

---

# Project Application

4111 Introduction to Database

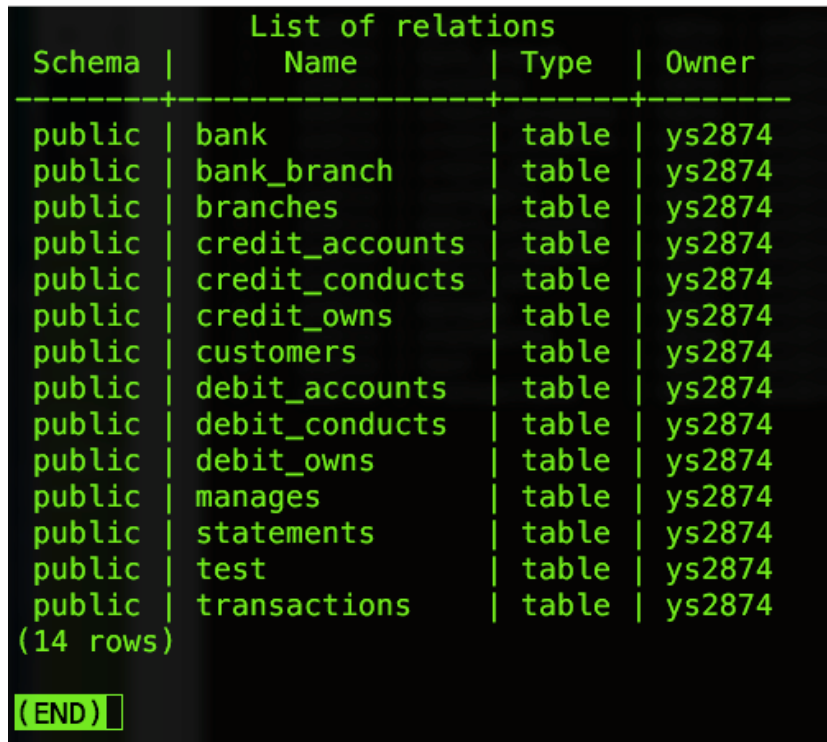
Yu Wang , Yuchen Shi

Mar 28

---

In this project, we built an online banking system which is a simplified version of banking system. The application will enable a customer to login , handle operations with regard to their bank account and modify the account information. The application also has an admin console which allows an administrator to look up for specific information such as branches, customers or transactions.

We have 14 tables in database in the final database according to the schema design.



Schema	Name	Type	Owner
public	bank	table	ys2874
public	bank_branch	table	ys2874
public	branches	table	ys2874
public	credit_accounts	table	ys2874
public	credit_conducts	table	ys2874
public	credit_owns	table	ys2874
public	customers	table	ys2874
public	debit_accounts	table	ys2874
public	debit_conducts	table	ys2874
public	debit_owns	table	ys2874
public	manages	table	ys2874
public	statements	table	ys2874
public	test	table	ys2874
public	transactions	table	ys2874

(14 rows)

(END)

For the application, we have 3 modules in the application and each of them will be elaborated in next section.

- login module

- 
- admin module: admin console for the administrator
  - customer module: customer operation panel

## SQL injection

SQL injection is prevented in the application by passing parameters into SQL statements instead of writing queries directly. For example, we would adopt:

```
g.conn.execute("SELECT cid, password FROM Customers WHERE name = %s",  
post_name)
```

rather than :

```
statement = "SELECT * FROM users WHERE name = '" + userName + "';"
```

## How to Run

<http://40.76.22.53:8111>

Administrator Account

Name : admin

Password: adminadmin

Sample User Account

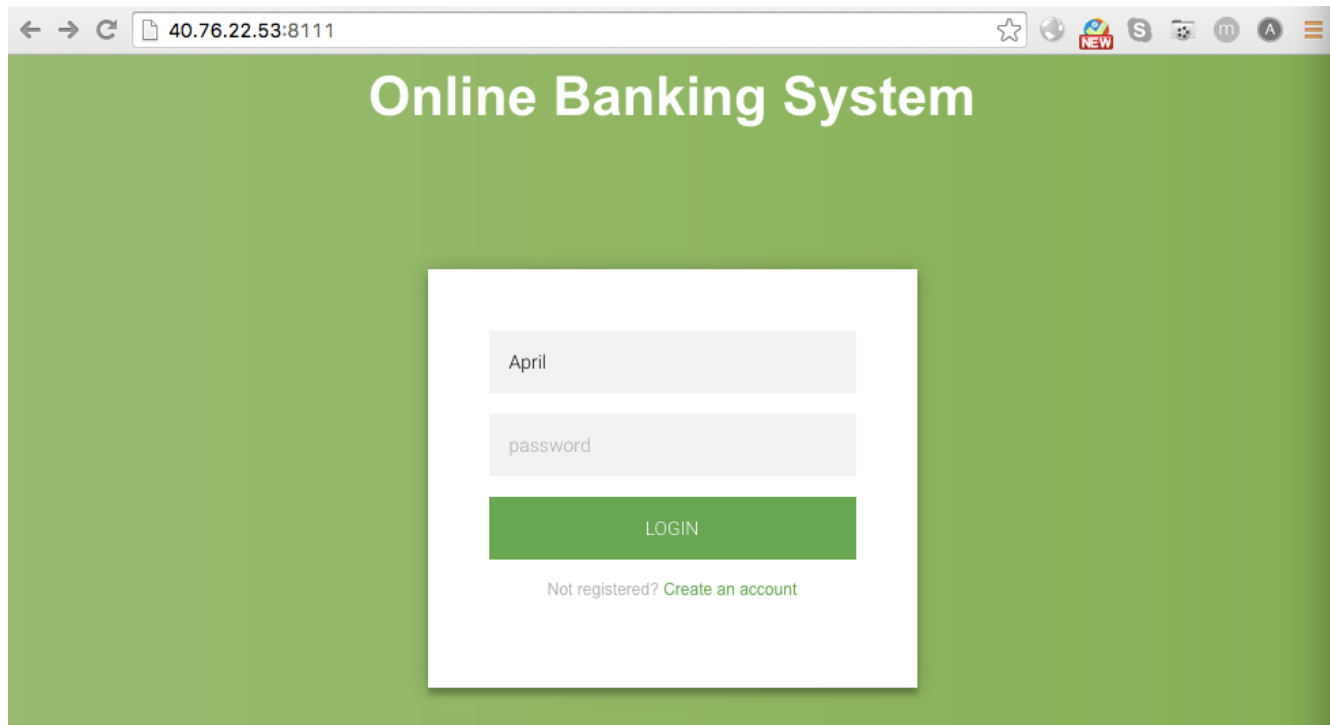
April Chu

password: password

---

## Login module

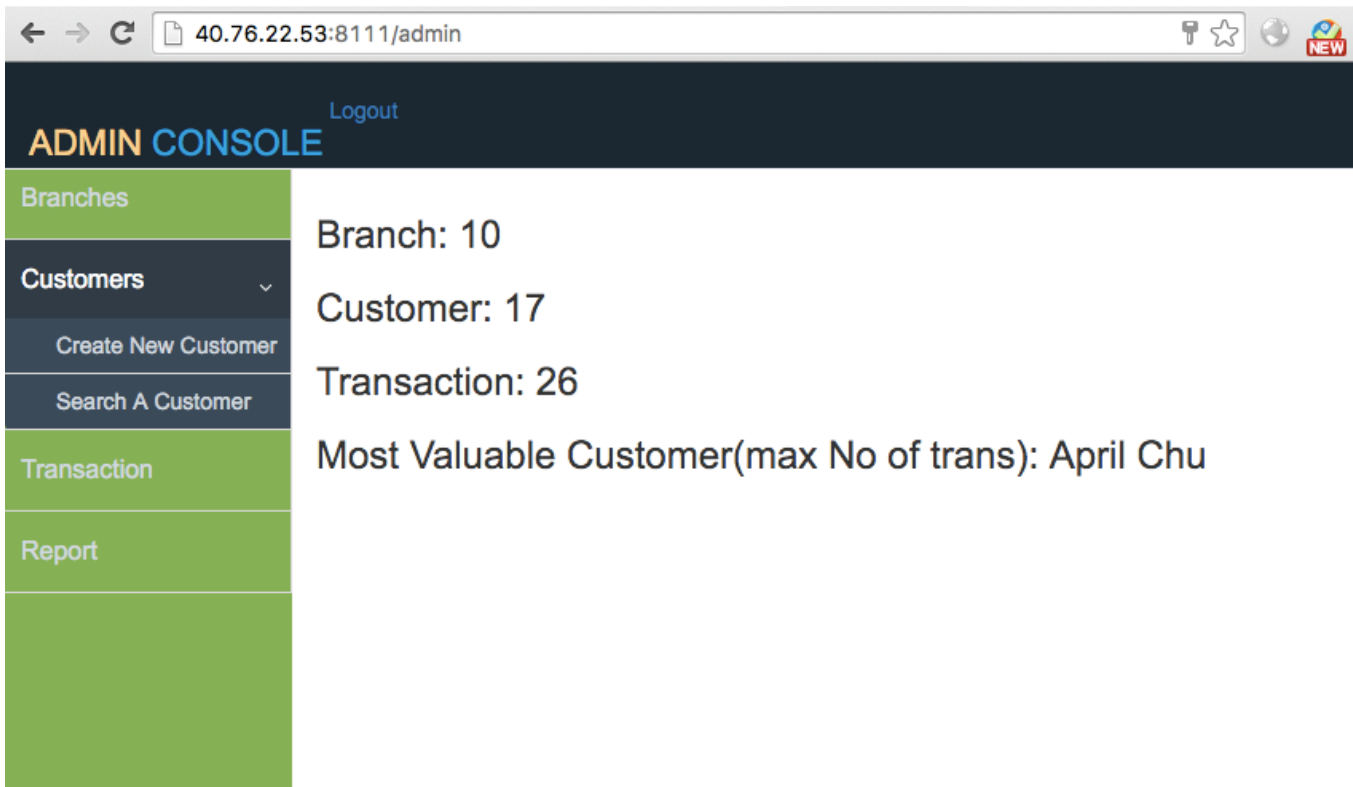
This module handles user login and identify user information, distribute users to the corresponding page, either admin console or customer console. (fig 1)



*figure 1 user login*

## Admin Console module

This module has the following features listed as tab options in the webpage. (fig 2)  
In the index page, we designed the aggregation report to display situation of current database, such as total number of branches, total number of transactions, etc. Particularly, we will display the 'most valuable customer' who has conducted most transactions base on current record.



*figure 2 admin console index page*

- Branches— look up bank branches according to the branch ID or branch name
- Customers
  - Add a new customer and account, which insert a new record in the database
  - Search for customer information, either by customer\_id or name
- Transactions

Search for transactions that satisfy the query statement. Admin is able to search by:

- Start and End date
- Transaction type(ATM, transfer, purchase)

## Customer module

The module is primarily for customer operations, which include:

- Account basic information
  - Display basic information of the account(s), such as balance and debit card NO
- Statement/Transactions

Query all transactions performed by current account

- Transfer Service

Online transfer service which allow customer to transfer money into another account

- Editing Profile

Check and edit current profile

The screenshot shows a web browser window with the address bar displaying '40.76.22.53:8111/statement'. The page has a dark blue header with a 'Logout' link and a 'WELCOME APRIL CHU' message. On the left, there is a green sidebar with navigation links: 'Account', 'See Transactions', 'Transfer', and 'EditProfile'. The main content area displays a table of transactions with columns: Name, Transaction No, Amout, Date, Description, and PayTo. A 'Query' button is located above the table.

Name	Transaction No	Amout	Date	Description	PayTo
April Chu	521325794	-25	2015-04-29	purchase	ebay
April Chu	521325795	-350	2015-05-16	purchase	ebay
April Chu	521325796	-179	2015-08-26	purchase	ebay
April Chu	521325797	-53	2015-10-28	purchase	amazon

*figure 3. Customer Panel: See transactions*