# BANKING MANAGEMENT SYSTEM

A Banking Management System is a software built to handle all online bank activities easily and safely. This System will give the bank management power and flexibility to manage the entire system from a single portal. The System allows the manager to keep track of all the users in the system as well as their transactions.

## System Requirements:

We'll focus on the following set of requirements while designing the Bank Management System:

1. The system shall support the creation of new accounts.
2. Registered users should be able to log into their account.
3. The system should be able to store information regarding the user's account details.
4. The system should also support the deletion of accounts.
5. The system should allow the admin to look at the list of all the user accounts.
6. The system should provide the users with an interface to deposit/withdraw money, transfer money, check balance etc.
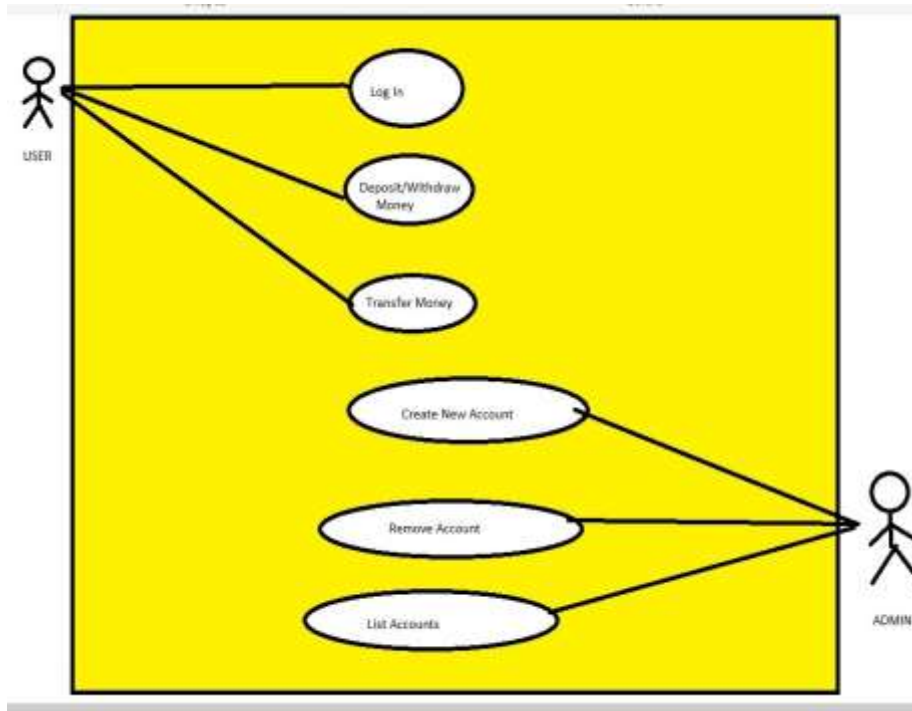
## Use Case Diagram:

Here are the main actors of our system:

1. User: All users can log in into the system.
2. Admin: Admin can add users, remove users and view user details.

Top Use cases:

1. Create User Account
2. Log In
3. Deposit money
4. Withdraw money
5. View Balance
6. Delete Account

## CLASS DIAGRAM:

1. User: This class will be working to get the login details of the user.

2. Admin: This class will help in listing all the accounts stored, removing or creating account and also will take care for the password authenticity entered by the user to access his account.

3. Account: Will work to fetch the normal account data from the user.

4. SavAccount: A class executing functions to create saving account.

5. CurrAccount: A class executing functions to create saving account.

ACCOUNT
- balance: double
- accountNumber: string

- Account(double balance)
- ~Account()
- deposit: void
- withdraw:void
- transfer:void

"has-an"

USER
- userId: string
- passwordhash: long long
- account: Account*

- User()
- ~User()
- updateFile: void
- hash(string pwd): long long

ADMIN
- listAllAccounts(): void
- addNewAccount():void
- removeAccount():void
- authenticateUser():bool

"is-a"

savingAccount
- withdrawlimit: double

"is-a"

currAccount
- minBalance: double