# Introduction

## Problem Statement

Create a typical consumer online banking application that provides for a customer, an account and a transaction. The functions that should be included in the application are:

1. Customers should be able to **create** an *account* with the bank
2. Customers can specify the *amount* to **lodge** with the *card* that will be debited
3. Customers can specify the *amount* to **transfer** and the *account* to transfer to
4. Customers can specify the *amount* to **withdraw** and the *card* that will be credited
5. Customers can request to view their **balance** on any of their *account(s)* at any time

## Proposed Solution

We will be using Netbeans to create a RESTful API to satisfy the problem statement. MySQL will be used to host the database required for the API and will be described in the Entity-Relationship diagram below.

We will create a server side application that allows customers to interact with the API on the website, that will be designed using HTML and CSS codes.

## Entity Relationship Diagram

C:\Users\User\Downloads\Untitled Diagram (1).png

**Customer –** Is a person or company with personal details, can hold one or more accounts and can make different transactions.

**Account –** holds information about type of account and balance after and before movements.

**Transaction –** describe what type of movement is the customer doing (withdraw, transfer, deposit), give us a description of the movement, the amount and date.

## Security Concerns

As we are talking about a Banking system is a priority protect the user’s information, is very important develop a way to protect all personal information provided by users. To make this possible, this application is going to have an authentication system.

Normally the process of authentication consist of two phases:

* Identification: Which provides user identity to the security system (User ID). What the system is going to do is search through the databases to find the specific user. Once this user is located, will be able to be identified.
* Authentication is a process in which the credentials that the user provide previously are compared with a database of authorized users. If the credentials match, the user will be able to access into the system.

Once the user has been authenticated will be able to launch other process inside the system.

To make the authentication we can use different techniques such as:

**Password requirements,** this means enforcing the password with especial requirements such as:

* Minimum 6 Characters
* Must contain at least 1 lowercase letter (a-z)
* Must contain at least 1 lowercase letter (A-Z)
* Must contain at least 1 number (0-9)
* Must contain at least 1 special character (!,@,#,$,%,etc)

**Hashing passwords,** to make this possible, is necessary use bcrypt which is a platform file encryption utility based on the Blowfish symmetrick block cipher cryptographic algorithm and can be used in all supported operating systems. Passwords must have between 8 and 56 characters.

## The RESTful API:

Function: SignIn

Description: allow user to signin the account

URI: BankingSystem/api/customer/login

HTTP: Get

Parameters: email, passcode

Resource contents: accepts users input, email and passcode and find the correct user if user lognin successfully go to home.

Pre-Conditions: User must has register in JJ’s Bank

Post-Conditions: Allows user to do transaction functionalities

API Name: Create Account

Description: Allows the addition of a new account into the system

URI: /BankingSystem/api/bankAccount

HTTP: GET and POST

Parameters: cusId

Resource contents: Boolean status, create account or unable to create account

Pre-Conditions: User must has register in JJ’s Bank

Post-Conditions: allows user to do transaction functionalities

API Name: Withdraw

Description: Allows user to take money from the account.

URI: /BankingSystem/api/transaction

HTTP: SET and GET

Parameters: cusId, from, to

Resource contents: user

Pre-Conditions: User must has enough funds in its account

Post-Conditions: Balance will be updated

API Name: Transfer

Description: Allows user to send money to another account

URI: /BankingSystem/api/transaction

HTTP: SET and GET

Parameters: cusId, from, to

Resource contents: user

Pre-Conditions: User must has enough funds in its account, have a destination account

Post-Conditions: Balance will be updated.

API Name: Deposit

Description: Allows user to make a deposit

URI: /BankingSystem/api/transaction

HTTP: SET and GET

Parameters: cusId, from, to

Resource contents: user

Pre-Conditions: Must exist a registered account in JJ’s Bank

Post-Conditions: Balance will be updated.

API Name: Balance

Description:

URI:

HTTP:

Parameters:

Resource contents:

Pre-Conditions

Post-Conditions:

API Name:

Description:

URI:

HTTP:

Parameters:

Resource contents:

Pre-Conditions

Post-Conditions:

API Name:

Description:

URI:

HTTP:

Parameters:

Resource contents:

Pre-Conditions

Post-Conditions:

API Name:

Description:

URI:

HTTP:

Parameters:

Resource contents:

Pre-Conditions

Post-Conditions:

API Name:

Description:

URI:

HTTP:

Parameters:

Resource contents:

Pre-Conditions

Post-Conditions:

API Name:

Description:

URI:

HTTP:

Parameters:

Resource contents:

Pre-Conditions

Post-Conditions: