BANKING MANAGEMENT SYSTEM

PREPARED FOR

JEE full Stack 2.0 react

Training program

Spring 1 project

Capgemini Technology Services

PREPARED BY

Harshit Dawane

Ishika Galav

Khushi Kala

Pralipta Pridarshniya Jena

Tulsi Sethiya

Yogesh Waiker

Abstract:

The bank management system project is a program that keeps track of a client’s bank account. This project demonstrates the operation of a banking account system and covers the essential functions of bank management software. It develops a project for resolving a customer’s financial applications in a banking environment to meet the needs of an end banking user by providing multiple ways to complete banking chores. Additionally, this project is to provide additional features to the user’s workspace that are not available in a traditional banking project. The project’s bank management system is built on cutting-edge technologies. This project’s main goal is to create software for a bank account management system. This project was designed to make it simple and quick to complete previously impossible processes with manual systems which are now possible with this software.

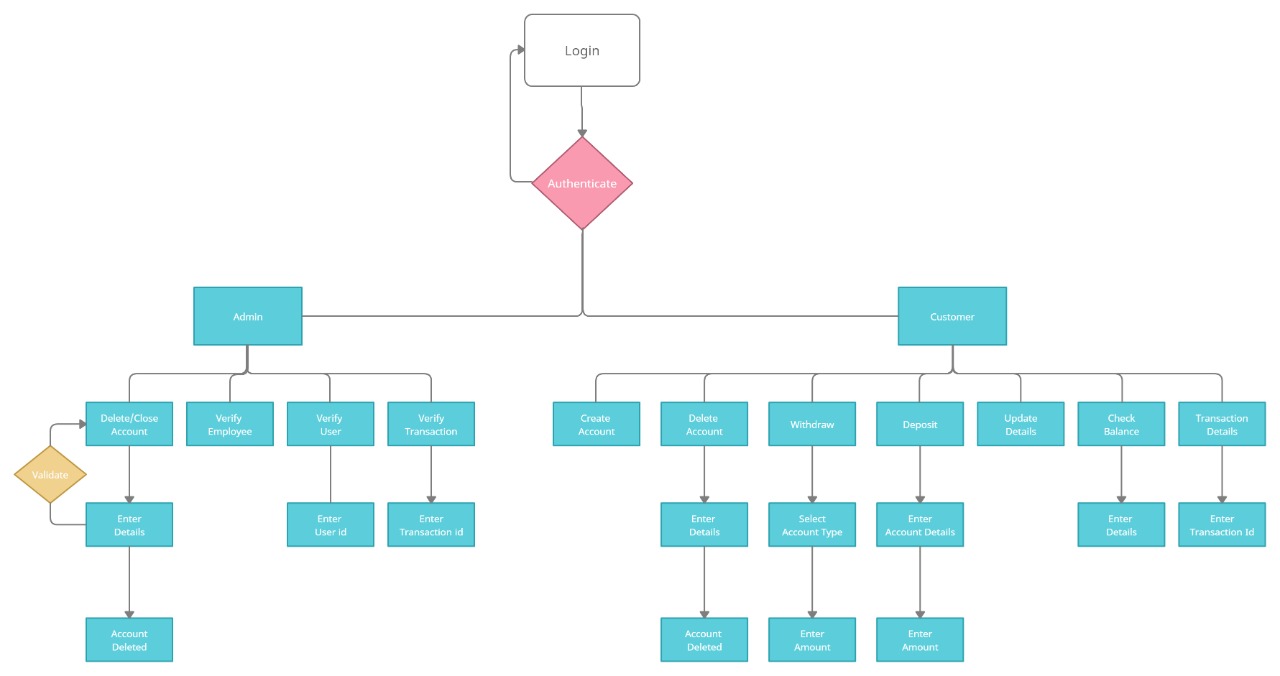
Scope:

Following are the functionality provided by the system.

There are two categories of people who would access the system : customer and administrator. Each would have some exclusive privileges.

1. The Customer can
2. Create user account.
3. Login into the application.
4. Make transaction.
5. Check balance.
6. Edit or modify detail.
7. The administrator can:
8. Login into the application
9. Add and Delete account
10. Manage transaction

Flow chart:



Sequence Diagram for User:

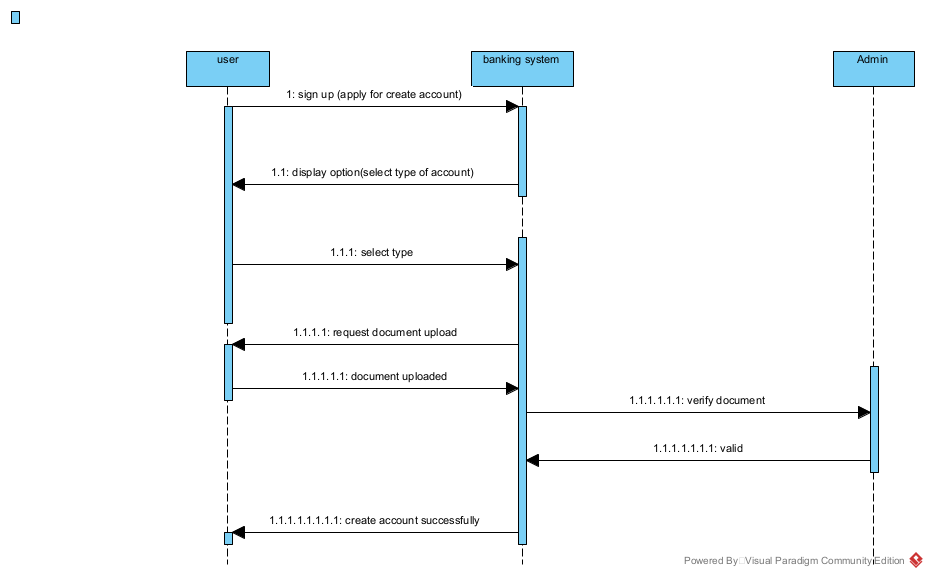
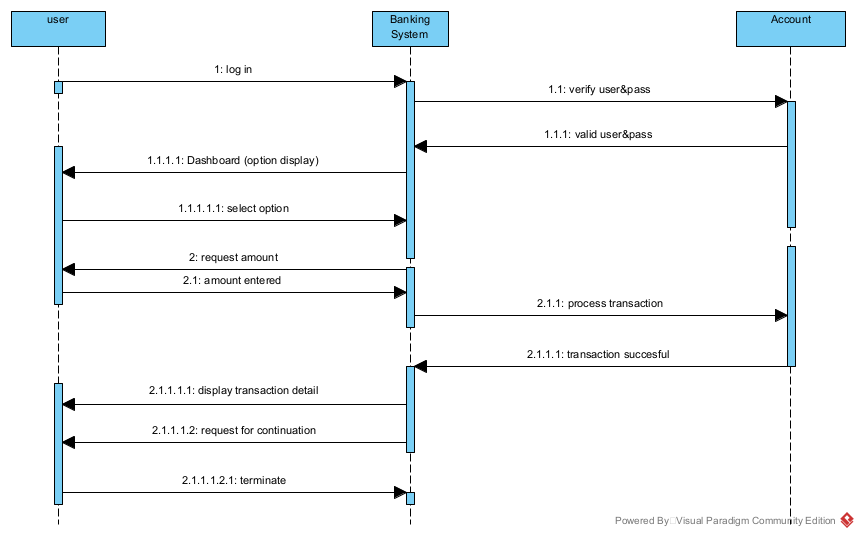
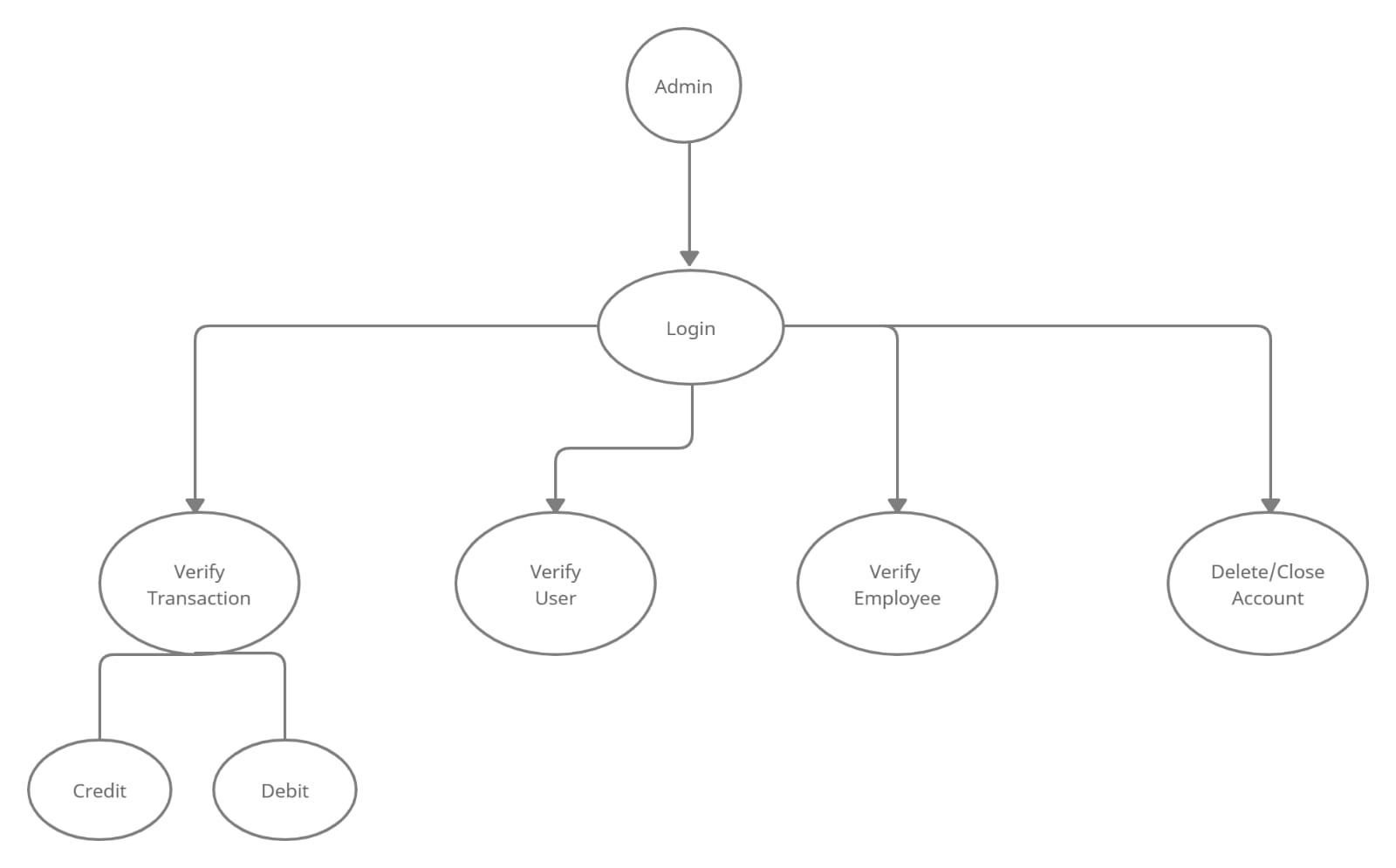


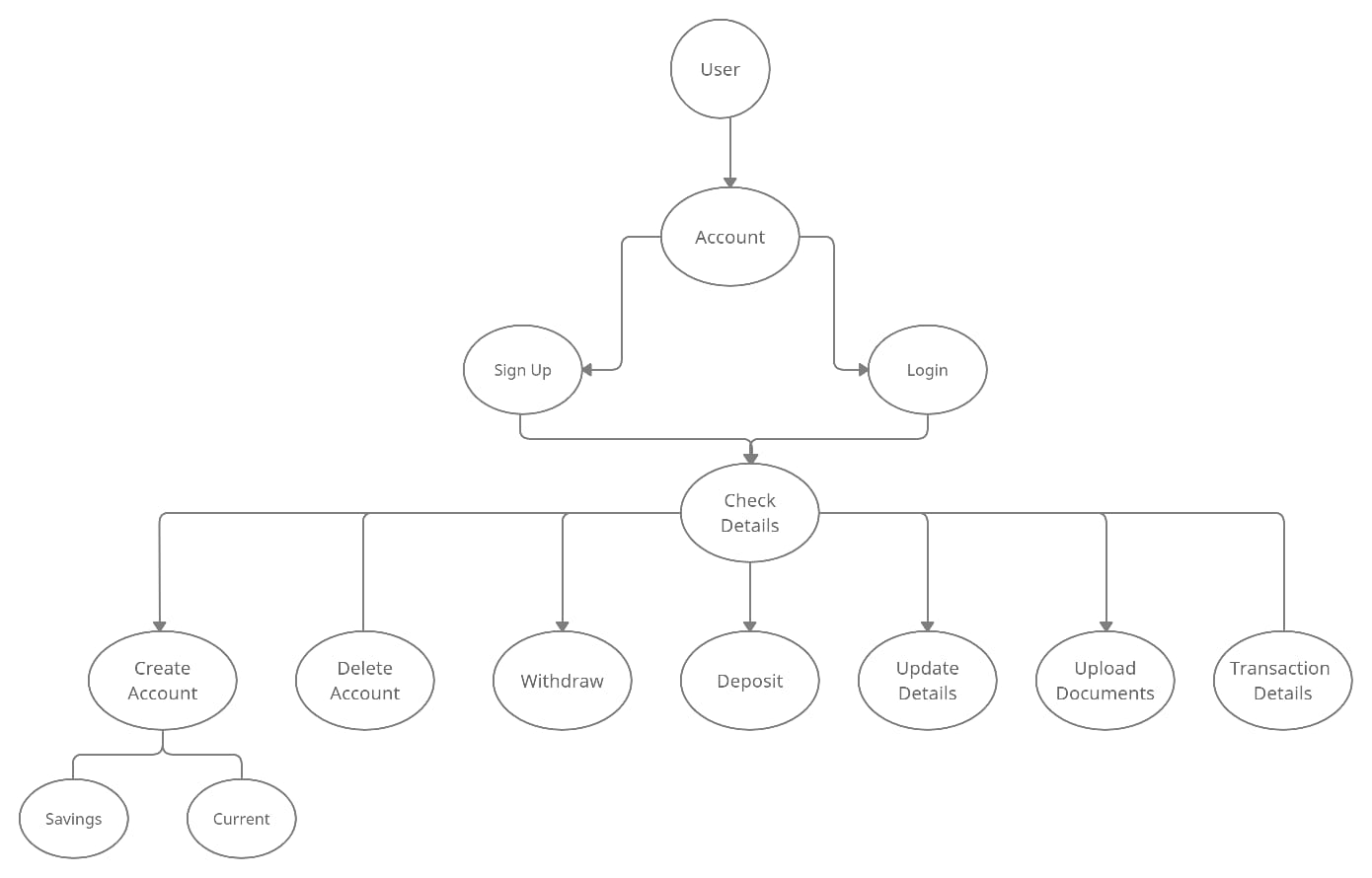
Diagram 2:



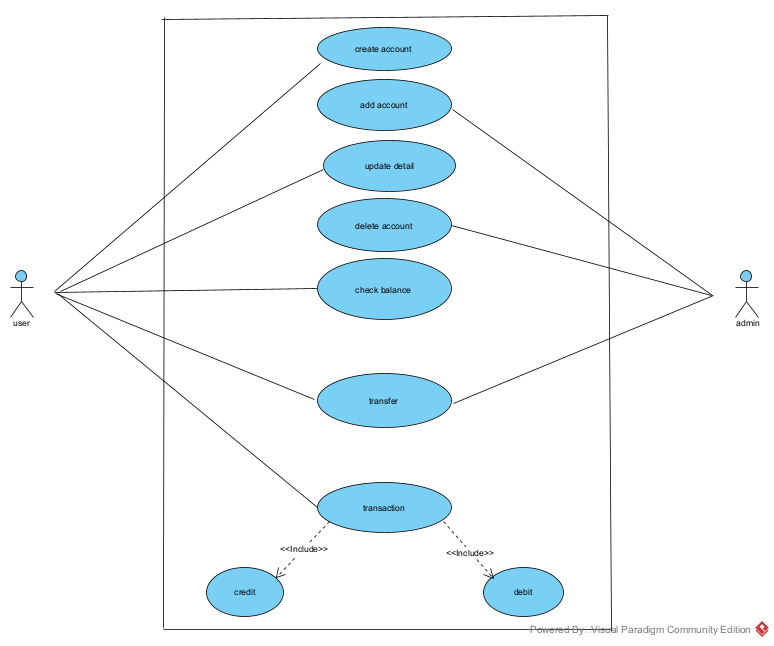
Activity Diagram for Admin:



Activity Diagram for User:



Use Case Diagram:



Class and Method Description:

DTO layer:

1) Account Detail

Attributes:

Contact: Big Tnteger

Account number: Big Integer

Account Type: String

Account Balance: double

2) Address

Attributes:

Contact: String

city: String

State: String

Pin: Integer

Address line 1: String

Address line 2: String

3)Contact us

Attributes:

name: string

email: string

message: string

contact number: integer

4)counter

Attributes:

name: string

count: integer

5)Counter Details

Attributes:

name: string

availability: string

assigned-counter: string

status: string

mobile no: string

token no: integer

6)Employee Details

Attributes:

name: string

email: string

contact: Big integer

password: string

assigned counter: string

7) Deposit

Attributes:

date: string

account no: Big integer

deposite amount: double

account type: string

8) User Detail:

Attributes:

first name: string

last name: string

middle name: string

contact: big integer

email: string

password: string

aadhar card number: string

9) Response Detail

Attributes:

status code: integer

message: string

time: date

object: List

10) Transaction:

Attributes:

Transaction\_id: string

Transaction\_amount: double

Transaction\_date: string

Transaction\_from: big integer

Transaction\_to: big integer

Transaction\_fromtype: string

Transaction\_toType: string

11) Withdraw:

Attributes:

date: string

account no: big integer

withdraw amount: double

account type: string

12) User Login

Attributes:

email: string

password: string

time: string

Controller Layer :

1) Admin Controller

Method:

createAdminProfile ()

editPassword (Old Password, new Password) ; string

employeeRegister ( Employeedetail) : Employees detail

adminLogin (adamin): Map<String, string>

customerAppointment (customerAppointment): customerAppointment

employee Details (): Map <string, string>

createCounterDetails (counterDetails): CounterDetails

getdetailByDate (Counter Details): CounterDetails

getDetail (counterdetail): Counterdetails.

2) contact Us Controller

Method:

Saveuserdata ( ContactUs): ContactUs

ContactUs Details ( String): list <Contact Us >

3) EmployController

Method:

deposite Amount (Deposit) : Deposit

withdraw Amount (withdraw): withdraw

4) Transaction Controller

Method:

Transfer Amount (transaction) : transaction

update details (AccountDetail sender, AccountDetail reciever)

5) User Detail Controller

Method:

Create User (user detail); Map <string string>

create Saving Account (user detail) : String

Create current account (userDetail): String

updateUser(userDetail):userDetail

6) userlogin controller

Method:

Login user ( user detail) : map <string, string>

Service Layer:

1. DepositServiceImpl

Method:

Save(Deposit):Deposit

1. TransactionServiceImpl

Method:

Save(Transaction): Transaction

1. WithdrawSetrviceImpl

Method:

Save(Withdraw): Withdraw