**Online Bank Management Application**

**Tools and Technologies**

* Front-End: Html, CSS, bootstrap, Angular 8
* Frameworks: Spring boot, Spring MVC, Spring Data, Spring security.
* Technology/Domain: Java
* Database: MYSQL
* Server: Tomcat 9.5

**Objective of Bank Management application**

The bank is a major industry and still at some stage banks are doing some manual processes. So it’s really good to automate some manual processes that can help bankers and customers.

So this banking management system is a web application to manage bank activities. Where we can manage the customers,bank employees and Transection histories, Also a customer can do online payments.

**There are three types of actors in the application. Bank Manager, Bank Employee, Customer**

**Bank Manager:**

Manager is the root user of the application. The is the admin of the application

**He/she can do following activities:**

* The manager can ADD/EDIT/UPDATE bank accounts to the bank.
* manager can ADD/EDIT/UPDATE bank employee
* Bank manager authorized an bank employee (clerk) to do bank transactions
* Bank manager will manage all the online payment logs.
* When a new user will be added to the bank. A unique account number will generate automatically for the customer that the user will use during the transactions.

**Bank Employee:**

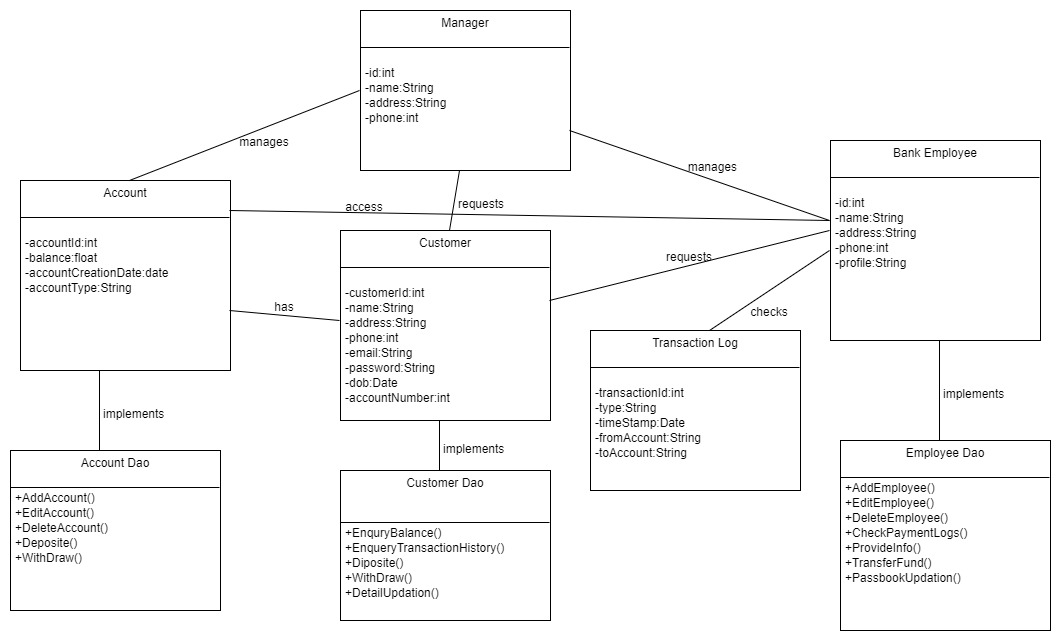
* The bank employee can do transactions as per the customer’s request.
* He/she can transfer fund, deposit amount, withdraw amount for an customer
* He can print customer transaction details on customer passbook

**Bank Customer:**

* Customer can login into the application once approved by bank manager
* Customer can check his bank balance and transaction history
* Customer can send money to another Customer
* The Customer can update his address details subjected to manager approval

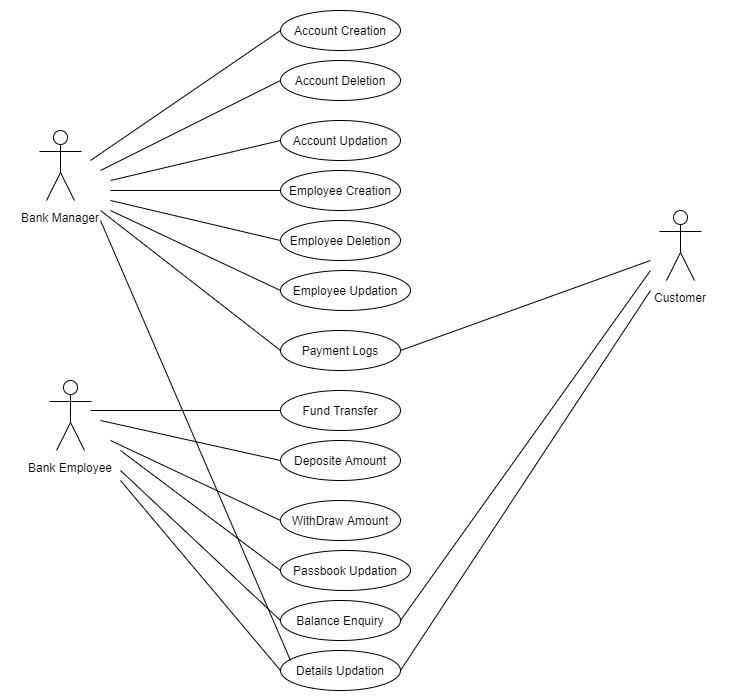
**ClassDiagram:**

Class diagram a type of **static structure diagram** that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

****

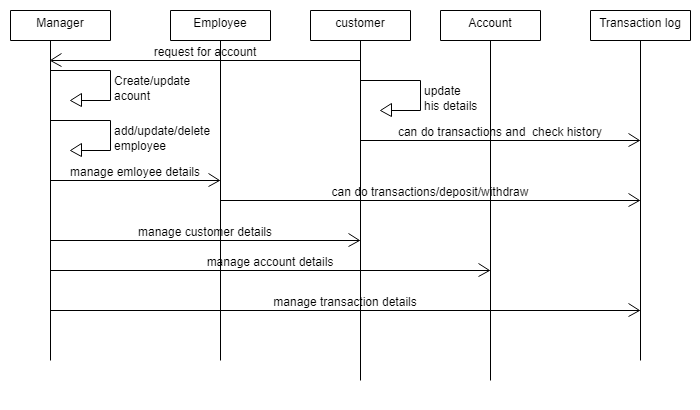
**UseCase Diagram:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases  in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.



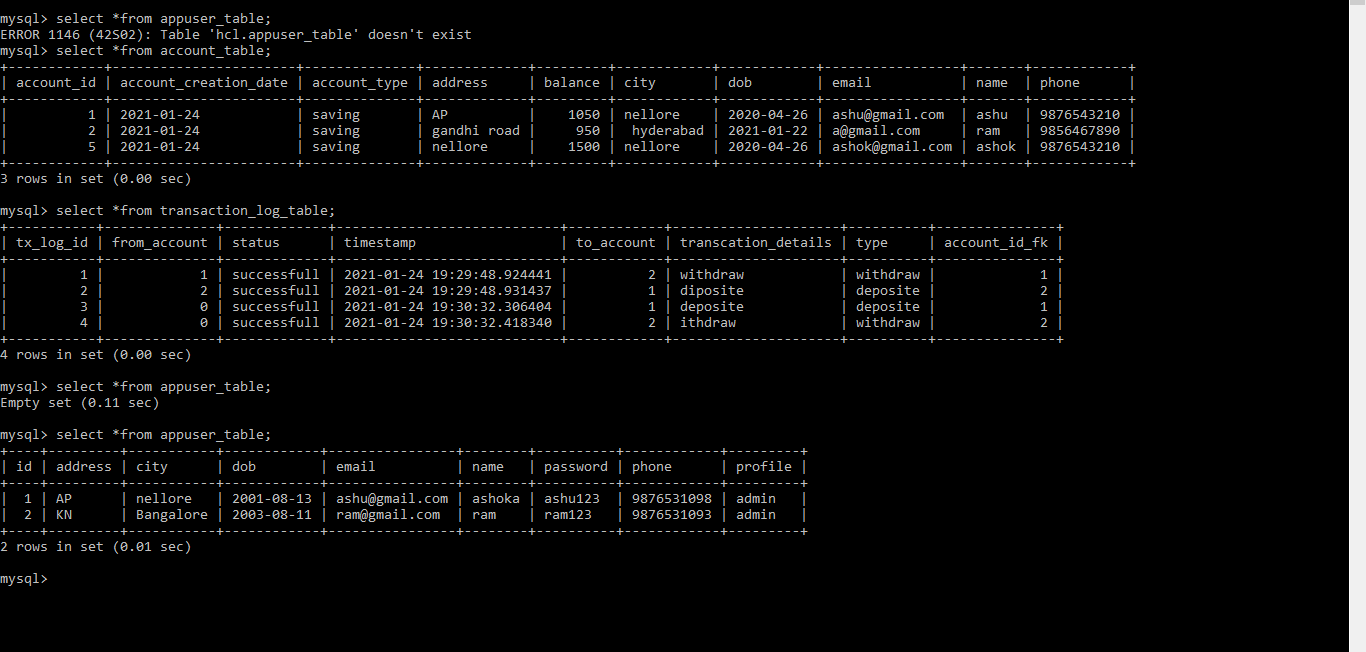
**Sequence Diagram:**

A sequence diagram shows, as parallel vertical lines, different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

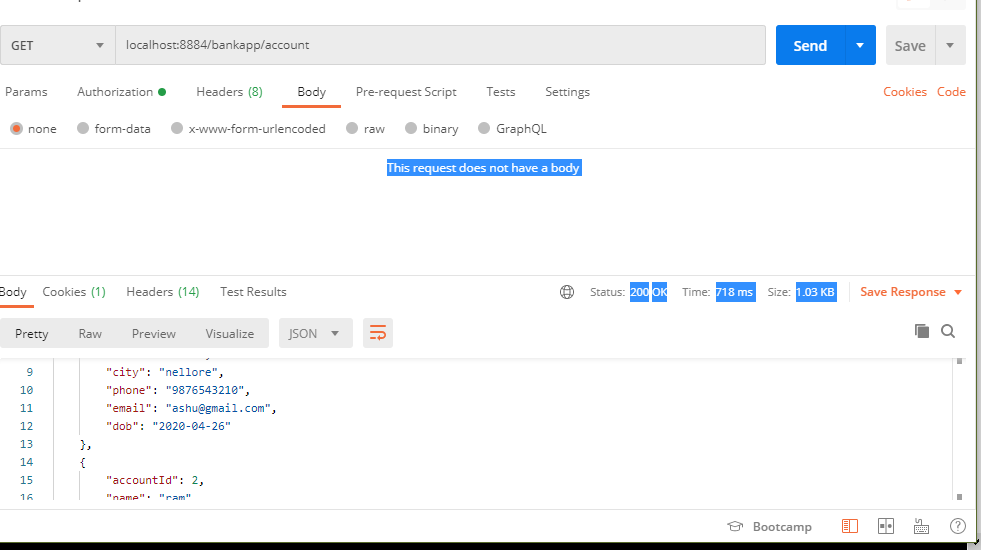


**Results:**

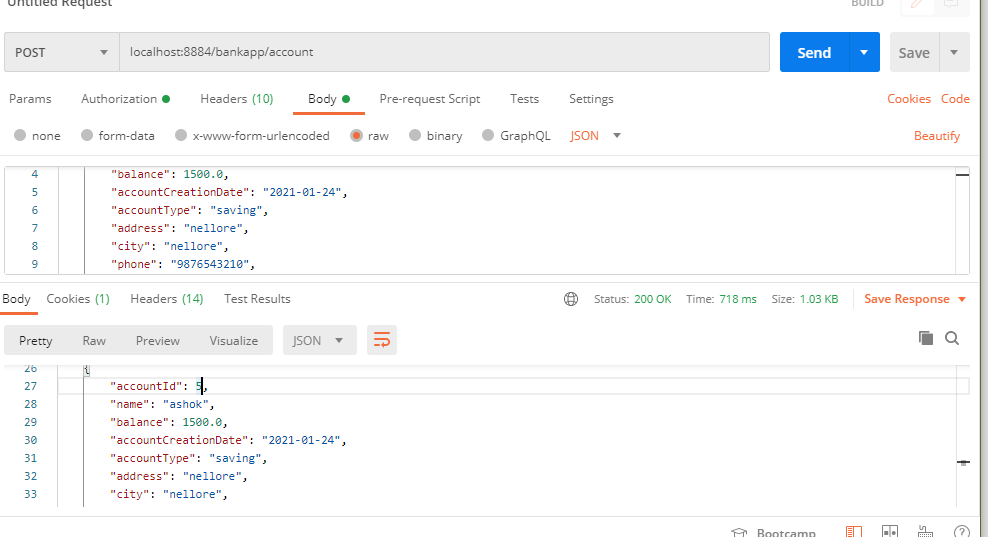
Database Tables:

****

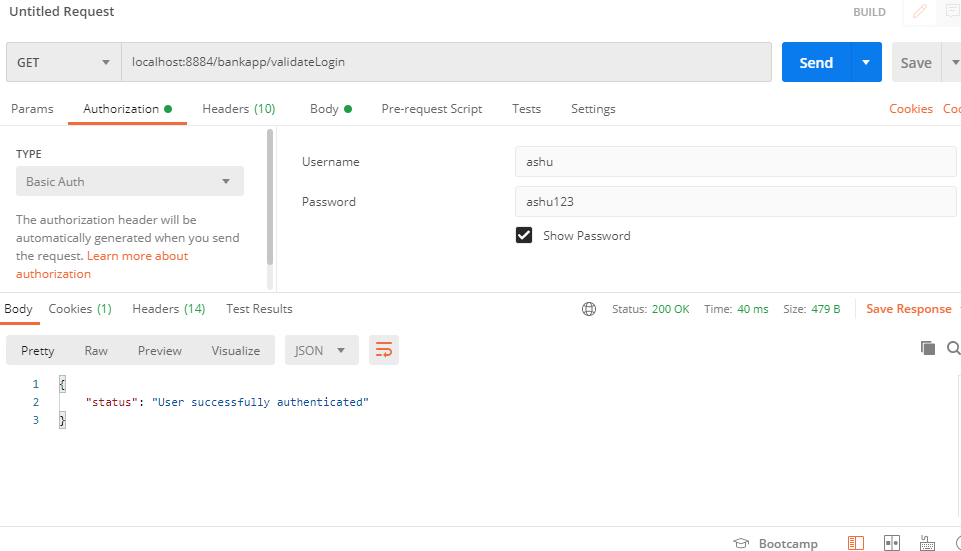
PostMan:

By using localhost:8884/bankapp/account we can get all the records

Using POST Method We can add the data to the server

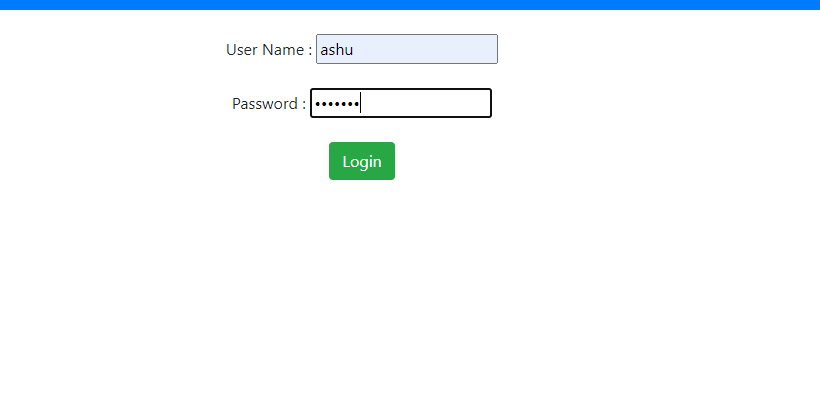
****

I f use security we need to pass the password and user name to the server ,after passing the values we can modify the details as well as we got all the access

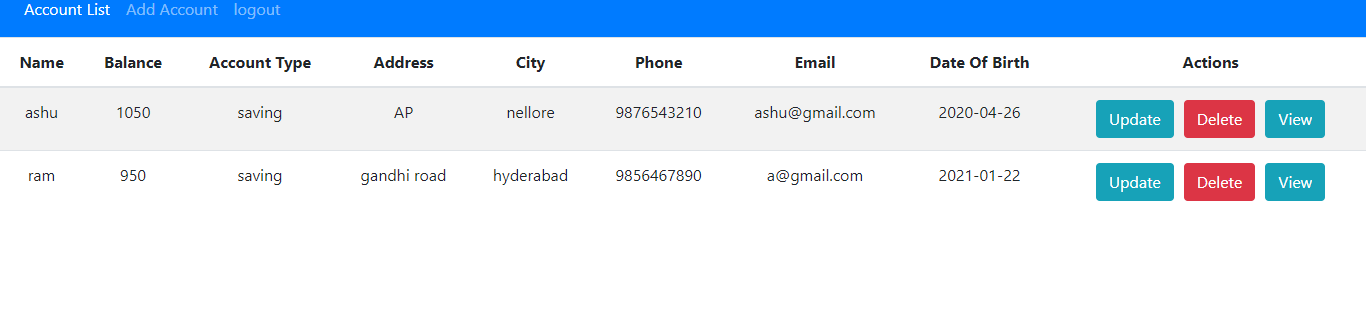
****

Angular Side:

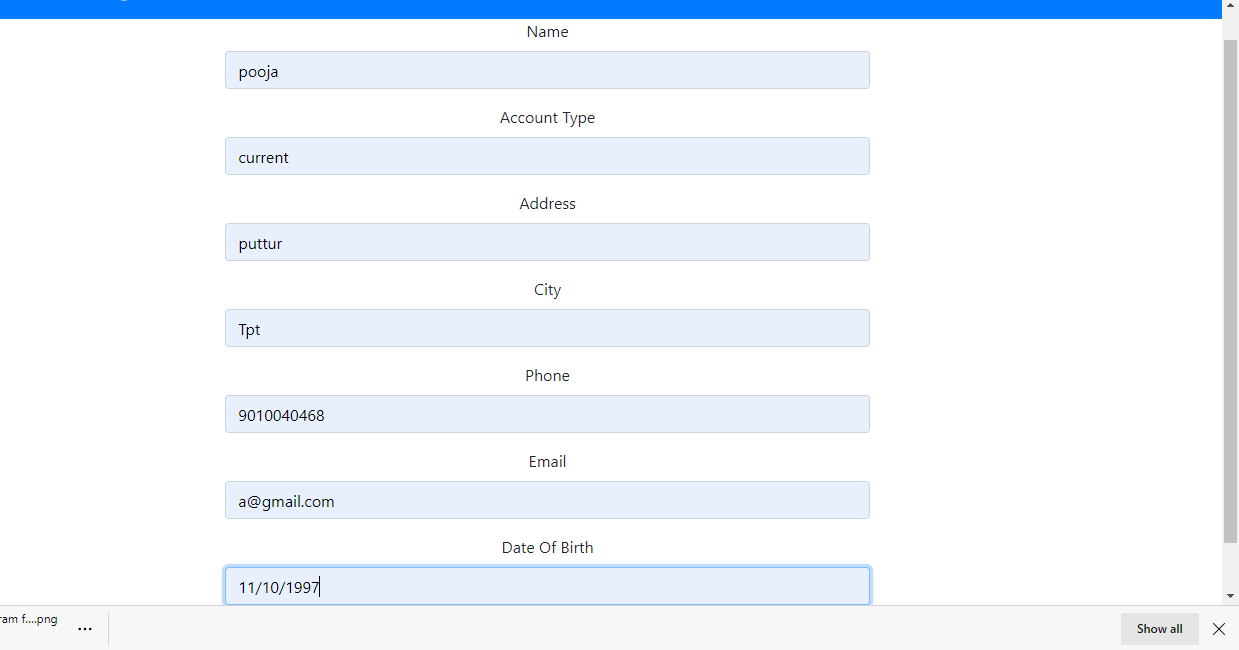
After applying security to angular we can login to the portal using credintials

****

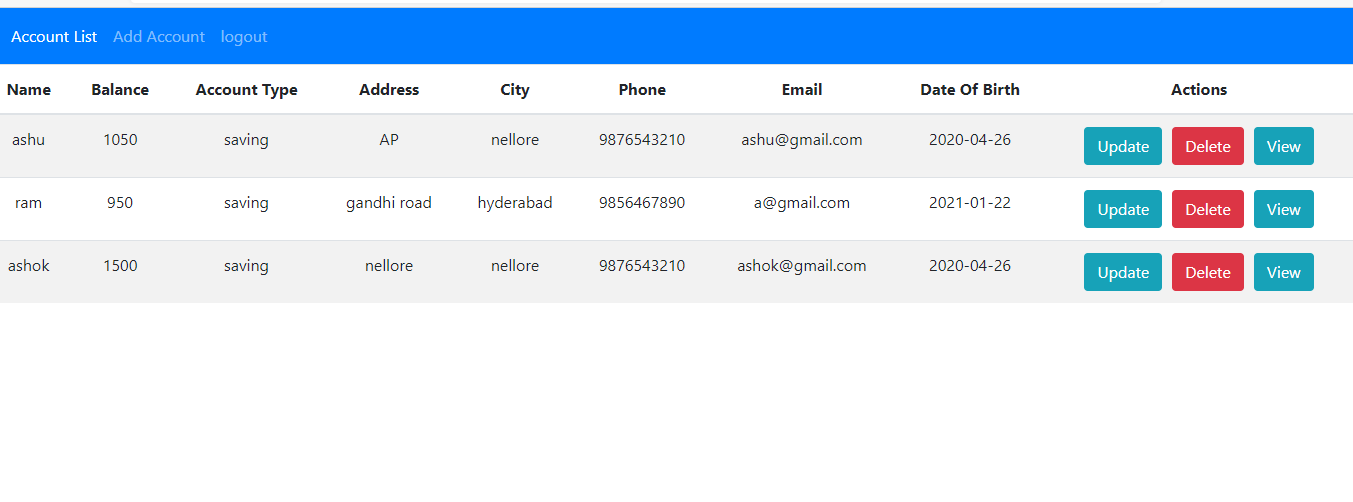
**After login we can see this details**

****

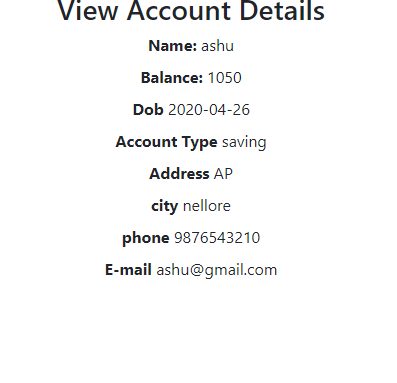
Add method adds the another record to the server



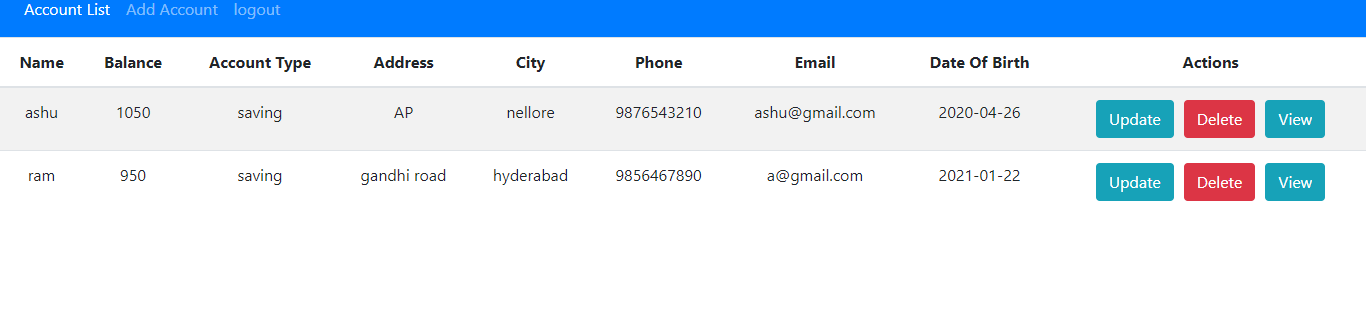
After add the details of another user record data will be dispaly like this

****

By using view action we can see the particular details of the user

****

delete action used to delete the data from the server .After deleting the one record it shows like this

****