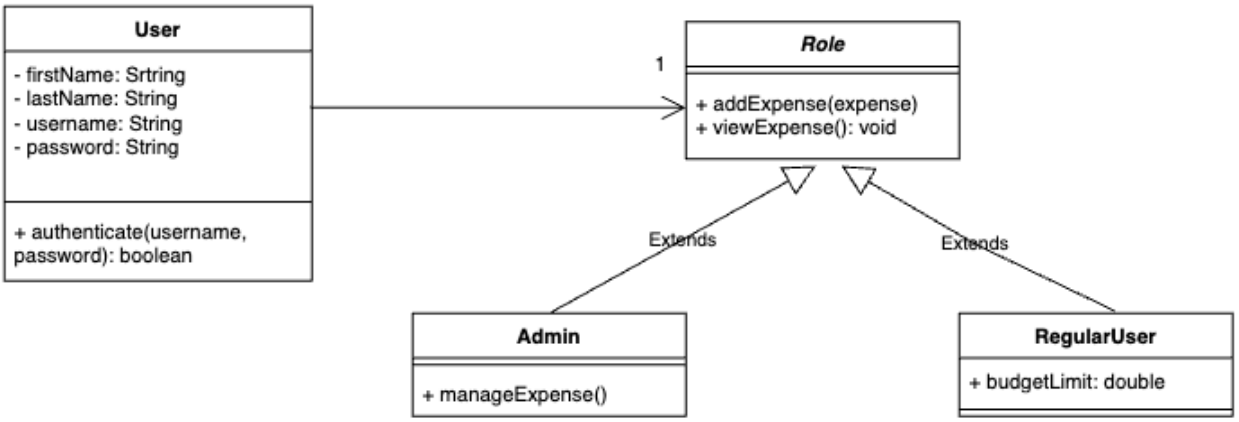
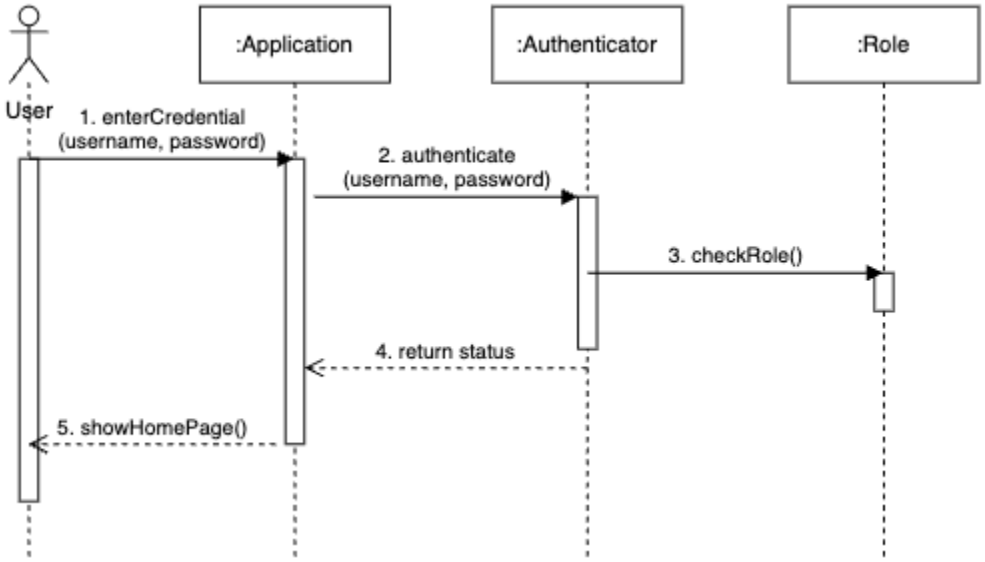
Project: week 1

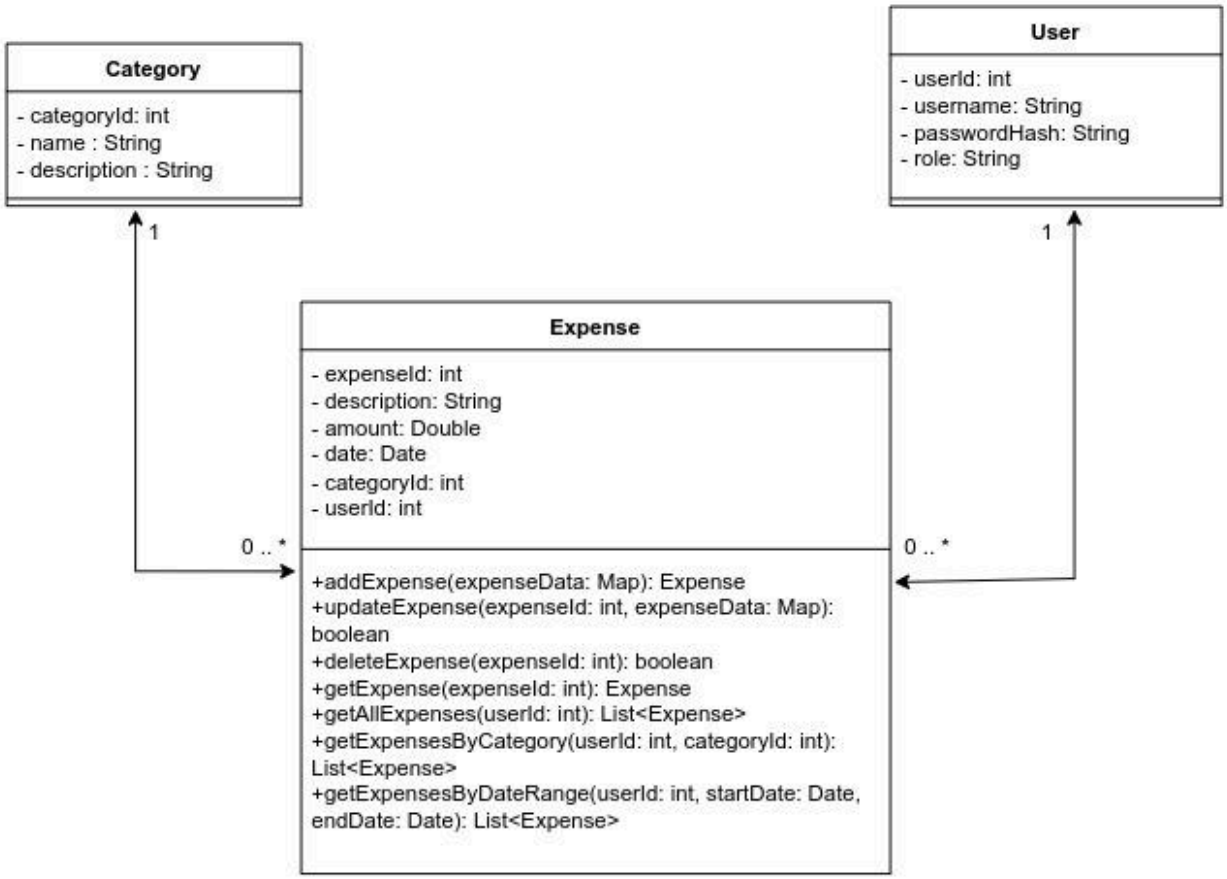
Features:

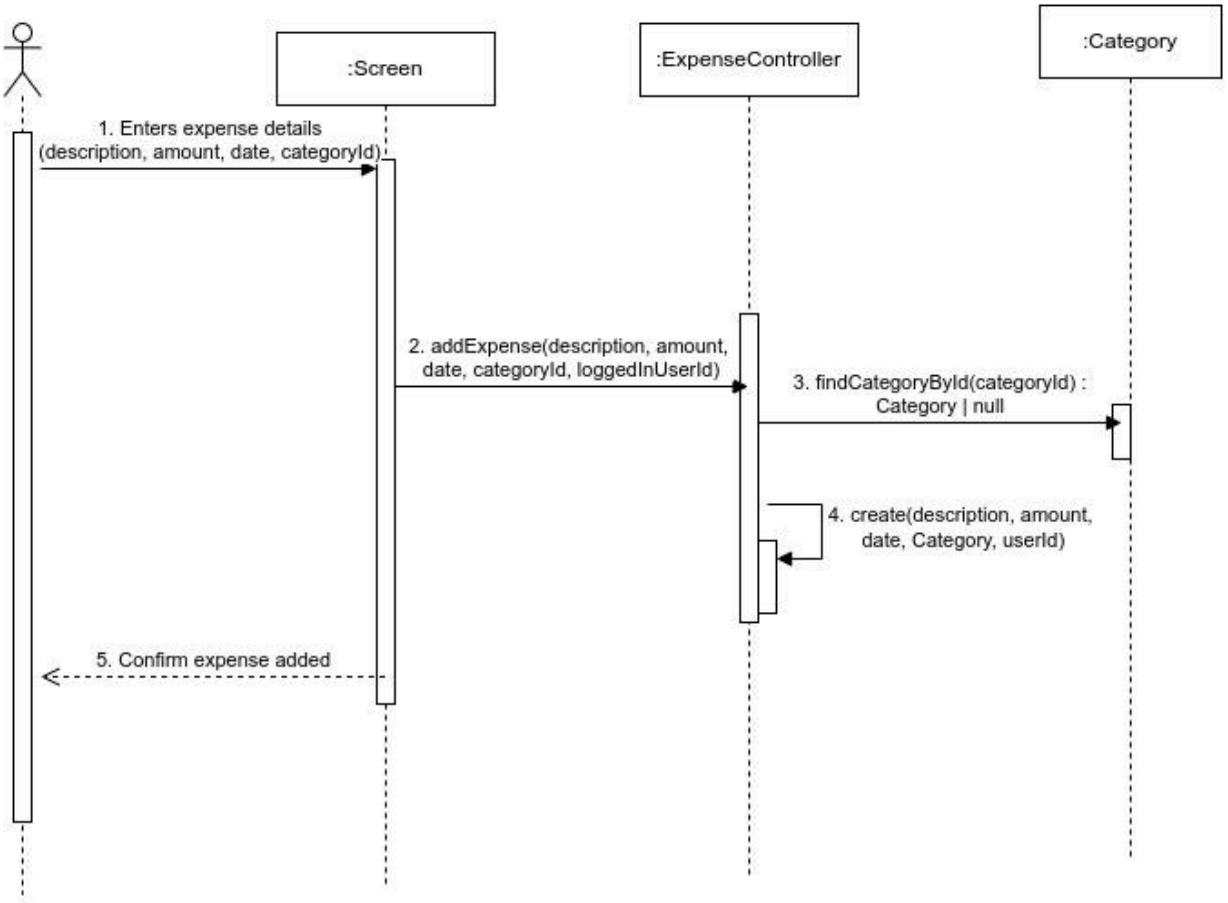
1. **User Authentication (Student: Cheikh Ahmadou Bamba Fame)**

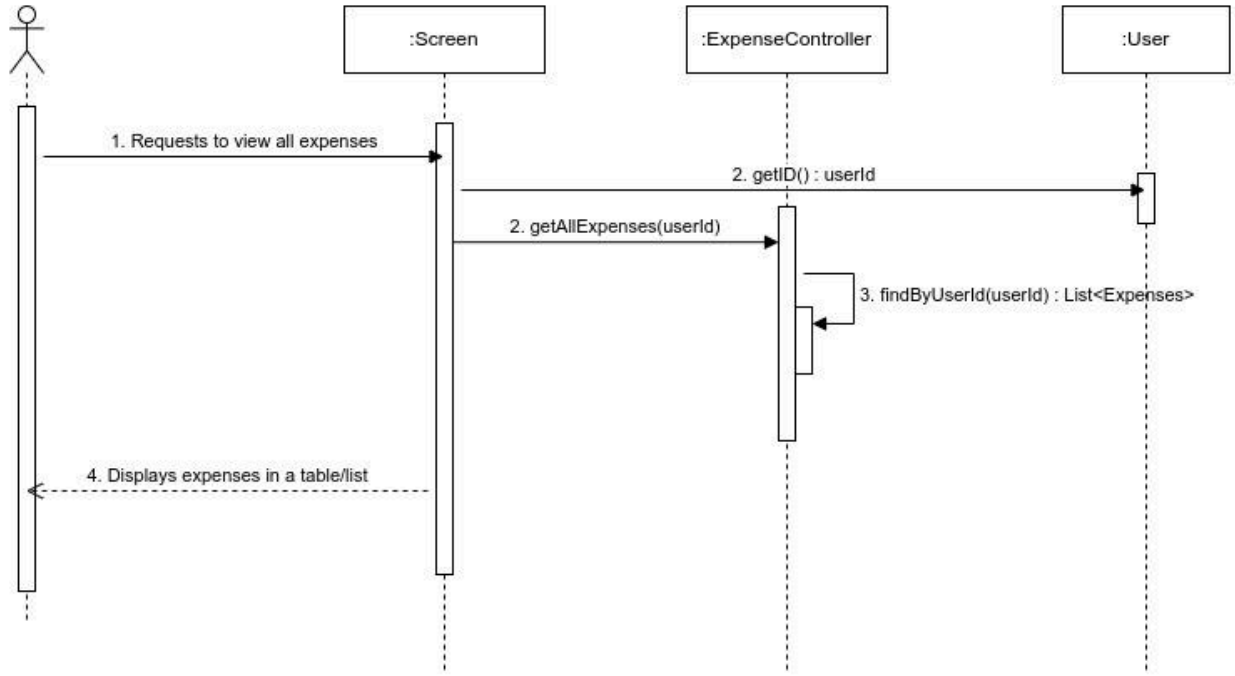




2. **Expense Management (Student: Yaakoub Kadri)**


Category
- categoryId: int
- name : String
- description : String
User
- userId: int
- username: String
- passwordHash: String
- role: String
1
1
Expense
- expenseId: int
- description: String
- amount: Double
- date: Date
- categoryId: int
- userId: int
0 .. *
0 .. *
+addExpense(expenseData: Map): Expense
+updateExpense(expenseId: int, expenseData: Map): boolean
+deleteExpense(expenseId: int): boolean
+getExpense(expenseId: int): Expense
+getAllExpenses(userId: int): List<Expense>
+getExpensesByCategory(userId: int, categoryId: int): List<Expense>
+getExpensesByDateRange(userId: int, startDate: Date, endDate: Date): List<Expense>


:Screen
:ExpenseController
:Category
1. Enters expense details
(description, amount, date, categoryId)
2. addExpense(description, amount,
date, categoryId, loggedInUserId)
3. findCategoryById(categoryId) :
Category | null
4. create(description, amount,
date, Category, userId)
5. Confirm expense added

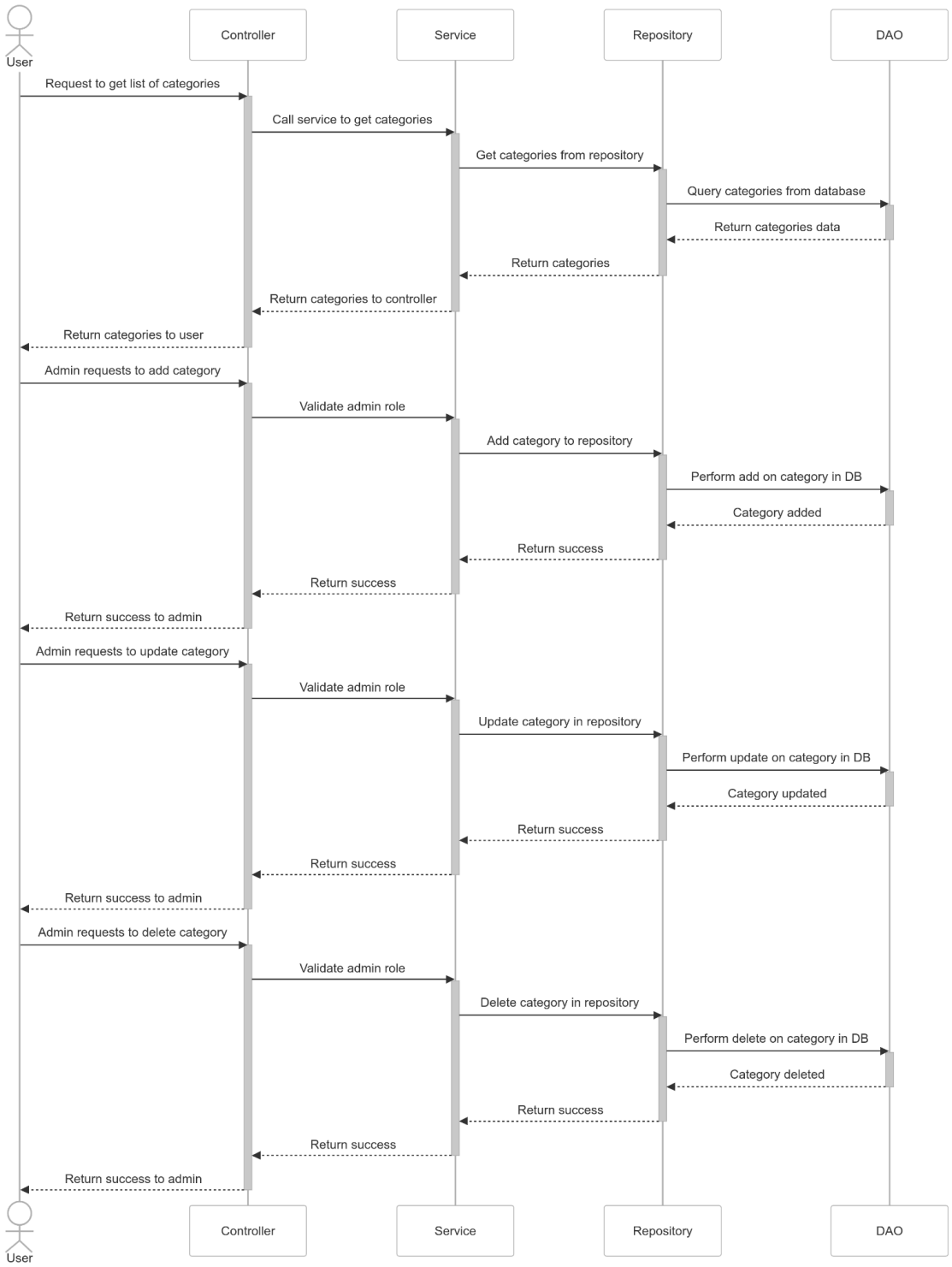

:Screen
:ExpenseController
:User
1. Requests to view all expenses
2. getID() : userId
2. getAllExpenses(userId)
3. findByUserId(userId) : List<Expenses>
4. Displays expenses in a table/list

This document presents the sequence diagrams for the Add and View expense use cases.

The Update and Delete operations follow a similar flow to Add, involving user interaction with the Screen, Controller, Category(and DB later).

As such, their diagrams were omitted to avoid redundancy and maintain clarity.

3. **Category Management (Student: Moges Tesfay Yohannes)**


User
Controller
Service
Repository
DAO
Request to get list of categories
Call service to get categories
Get categories from repository
Query categories from database
Return categories data
Return categories
Return categories to controller
Return categories to user
Admin requests to add category
Validate admin role
Add category to repository
Perform add on category in DB
Category added
Return success
Return success
Return success to admin
Admin requests to update category
Validate admin role
Update category in repository
Perform update on category in DB
Category updated
Return success
Return success
Return success to admin
Admin requests to delete category
Validate admin role
Delete category in repository
Perform delete on category in DB
Category deleted
Return success
Return success
Return success to admin
User
Controller
Service
Repository
DAO