

**Department: computer and systems**

**Banking System Document**

**Class Code: CSE 321 Software Engineering**

**Submitted By :**

Eman Mohamed Abd El Mohsen 1600354

Zeinab Ismail 1600613

Fatma Elzhraa Mohamed Abd Elhamid 1600968

Mostafa Ashraf 16t0107

Ali Alam Eldein 1600841

AbdElhamed Khaled 1600724

Abdallah Amr Mohamed 1600769

**Abstract:**

Before the evolution of computers, all the details in a banking system used to be maintained manually. This is not advisable because maintenance and retrieval of information becomes tedious. Our project on ‘Banking System’ is a fully computerized one which provides an account holder to view his account information such as checking balance, recent transactions, transfer of funds, etc. User can also view his account information on his personal computer.

**Table of Contents:**

|  |  |
| --- | --- |
| Page Number | Contents |
| 3 | Introduction |
| 3:5 | General Description |
| 6 | system Requirement |
| 7 | Use-Case Diagram |
| 8:15 | Narrative Description |
| 16 | Data Model |
| 16:17 | Requirements Validation |
| 17 | Class Model |
| 18 | State Diagram |
| 19:20 | Interaction Diagram |
| 21 | Data Model Design |
| 22:23 | User Interface Design |
| 24 | Client-Object Relation Diagram |
| 24 | Estimated Project Cost |
| 24:26 | User Guide |
| 27:31 | Testing |
| 31:33 | Detailed Design |

1. **Introduction:**

1.1 Purpose:

A customer can view his account details depending on the requirement and the services

provided by the bank.

1.2 Scope:

WallStreet Bank allow Customer (i.e., user) is provided with various operations

like view balance, view transactions, transfer of money etc., from his account.

Administrator can login with username and password which is maintained as highly

confidential. He can add/delete customers as well as the other bank related details.

1.3 Overview:

Our project on ‘Banking System’ is a fully computerized one which provides an account holder to view his account information such as checking balance, recent transactions, transfer of funds, etc. User can also view his account information on his personal computer

you can check table of content to reach to any topic in the document.

**2. General Description**

2.1 Product Perspective

The Wallstreet Website connected with different systems as:

1. Facebook
2. Gmail
3. Governmental website to pay Electricity bills, water……etc.
4. Other bank systems,

2.2 General Capabilities

1. View balances: user can take a look at his balance and past transactions. If user have more than one account, he can also do transfers between accounts.

2. Transfer funds: When user select Transfer Funds, he'll be asked where to transfer the money to and from, when, and the amount.

3. Set up recurring bill payments or transfers: If user make a regular payment every month, it might be convenient to set up an automatic withdrawal from your account.

4. Monitor WallStreet investments: If user have any WallStreet investments, he can keep an eye on those stocks or mutual funds here.

5. Pay bills: To pay your bills online, user just need to add to his account the names of the companies he wishes to pay bills to.

6. View our VISA\* accounts: Always a good place to monitor his spending. he can make his credit card payments online, right from his account.

7. Order Cheques: We don't need them much anymore due to online banking and debit purchases, but if user still use cheques, he can order them directly from the WallStreet website.

8. Take Loan:

User can lend amount of money from the bank .

2.3 General Constraints

System constrain :

Can be limited user depends on server

User constrain :

1.user can’t withdraw more than 10,000

2. user can’t make Do balance process for more than

10,000

3. user can’t pay bill with more 10,000.

4. can’t sign in with an existing email.

2.4 User Characteristics

1. User should register for WallStreet bank with the branch where he/she maintain the account.

2. If user maintain accounts at more than one branch, he/she need to register at each branch separately.

3. Normally WallStreet Bank services will be open to the customer only after he/she acknowledges the receipt of password.

4. We invite customers to visit his/her account on the site frequently for transacting business or viewing account balances. If he believe that any information relating to his/her account has a discrepancy, please bring it to the notice of the branch by e-mail or letter.

5. All accounts at the branch whether or not listed in the registration form, will be available on the Wallstreet Bank. However, the applicant has the option to selectively view the accounts on the WallStreet Bank.

2.5 Environment Description

Front End/Language :

For Body: HTML

For page style : CSS and Java Script

Back End :

PHP

Database :

MYSQL

2.6 Other resources needed

Additional Tools :

XAPM Server, bootstrap to make our site responsive (work on different platform).

Operating System :

Windows 10

**3. system Requirement:**

**Specific Requirements:** How the online banking will interact with the environment, what will be the functional and non-functional requirement. These all the steps should be defined here for providing a powerful base to the design phase. The design of the project will completely depend on the functional and non-functional requirements. So, these should be defined clearly and accurately for the effectiveness •

**Functional:**

1 - Customer will be able to register and have an account on the bank website.

2 - It will be easy for the customer to log in to do his transactions.

3 - This system will provide a path to the customer of the bank to transfer his balance to other account.

4 - Customers will be able to pay the bills from their account.

5 – The customer will be able to do all his banking transactions online as withdraw or deposit.

6 – The customer can ask for loan and get it.

7 – The customer will be able to contact the bank with email & phone number.

8 – The currency Transfer can be done by the website.

9-- Bank offers job vacancies on the website and allows applicant to upload their cv.

**Non-Functional:**

10-This Project must be safe and secure because customers will directly contact their account through the internet.

11- Banks will feel free to adopt it because it will not be so much expensive.

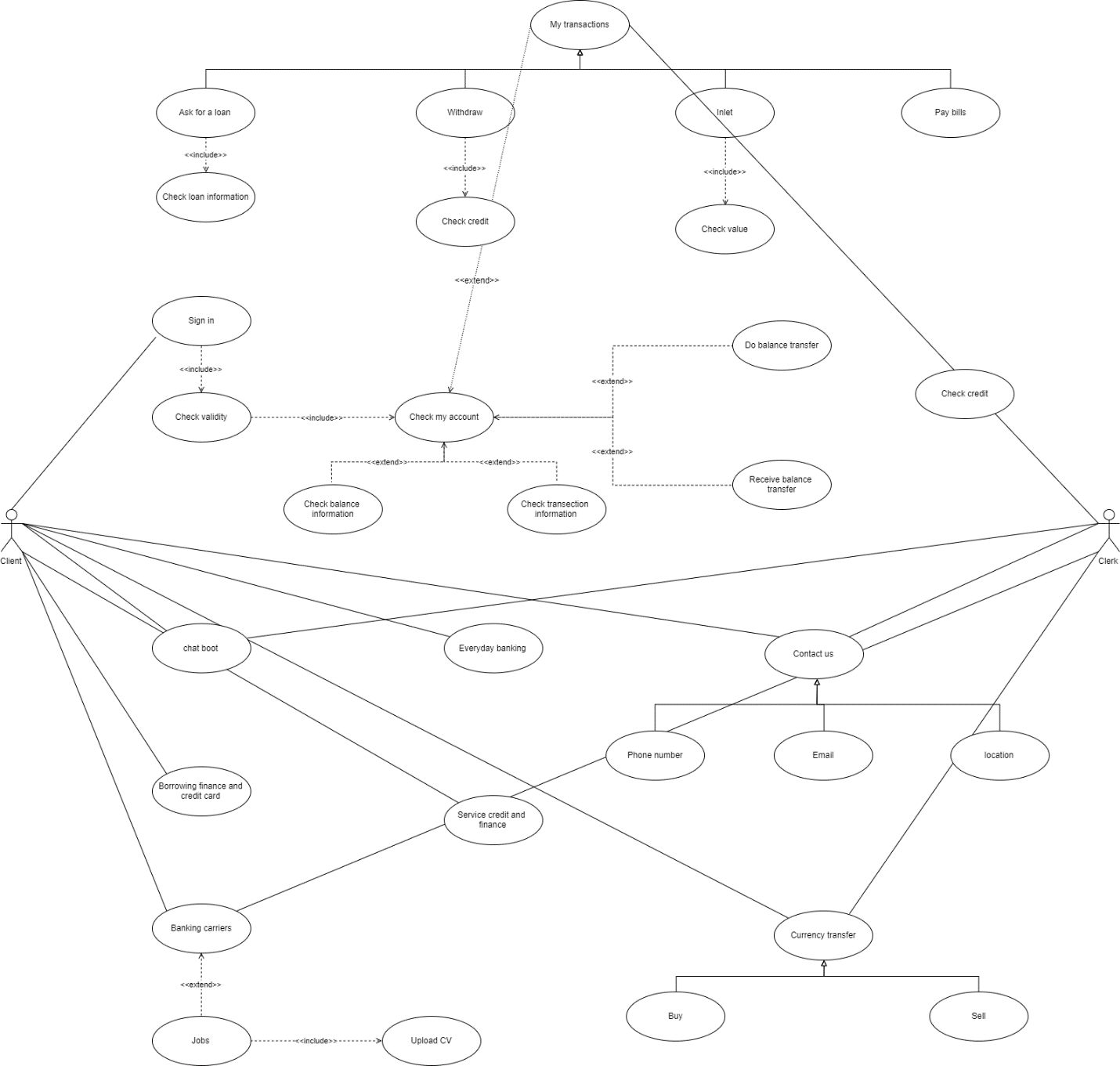
12- It isn’t sure that each visitor of the Bank’s website will be a customer. He/she would be a normal visitor interested in reading the features bank provides. The website’s main page should provide him the basic features and benefits of the bank to these types of users.

13- Application will be accessed through a Browser Interface

14- It should use less memory and will be easily accessible by the user. Memory management should be done wisely so that none of the memory part goes wasted.

15- The website should be Reliable, Testable, Secured and work with Fast speed.

**4. Use-Case Diagram:**



**5. Narrative Description:**

**1. Use case name: Ask for a loan**

**Related Requirements:** Requirement 6

**Goal in context:** The user asks for a loan from the bank

**Preconditions:** The user have to check the loan Information and provides all the needed papers

to bank staff

**Successful End condition:** The user successfully sends the request for the loan

**Failure End condition:** The user doesn`t send the request so doesn`t get the loan

**Primary Actors:** User

**Secondary Actor:** Bank Staff

**Trigger:** The user will ask for a loan and enters its information

**Included cases:** Check loan information

**Main** **Flow: Step Action**

1. The user asks for a loan

include: check the user check the loan information and conditions

Loan information

1. The user enters the information of the loan
2. The information is to be verified by bank staff
3. The loan is accepted or refused according to verification

**2. Use case name: Check Loan information**

**Related Requirements:** Requirement 6

**Goal in context:** The user checks the information and conditions for loans

**Preconditions:** The user searches for loan information or asks for a loan

**Successful End condition:** The user knows the information

**Failure End condition:** The user doesn`t know the information

**Primary Actors:** User

**Secondary Actor:** None

**Trigger:** The user asks to check the loan information on the website

**Main** **Flow: Step Action**

1. The user searches for the loan information to

read it

2 The user checks the loan information

**3. Use case name: Sign In**

**Related Requirements:** Requirement 1 , Requirement 2

**Goal in context:** The user signs in to his bank account

**Preconditions:** The user enters his account information

**Successful End condition:** The user signs in to his bank account successfully

**Failure End condition:** The user can`t sign in to his account

**Primary Actors:** User

**Secondary Actor:** None

**Trigger:** The user wants to sign in to his bank account

**Included cases:**  Check Validity

**Main Flow: Step Action**

1. The user enters his account information
2. The user press signs

Include: Check Validity

3 The user signs in to his account

**4. Use case name: Check Validity**

**Related Requirements:** Requirement 1 , Requirement 2

**Goal in context:** Check that the user’s account information is valid for sign in

**Preconditions:** The user enters his account information to be checked

**Successful End condition:** The information is checked and user signs in

**Failure End condition:** The information can`t be checked

**Primary Actors:** Bank Staff

**Secondary Actor** : None

**Trigger:** The user enters information

**Included cases:** Check my account

**Main Flow: Step Action**

1. The user enters his bank information
2. The information is checked if it is valid

Include:: Check If the information is valid the user can sign in to his

My account account easily

**5. Use case name: Contact us**

**Related Requirements:** Requirement 7

**Goal in context:**  The user will be able to contact with the bank by many methods

**Preconditions:** The user searches for a method to contact the bank or its

location

**Successful End condition:** The user contacts the bank successfully

**Failure End condition:** The user cant contact the bank by either ways

**Primary Actors:** User

**Secondary Actor:**  Bank Staff

**Trigger:** The user tries to reach the bank and contact with it

**Included Cases: Check Location , Use phone number , Use email**

**Main** **Flow: Step Action**

1. The user wants to know the bank location

Include: Check The user know the bank location so he can go to the bank

Location

2 The user wants to send emails to the bank about services

Include: Use The user sends emails to the bank to get what he wants

Email

1. The user can call the bank directly

Include: Use The user can speak directly with one of the bank staff

Phone number

**6. Use case name: Check Location**

**Related Requirements:** Requirement 7

**Goal in context:**  The user knows the location of the bank

**Preconditions:**  The user searches for the location to contact with the bank

**Successful End condition:** The user knows the location of the bank and can get to it easily

**Failure End condition:**  The user don’t know the location and can’t get there

**Primary Actors:**  User

**Secondary Actor:**  None

**Trigger:** The user searches for the location of the bank to go there

**Main** **Flow: Step Action**

1. The user searches for the location of the bank
2. The user finds the location of the bank
3. The user can go to the bank easily

**7. Use case name: Use Phone number**

**Related Requirements:** Requirement 7

**Goal in context:** The user can call the bank directly and speak with the bank staff

**Preconditions:** The user tries to contact the bank so he searched for the phone

number

**Successful End condition:** The user can call the bank easily and contact them

**Failure End condition:** The user can’t contact the bank through phone number

**Primary Actors:** User

**Secondary Actor :** Bank staff

**Main Flow** **: Step Action**

1. The user tries to call the bank so he searches for the number
2. The user finds the number
3. The user contacts the bank through phone

**8. Use case name: Use Email**

**Related Requirements:** Requirement 7

**Goal in context:** The user can easily send and receives emails from the bank

**Preconditions:**  The user wants to contact the bank using email

**Successful End condition:** The user contacts the bank by email easily

**Failure End condition:** The user can’t use email to contact the bank

**Primary Actors:** User

**Secondary Actor:** Bank Staff

**Main Flow: Step Action**

1 The user tries to email the bank so he searches for the email

2 The user finds the email address

3 The user contacts the bank through email

**9. Use case name: Use Chat Bot**

**Related Requirements:** Requirement 7

**Goal in context:** The user uses chat bot to chat with bank staff for services

**Preconditions:** The user wants to contact with one of the bank staff by chatting

**Successful End condition:** The user gets the service he wants successfully

**Failure End condition:** The user can`t contact the bank

**Primary Actors:** User

**Secondary Actor:** Bank Staff

**Trigger:** The user wants to chat with one of the bank staff so he uses

chatbot

**Main** **Flow: Step Action**

1. The user wants to contact with the bank through chatting
2. The user uses chat bot to ask for the services he wants
3. The bank staff chat with the user to help him with what he wants

**10. Use case name: Do transaction**

**Relation Requirements:**  A.1

**Goal in Context:** customer do transaction

**Preconditions:** customer login to his account

**Successful End Condition:** Customer do a certain transaction

**Failed End Condition**: Customer fail to do his transaction

**Primary Actor** : Actor

**Secondary Actor:** Server

**Trigger:** Customer enter his account

**Main Flow**

1) Customer login into account

2) Determine which transaction he want

3) Server make this transaction successfully

**Extensions**

3.1) server failed to do transaction

**11.Usecase name: Deposit**

**Relation Requirements:**  A.1

**Goal In Context:** Customer add certain value to his account

**Preconditions**: Customer enter “My transaction page”

**Successful End Condition**: Money added to Customer account

**Failed End Condition:** Customer failed to add money to his account

**Primary Actor:** customer

**Secondary Actor :** server

**Trigger :** Customer enter “My transaction page”

**Include :** Check value

**Main Flow**

1) Customer enter “My transaction page”

2) Customer enter his data

3) server add the value that user enter to his account

**Include** : check valuecustomer check that depositing value is added to his credit

**Extensions**

2.1) Customer enter wrong data

**12. Use case name: pay Bills**

**Relation Requirements** : A.1

**Goal In Context** : customer pay his bills online

**Preconditions** : customer enter “My transaction” page and select “ pay Bills”

**Successful End Condition**: customer pay his bills

**Failed End Condition** : cutomer fails to pay his bills

**Primary Actor** : customer

**Secondary Actor** : server

**Trigger** : customer enter “My transaction” page

**Main Flow**

1) customer enter “My transaction page” and select “Pay Bills”

2) Customer write his data and value of Bill he wants to pay

**Extensions**

2.1) customer write wrong data

**13. Use case name: Withdraw**

**Relation Requirements** : A.1

**Goal In Context** : Customer withdraw certain value of money

**Preconditions** : Customer enter in “my transaction” page

**Successful End Condition** : Customer get money

**Failed End Condition** : Customer doesn’t get his money

**Primary Actor** : Customer

**Secondary Actor** : server

**Trigger** : customer enter value of money he want to withdraw from his account

**Include case** : Check credit

**Main Flow**

1) Customer enter “Do transaction” page

2) Customer enter his data

3) Customer enter value of money he want to withdraw from his account

4) Server add this value of money to customer credit card

**Include : Check credit** customer check that withdrawen value is taken from his credit

**Extensions**

2.1) Customer enter invalid data

**14. Use case name: Do Currency Transfer**

**Relation Requirements** : Recruitment 9

**Goal In Context** : user can change (pay or sell) money from different currency

**Preconditions** : customer should have enough balance in his account

**Successful End Condition**: change money successfully

**Failed End Condition** : can’t change money

**Primary Actor** : customer

**Secondary Actor** : banking system

**Trigger** : customer click on change currency button .

**Main Flow:**

1. customer click on change money button
2. customer enter amount of money that he wants to transfer.
3. System check balance in the account.
4. The transfer process done successfully.

**Include : Buy ,Sell:** customer can buy or sell in different currency **.**

**Extensions**

4.1) server failed to do currency transfer

**15. Use case name: Search Bank Carriers**

**Relation Requirements** : Requirement 8

**Goal In Context** : anyone can search for available carriers in Wall-Streeter bank website

**Preconditions** : visit Wallstreet bank website

**Successful End Condition**: find suitable job and upload the CV.

**Failed End Condition** : Can’t upload CV

**Primary Actor** : Website visitor

**Secondary Actor** : Bank server

**Trigger** : Click on Search bank carriers button

**Main Flow**

1. User visit WallStreet website
2. User search for suitable job.
3. User upload CV.
4. User applies to the job.

**Include :** jobs

**Extensions**

4.1) Can’t upload CV due to oversize of the CV.

**16. Use case name Check my Account**

**Relation Requirements** : Requirement 1,2,3 .

**Goal In Context:** Customer can check his balance, latest transactions, make balance and

receive balance from different account.

**Preconditions** : the customer should log in first .

**Successful End Condition**: the customer

**Failed End Condition** : customer enter wrong name or password

**Primary Actor** : customer

**Secondary Actor** : bank server

**Trigger** : the customer click on log in button

**Main Flow**

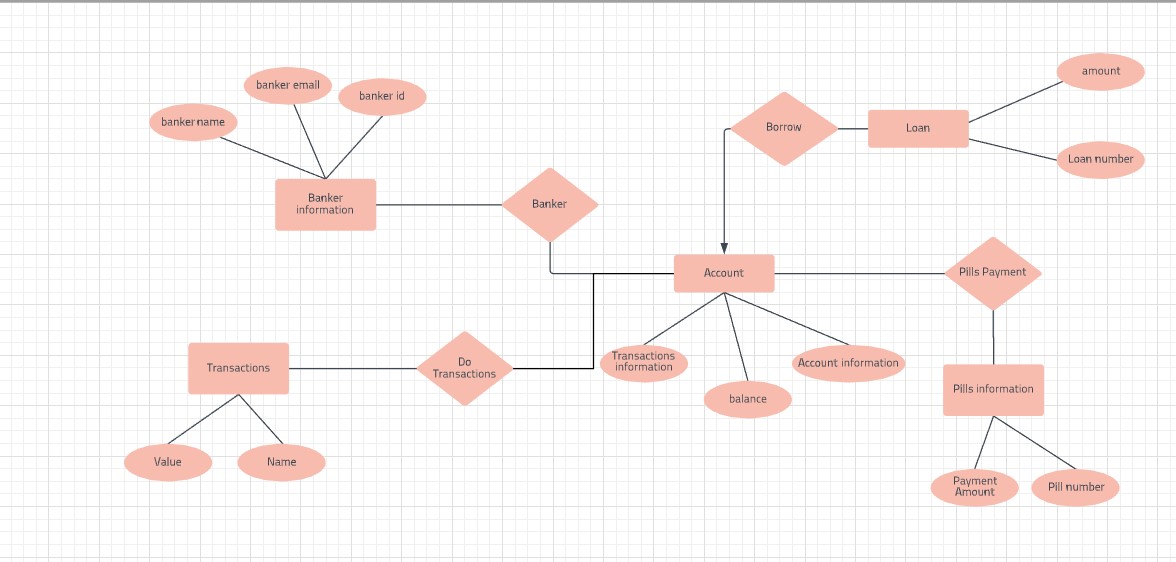
1. Customer visit WallStreet website.
2. Customer log in
3. Customer go to my transaction page.
4. In balance section customer can receive or do balance

Include: Do balance, receive balance.

1. In transaction section user can check latest transactions.

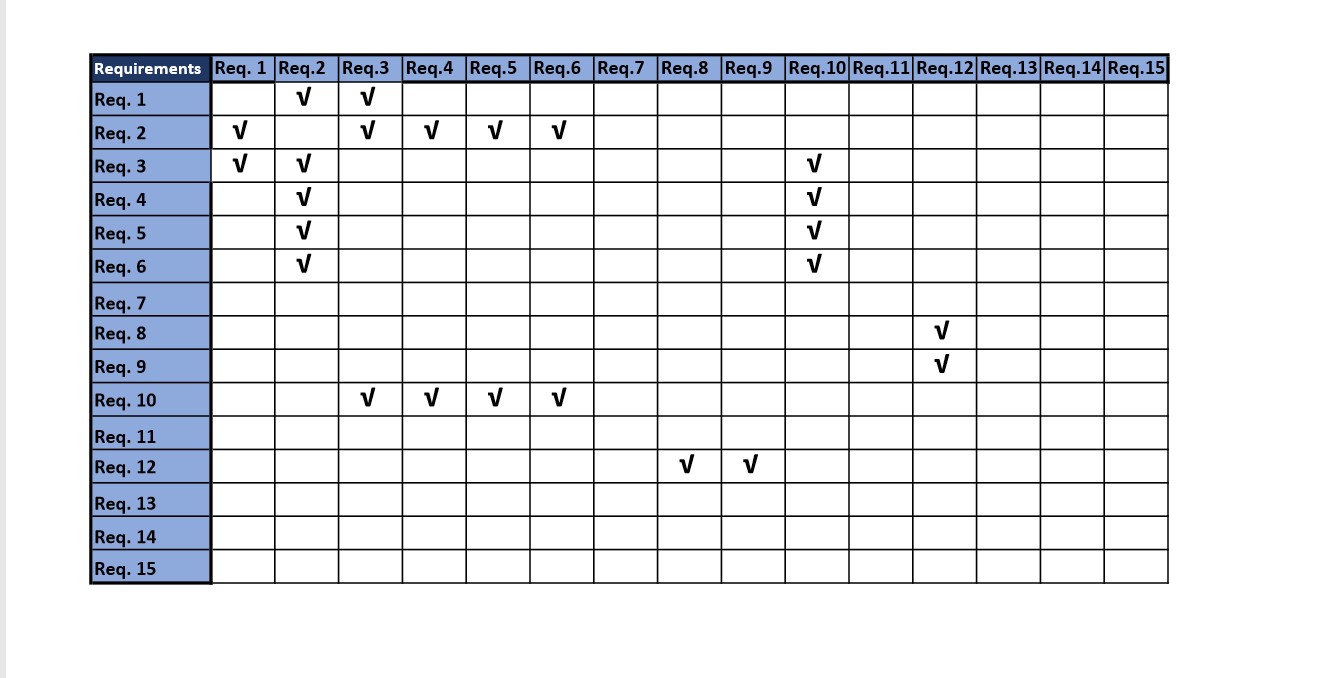
Include: check transaction.

**6. Data Model:**

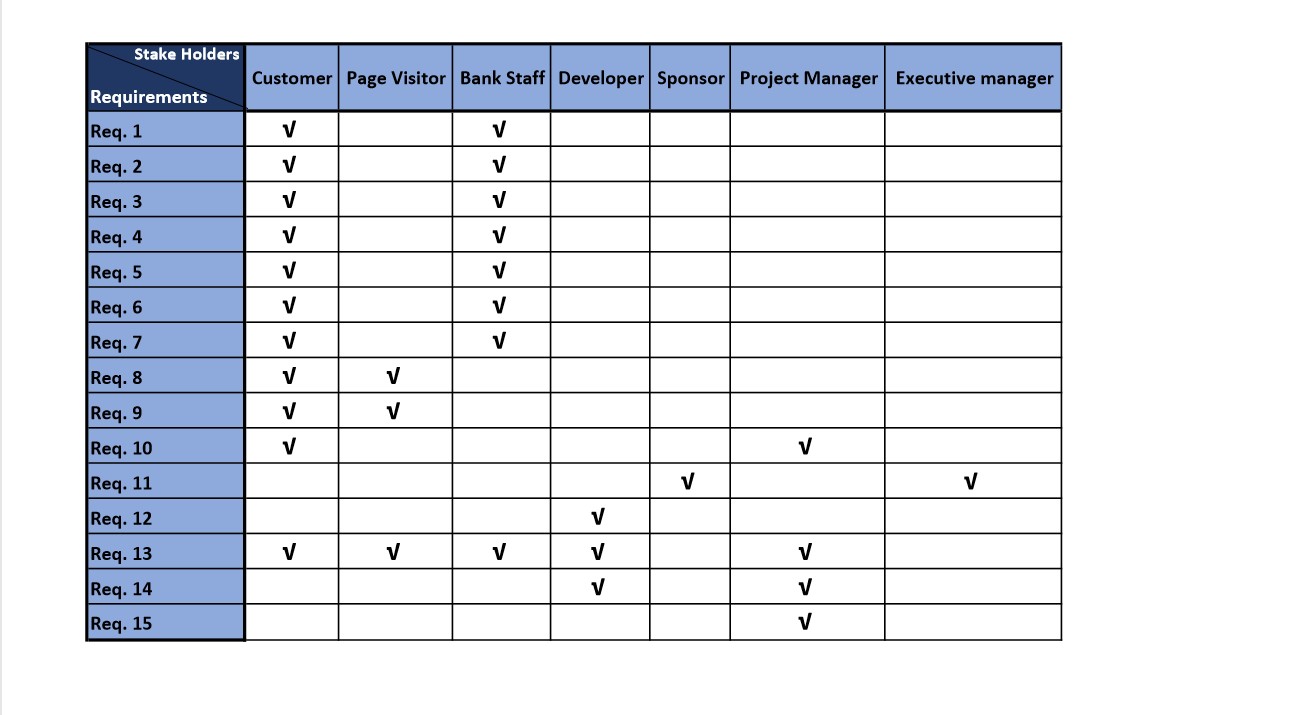


**7. Requirements Validation:**

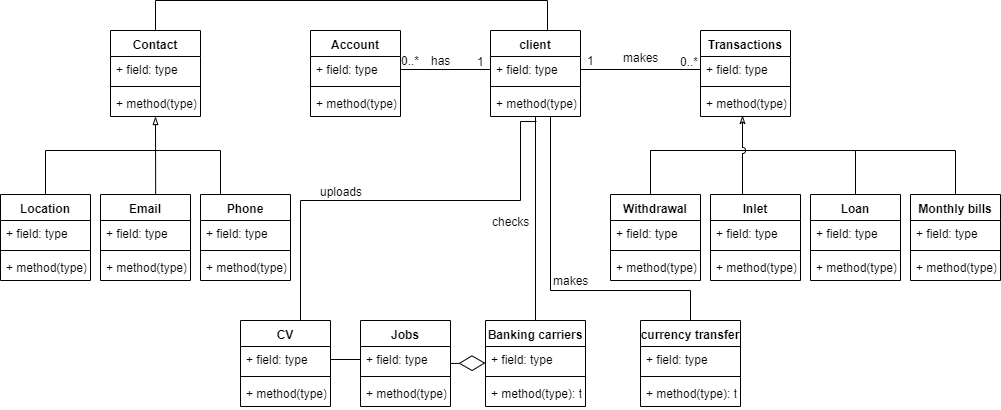
**Requirement traceability**



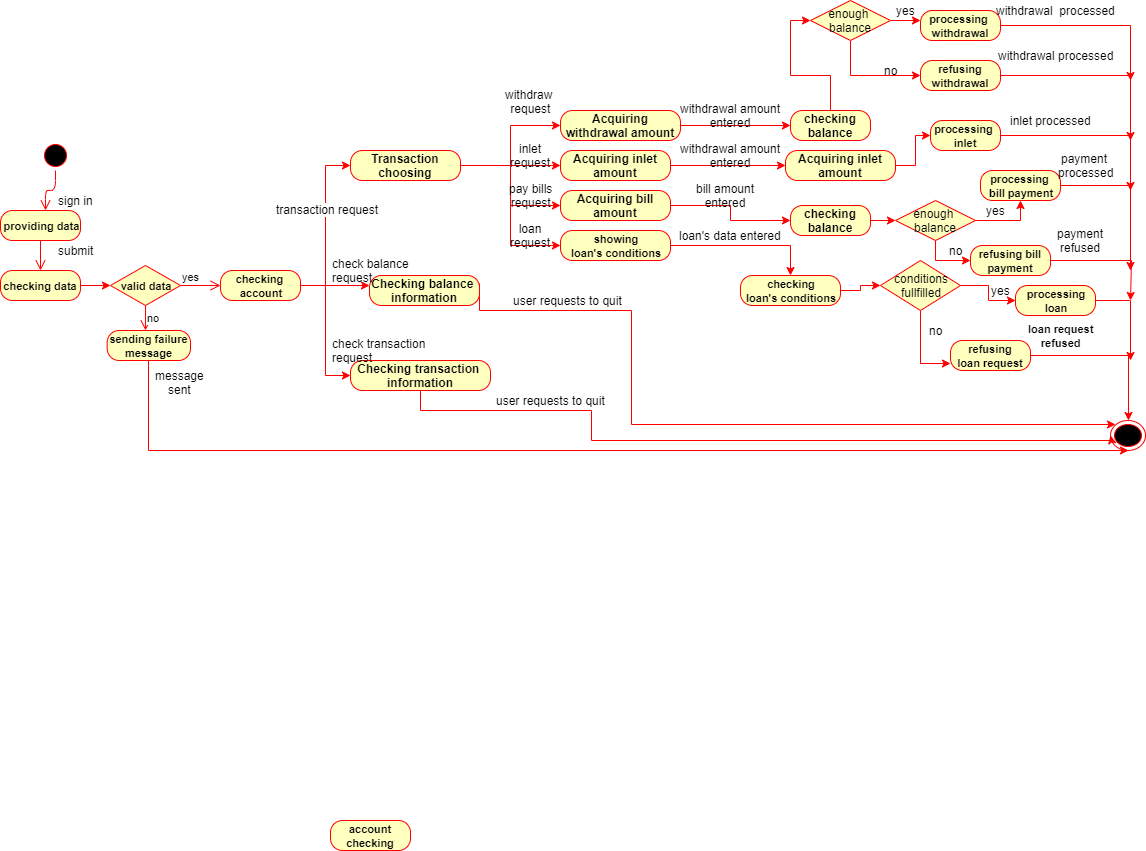
**Source traceability:**



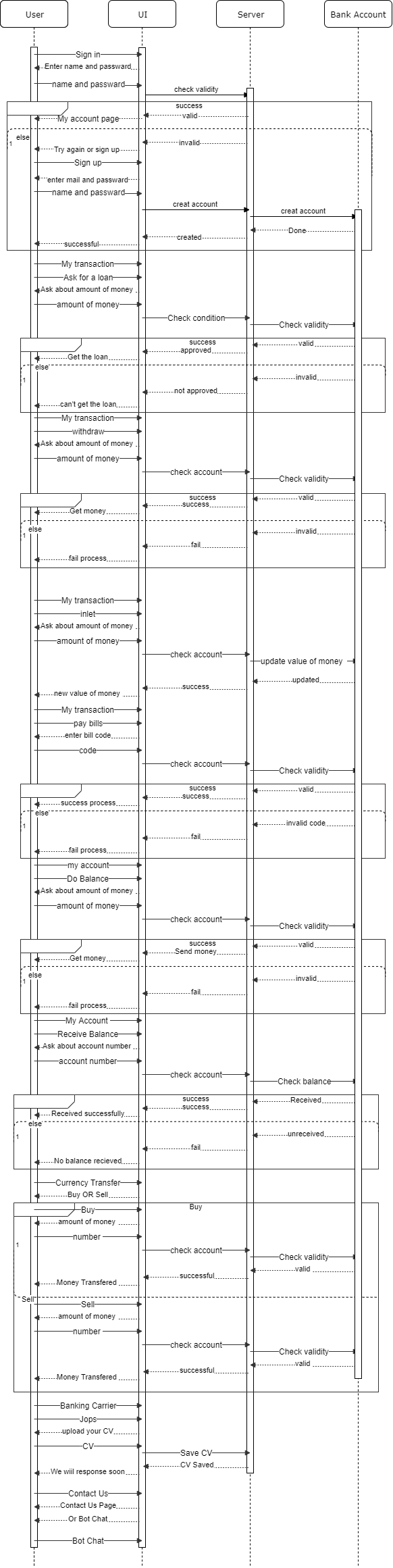
**8. Class Model:**

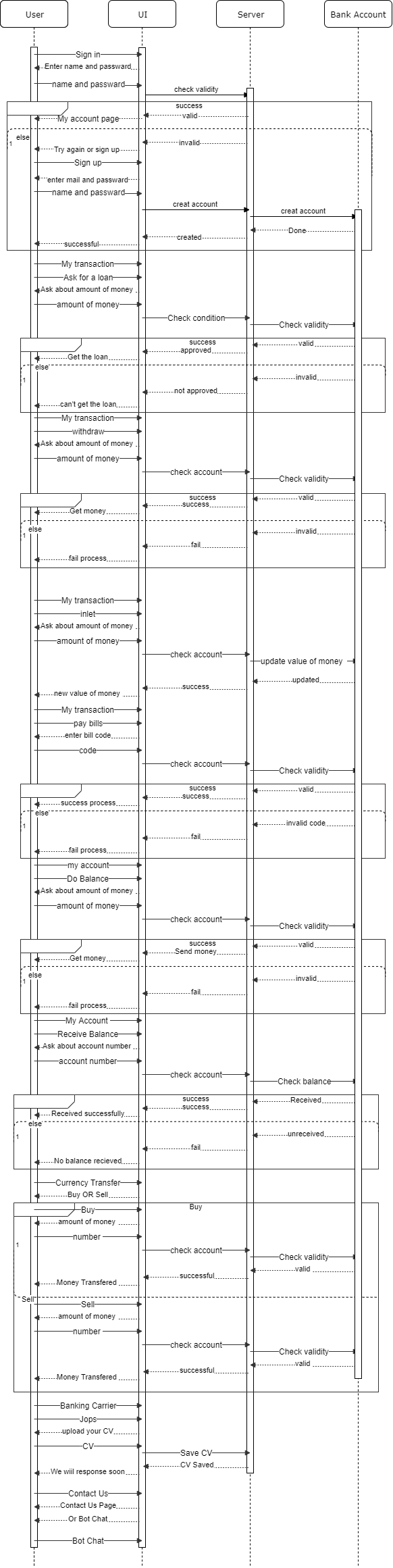


**9. State Diagram:**

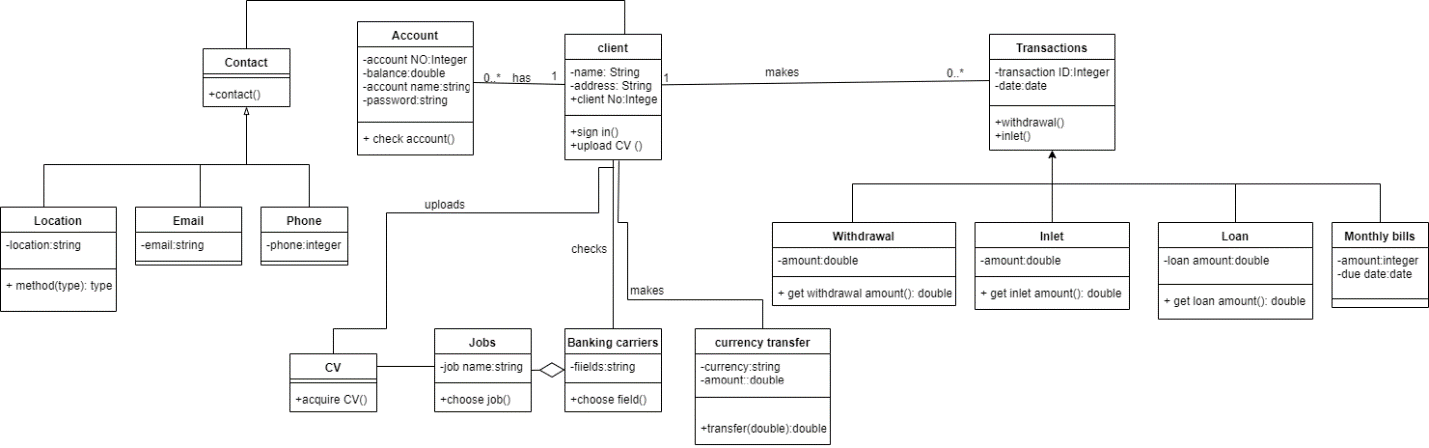


**10. Interaction Diagram:**

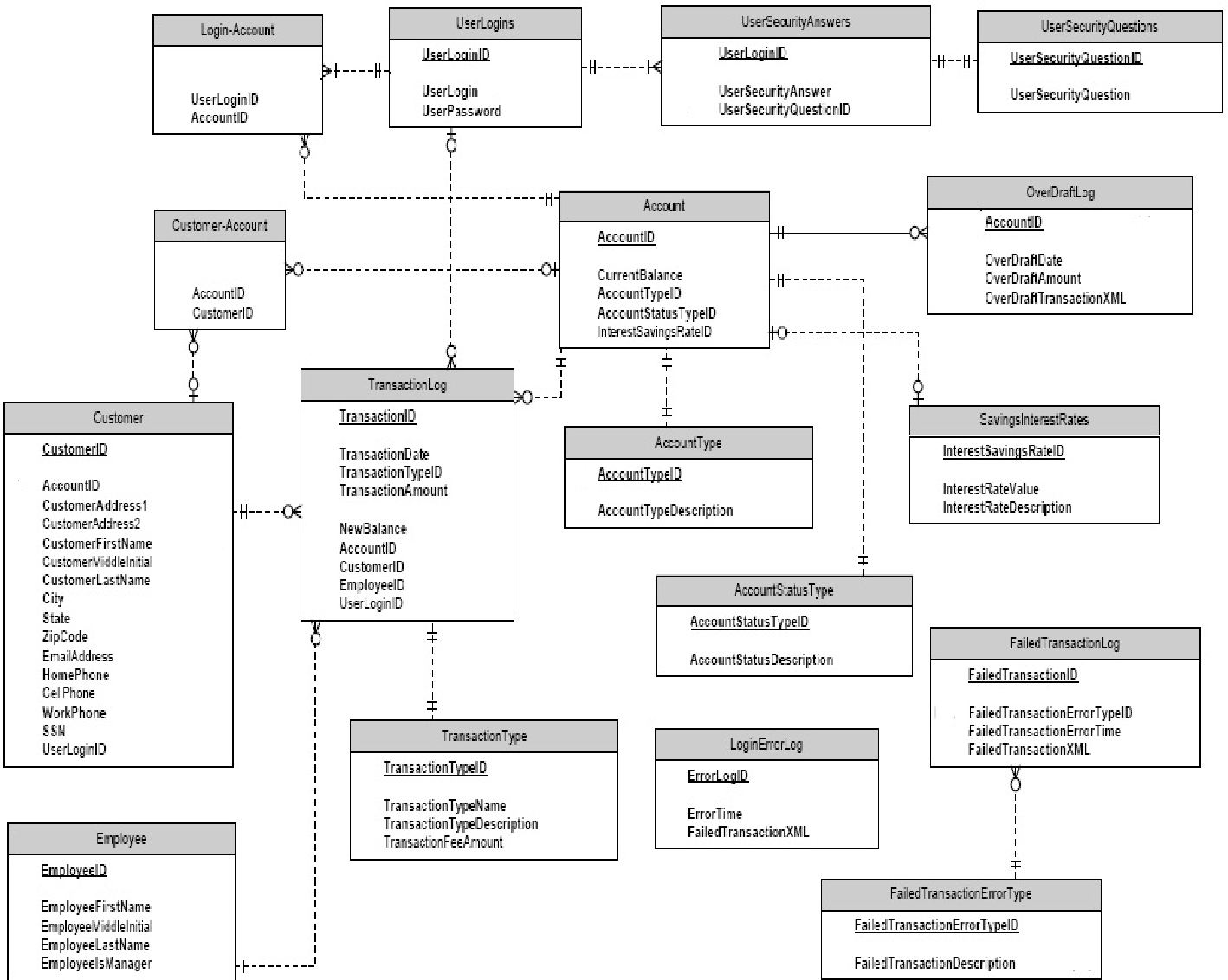
****

****

**11. Detailed Class Diagram**



**12. Data Model Design:**

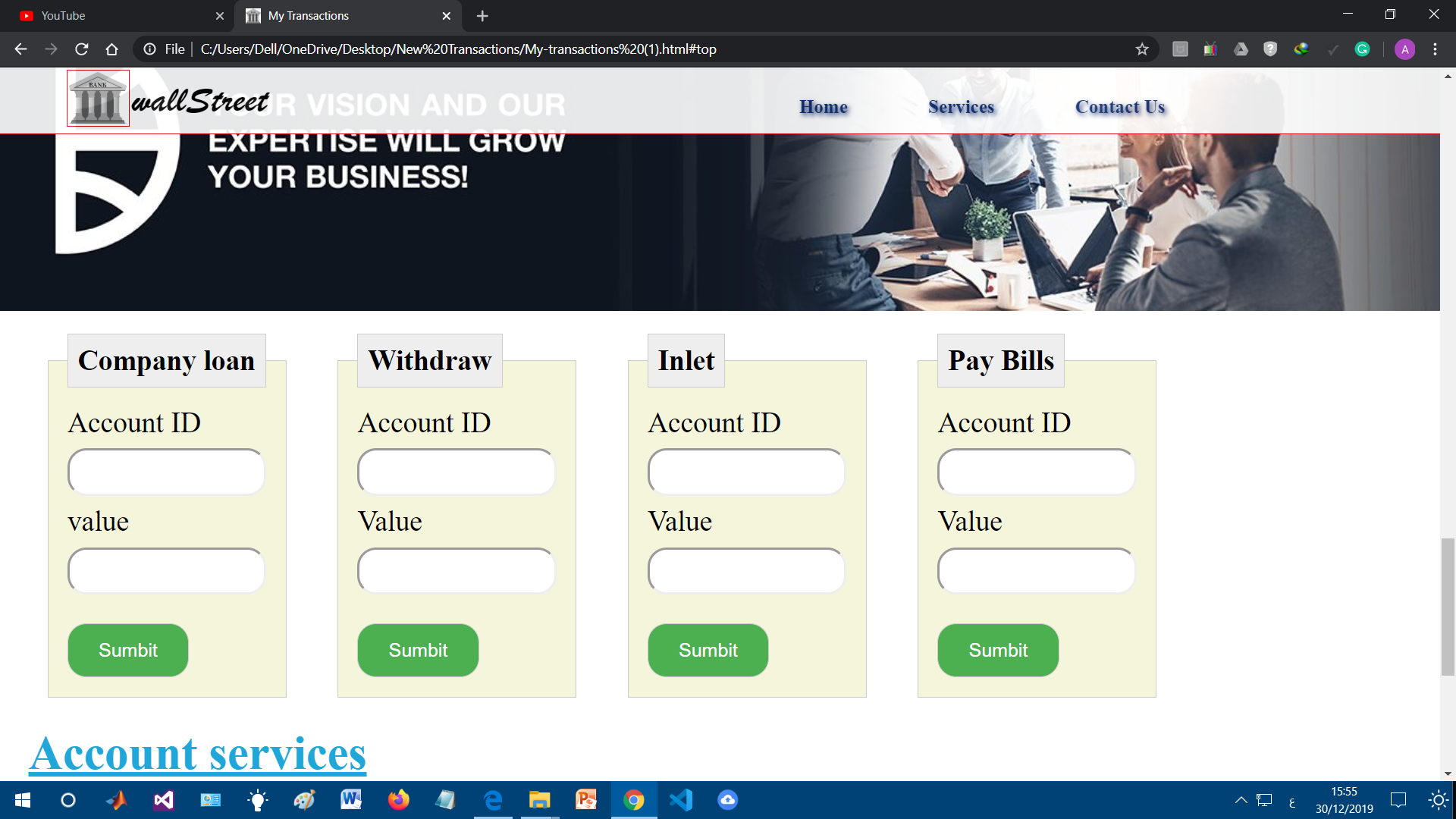


**13. User Interface Design**

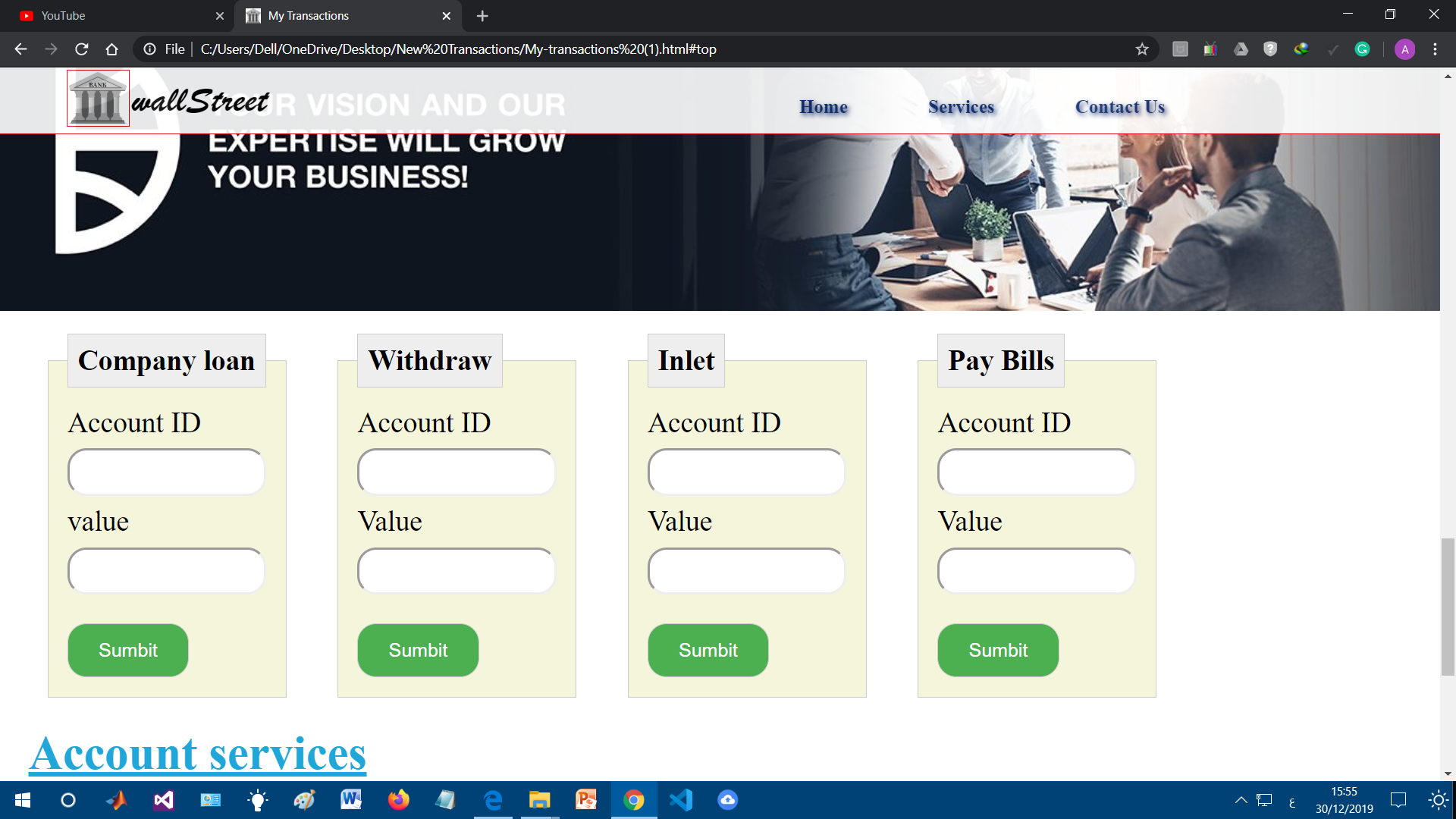
**UI Style:**

Forms-Based Systems

We used forms based interface for simple data entry that user can enter data easily .

****

* The system state changes here from form-based to information presentation when he clicks “submit” to show him information about his transaction .



* When the user enters the password and user name successfully and clicks log in this will take him to his account directly

A screenshot of a cell phone

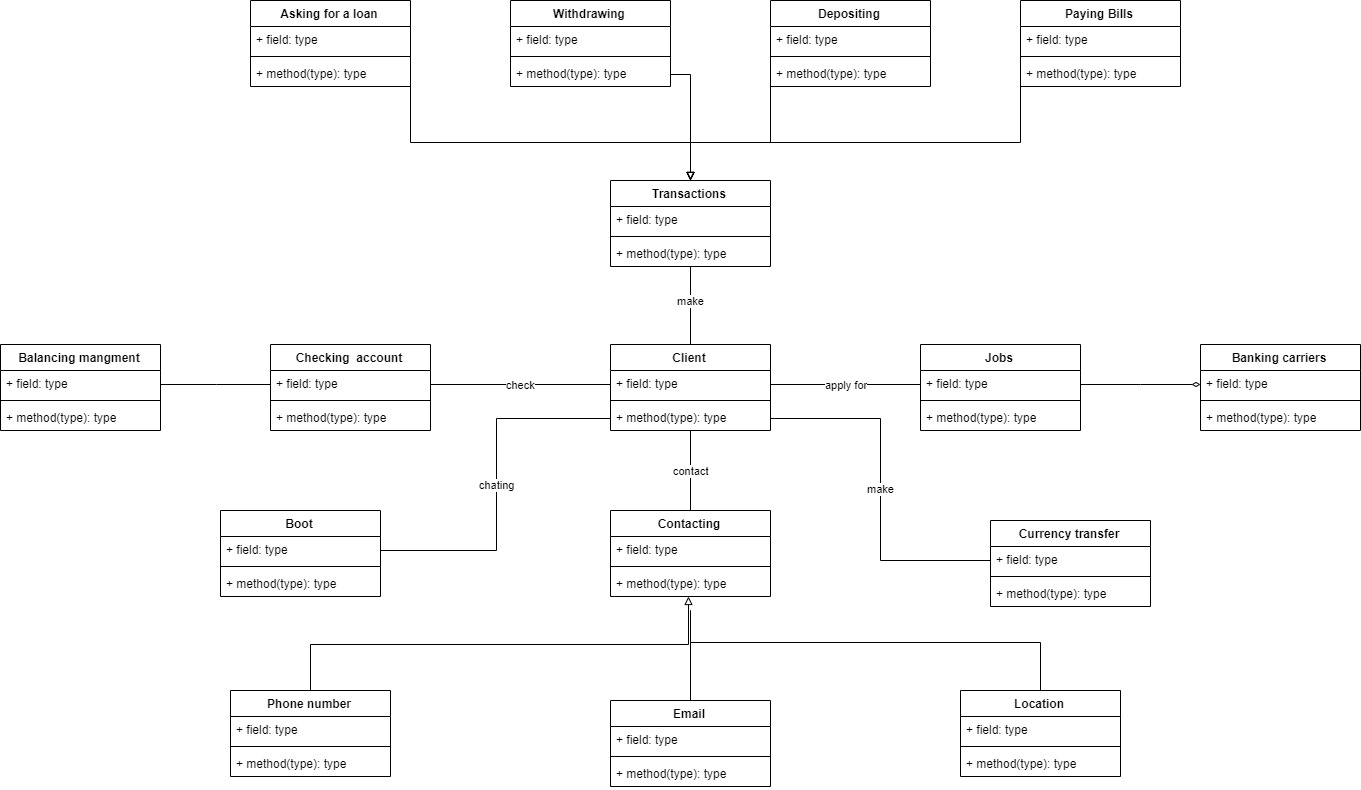
Description automatically generated

* Choosing from the main navigation menu will lead to go from the main home page to sub-pages

A screenshot of a cell phone

Description automatically generated

**14. Client-Object Relation Diagram:**



**15. Estimated Project Cost:**

**COCO Model is related to constructive cost Model**

It’s use to calculate a cost of project. And It’s a semi detached

A project cost it’s depend on multiple criteria

1. Number of lines used in code.
2. Experience develop.
3. Execution time.
4. Main storage constrains.
5. Required execution.
6. Project divided into small section each section do something.

**Our project have 2 main section:**

1. Front end section
2. Back end section
3. **Front end section:**

We use Html & CSS & java script

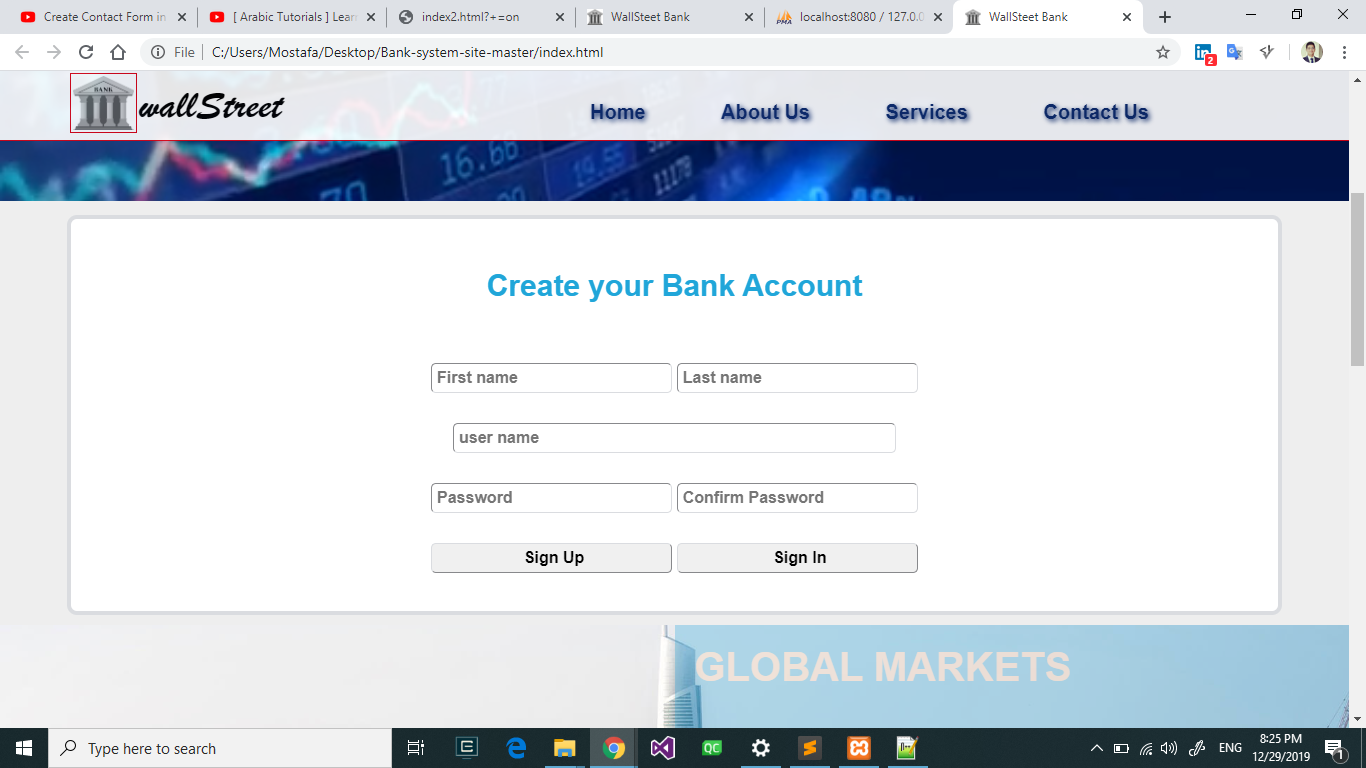
1. **Back end section:**

using php and MYSQL to connect pages and run on local server.

Excepted cost is 400 EGP.

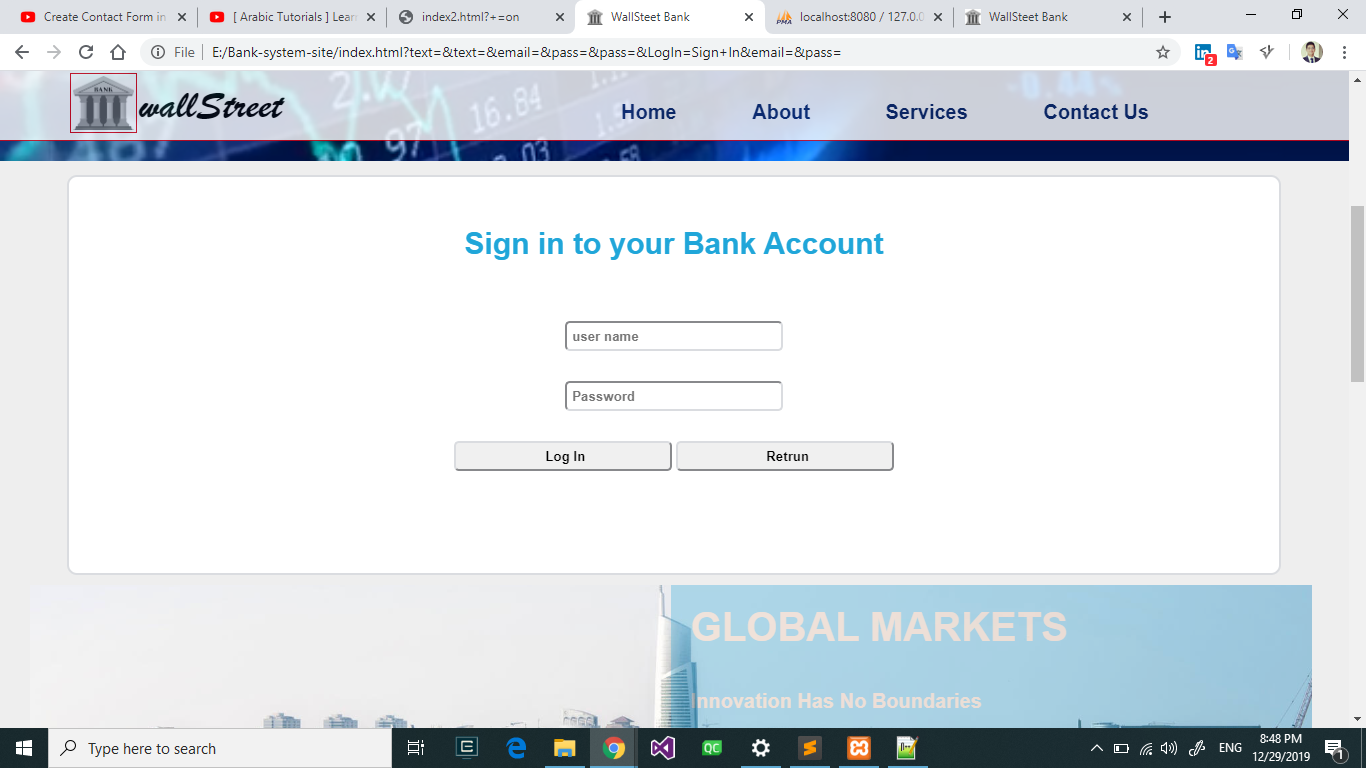
**16. User Guide**

To create new account fill the the field below and click sign up

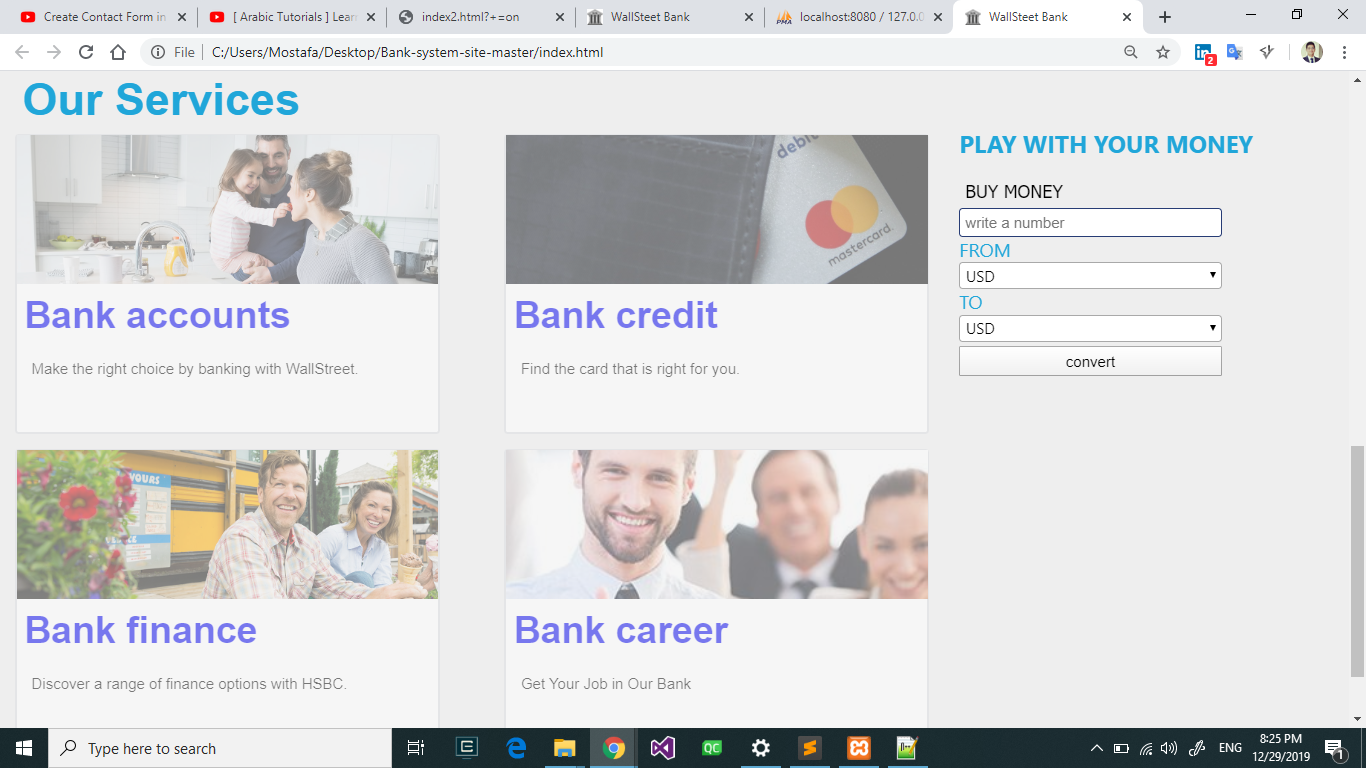


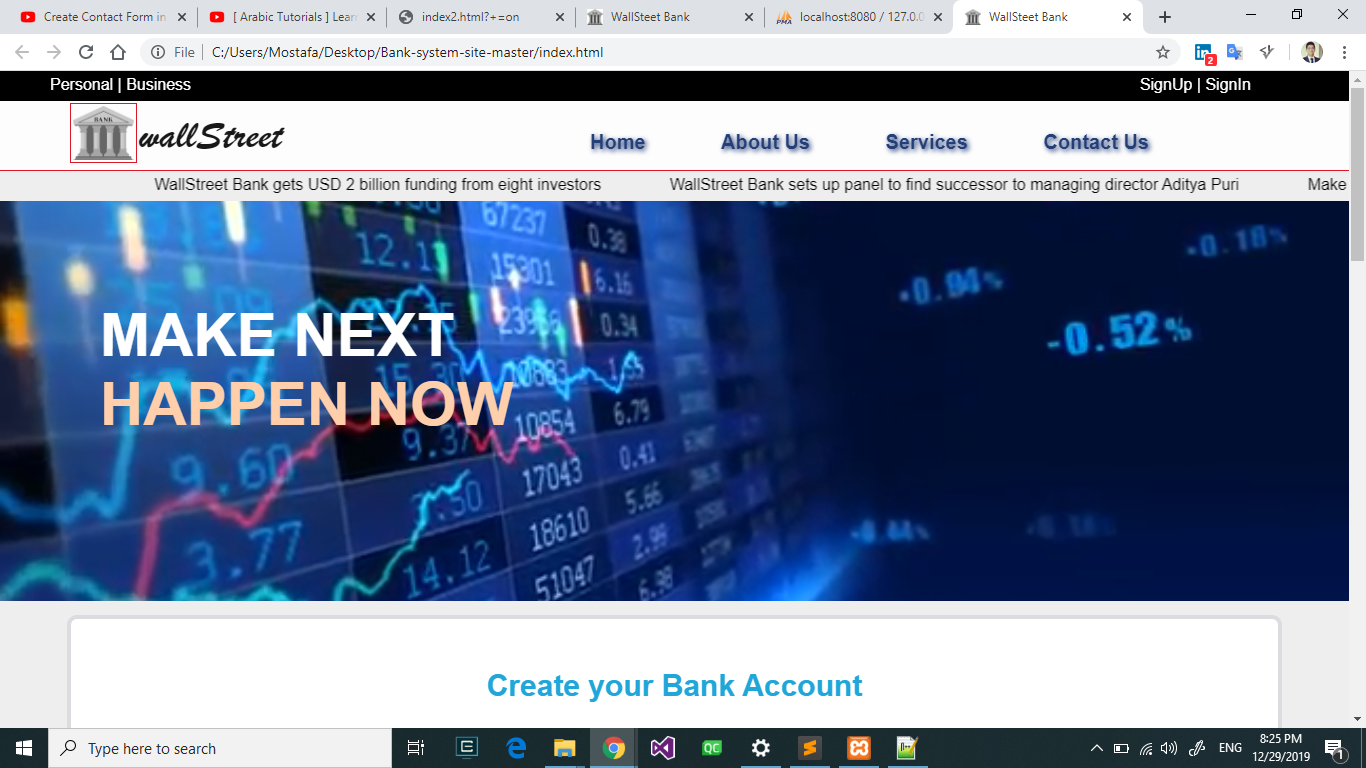
**17. Testing**

To sign in in existing account click sign in button the fill the user name and password and click sign in .

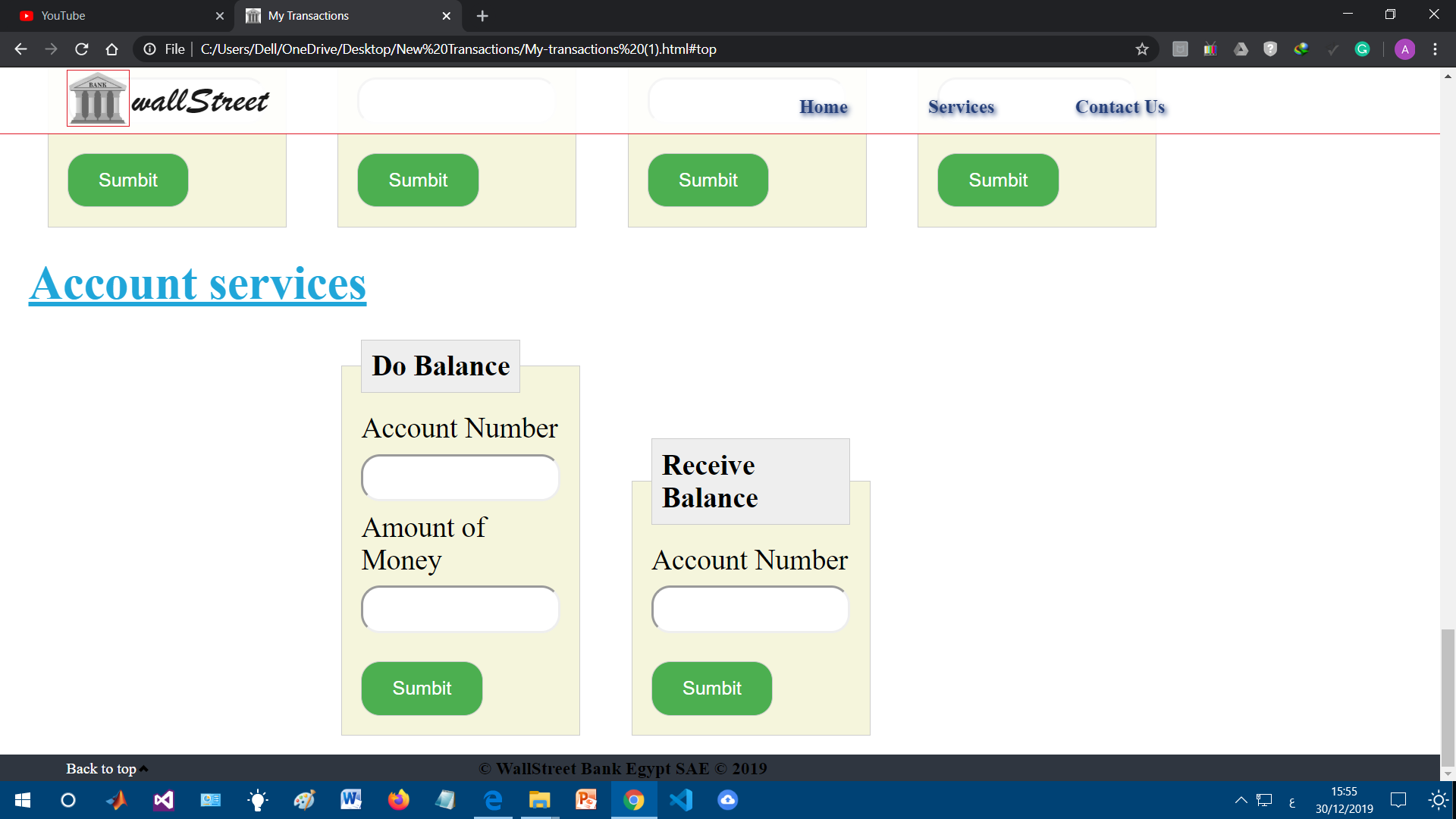
****

Make currency transfer using the following section in the home page

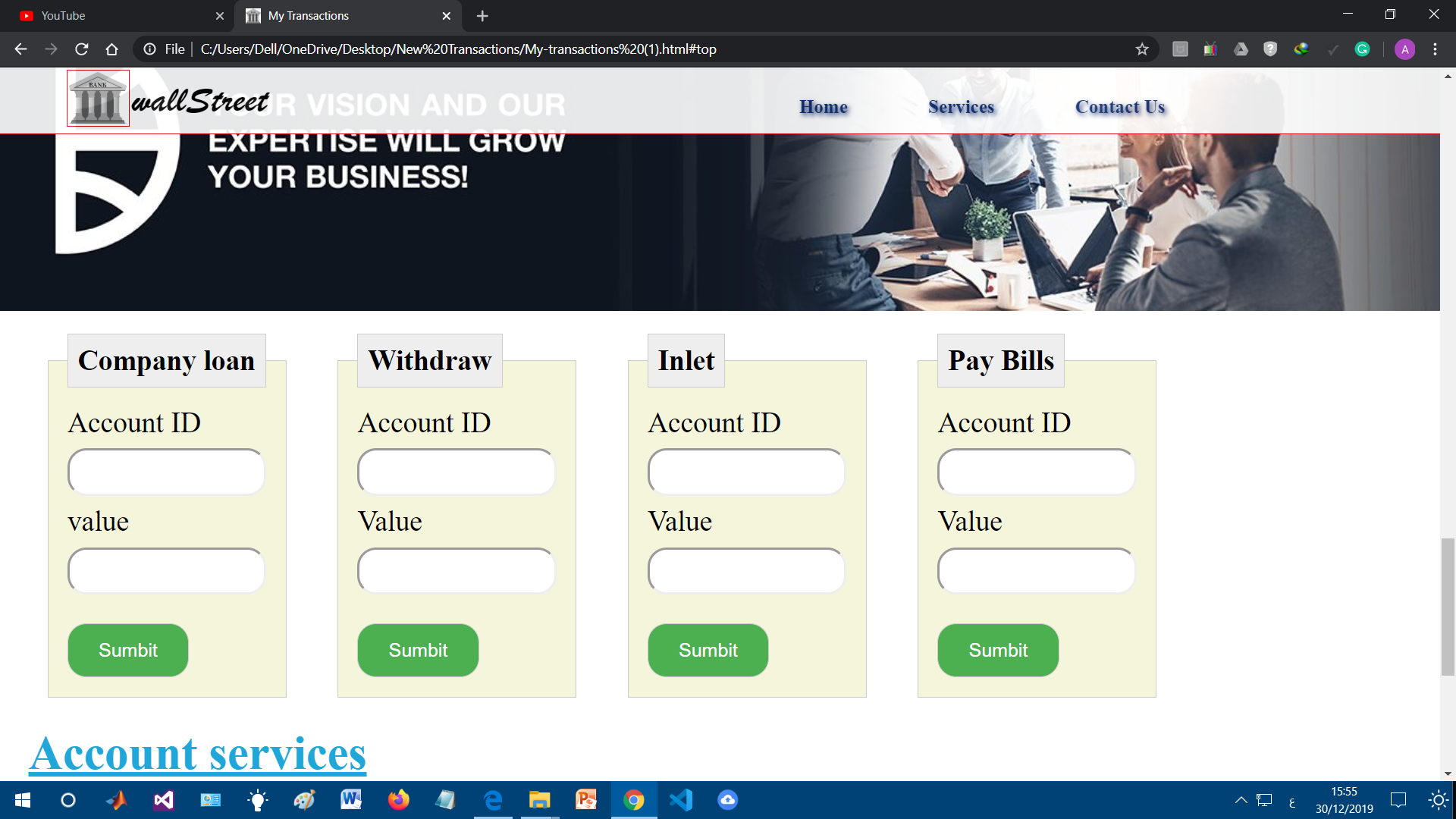


You can read latest news from here

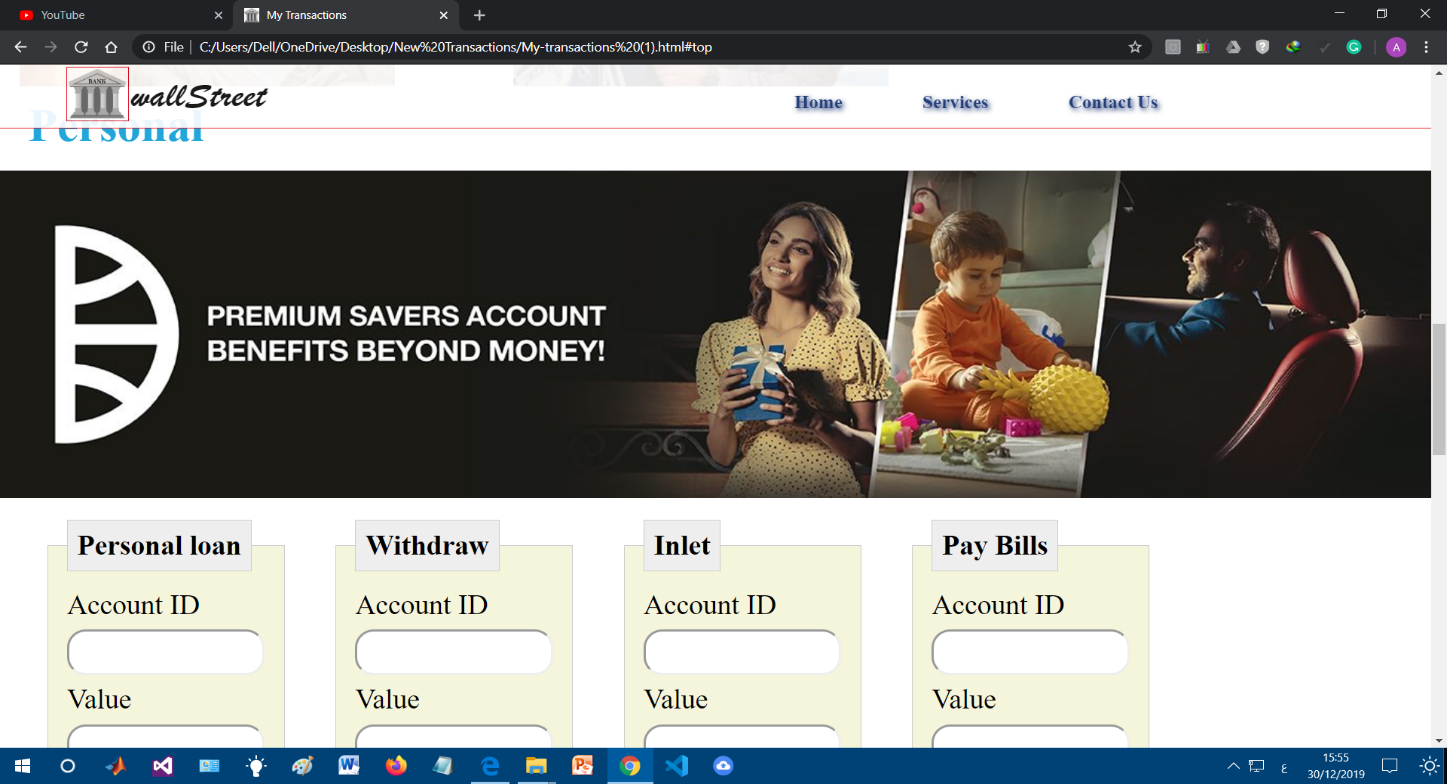
After signing in you can go to my account page and in account service section you can send money to different account using Do Balance or receive money using Receive Balance

****

After signing in go to my account page and using the following section you can Get Loan, Withdraw , Inlet and Pay Bills

****

If you still have a problem you can contact us using mail , phone or chat bot .

****

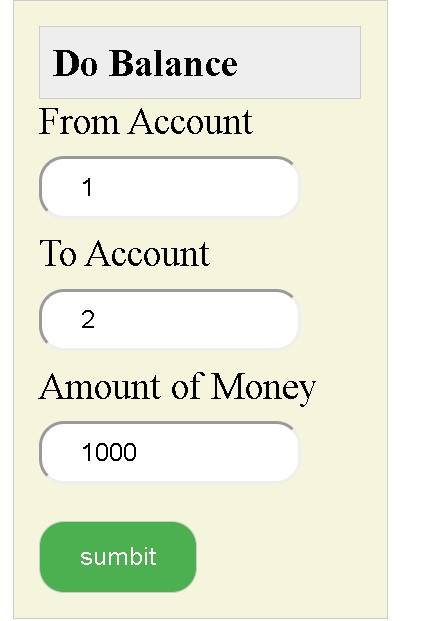
**17. Testing :**

Test case 1 :

**Do Balance test case :**

The user enter the ids of the two different account .

If the balance sufficient the transfer done successfully

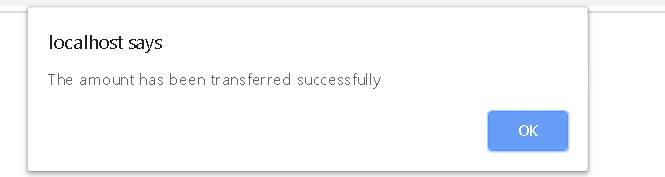


Database before the process

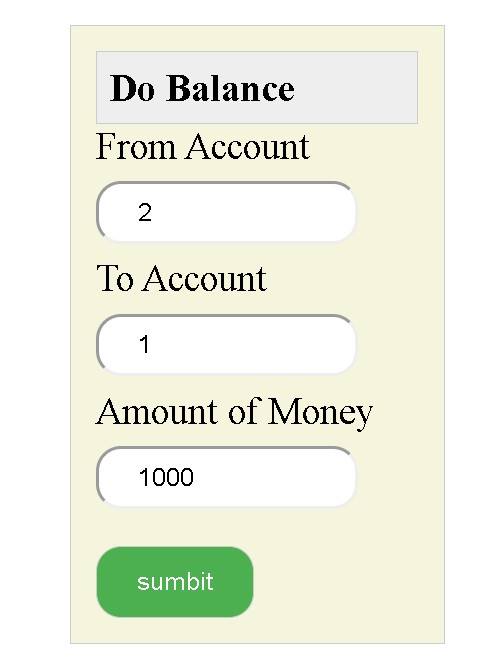


Database after the process :



****

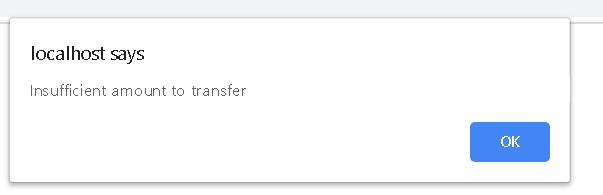
If the balance insufficient



Database before the process:



The result



Test Case two :

**Get Loan test case :**

The user enter the account id and the value of the loan that he want to get



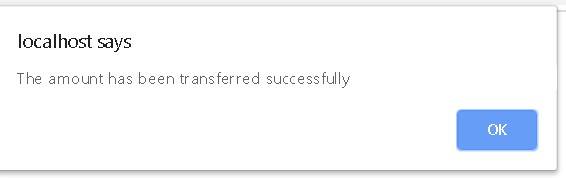
Database before the process :



Database after the process :



Result

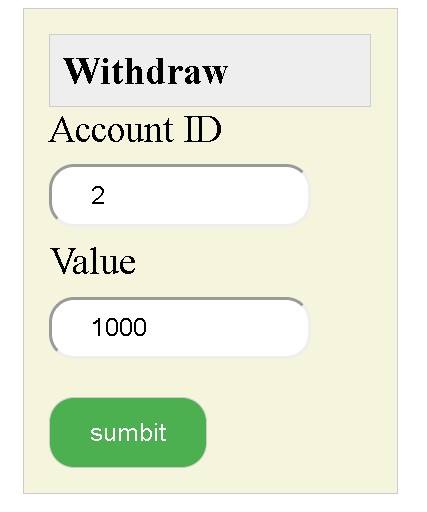


Test case 3:

**Withdraw test case :**

The user enter the account id and amount of money.

If the balance sufficient the transfer done successfully



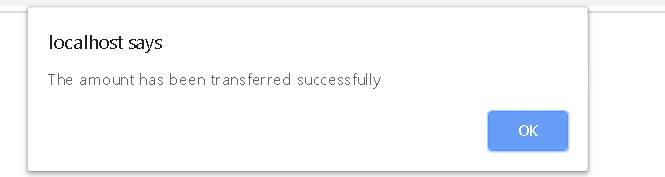
Database before the process



Database after the process :



Result

****

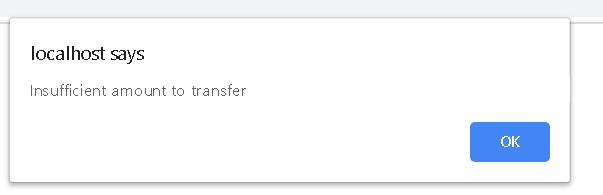
If the balance insufficient



Database before the process:



The result



**18. Detailed Design**

**1.Create New Account**:

Front end section:

Four boxes filled by new user information (first name , last name , password , confirm password and user name )

Button to submit .

Back end section:

If ( create new account ) {

If (email is valid )

Create the account;

Else

Echo error;}

**2. Login**:

Front end section:

Two boxes to get user name and password .

A button to submit .

Back end section:

If (Log in)

{

(email,password)=enter\_email\_password();

Valid=check(email,password);

If(valid)

Enter my account page;

Else

Echo not valid ;

}

**3. Do Balance Transfer :**

Front end section:

Three boxes to get from account id , to account id and amount of money

One button to submit

Back end section:

Post from account id ;

Post to account id ;

Post amount of money ;

If ( available amount < amount )

Echo “ insufficient amount “;

Else

{

From balance = balance – amount;

To balance = balance – amount;

Echo “ amount is transferred successfully “ ;

}

**4. Get Loan:**

Front end section:

Two boxes for account id and amount of money .

One button to submit .

Back end section:

if (Ask for a loan)

{

If (user information is correct)

{ Balance = Balance - value ;

Echo “ Amount has been transferred Successfully “; }

Else

{ echo “ insufficient amount to transfer” ; }

}

**6. Withdraw :**

Front end section:

Two boxes for account id and amount of money .

One button to submit .

Back end section:

If (Withdraw)

{

If (withdrawn value <= balance)

{ Balance = Balance – withdrawn value ;

Echo “ Amount has been transferred Successfully “; }

Else

{ echo “ insufficient amount to transfer” ; }

}

**7. Inlet :**

Front end section:

Two boxes for account id and amount of money .

One button to submit .

Back end section:

If (Inlet )

{

If (user information is correct)

{ Balance = Balance + Depositing value ;

Echo “ Amount has been transferred Successfully “; }

Else

{ echo “ insufficient amount to transfer” ; }

}

**8. Pay Bills :**

Front end section:

Two boxes for account id and amount of money .

One button to submit .

Back end section:

If (Pay Bills)

{

If (user information is correct)

{ Balance = Balance - value ;

Echo “ Amount has been transferred Successfully “; }

Else

{ echo “ insufficient amount to transfer” ; }

}