Generic Bank Database System

Cornelius Monahan & John Rosso

CSC 335-01: Database Systems

Mission Statement:

The purpose of the Generic Bank Database System is to maintain client and staff banking data to be used in facilitating simple banking functions for both clients and staff; as well as sharing branch data between branches for convenience and further decision making purposes.

Mission Objectives:

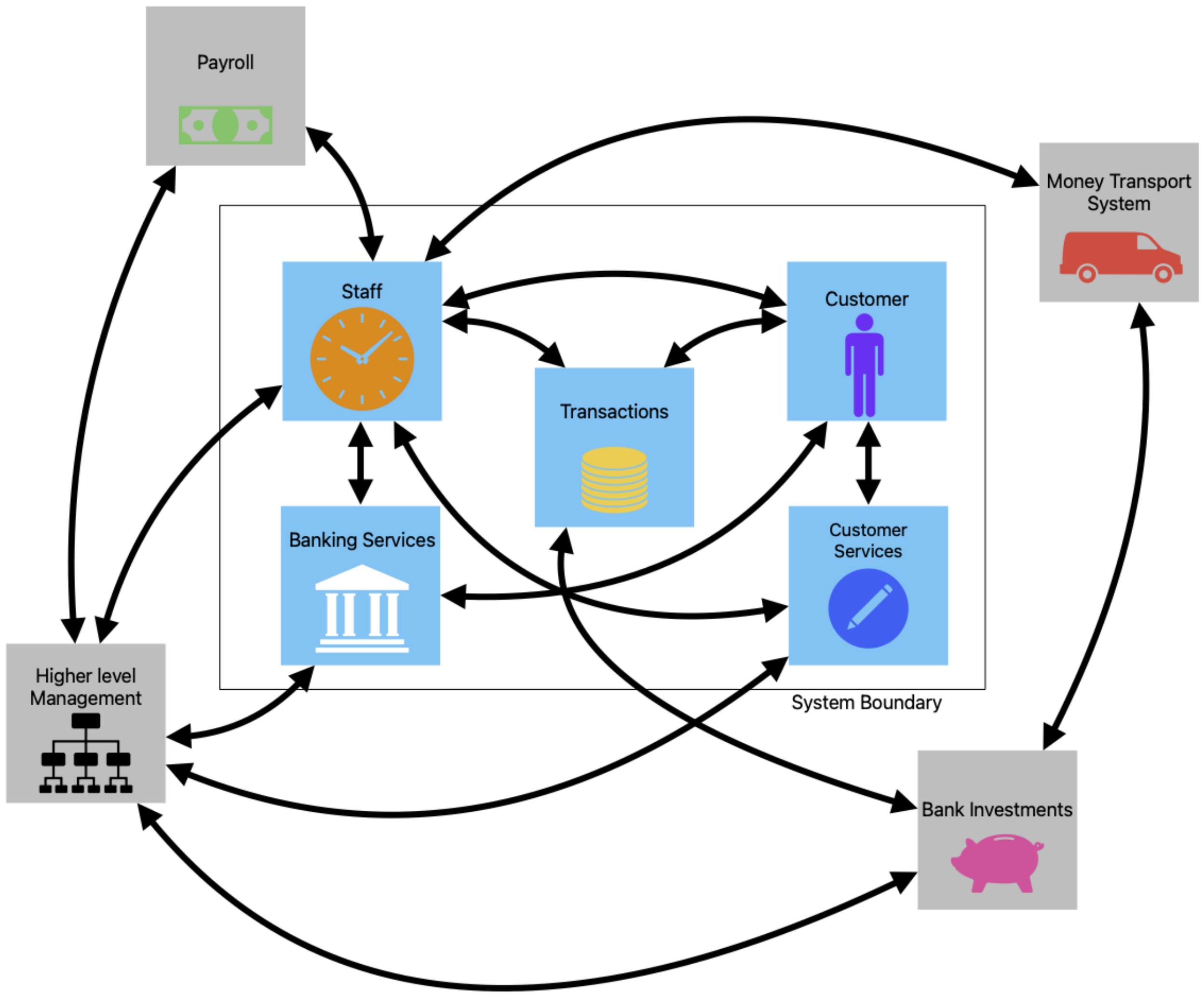
Staff Perspective

- ❖ To manage (add, delete, update) data on bank customers.
- ❖ To manage (add, delete, update) data on bank staff.
- ❖ To manage (add, delete, update) data on bank branches.
- ❖ To manage (add, delete, update) data on customer bank accounts.
- ❖ To manage (add, delete, update) data on given loans (amount, interest, paid to date).
- ❖ Allow for basic transaction services: withdrawal, deposits, loans.
- ❖ Perform searches on customers (account(s) information, branch, address).
- ❖ Perform searches on specific accounts (checking, savings, loans, date opened, additional notes, transaction history).
- ❖ Perform searches on staff (position, branch).
- ❖ Perform searches on branches (address, managers, operating hours).
- ❖ Perform searches on transactions (type, customer, account number, amount, date, additional notes).
- ❖ To track bank cash flow (total physical monies in house, manager access).
- ❖ To track the status of interest rates (to calculate loan interest rates).
- ❖ To track transaction security (flagging of suspicious transactions, information on said transaction).
- ❖ Report on staff (position, salary, hours worked, employment status).
- ❖ Report on client (loan status, account(s) status, customer comments - help notifications).

Customer Perspective

- ❖ Initiate transactions - deposits, withdrawals, loan payments.
- ❖ Access active account(s) - view transaction history and staff notifications.
- ❖ Request customer assistance - leave comments for bank staff to see (ties to staff reports).
- ❖ Manage customer accounts (email, phone numbers, address).

System Boundary Diagram:



Data Items:

Information written in this section is represented in table format, but only represents possible data items needed for each objective outlined under "Mission Objectives". The reasoning for the needed data is given as a brief description with other thoughts. No relations are created.

Customer Data	
Name	First and last name of the customer.
Date of birth	(mm/dd/yyyy)
Address	Street, city, and state.
Phone Number	Numeric sequence of digits.
Email	Any customer email address.
Customer ID	Unique numeric identifier per customer

Staff Data	
Name	First and last name of the staff member.
Date of birth	(mm/dd/yyyy)
Address	Street, city, and state.
Phone Number	Numeric sequence of digits.
Email	Any customer email address.
Staff ID	Unique numeric identifier per staff member.
Staff Branch	Location of the branch in which a staff member works.
Staff Position	Position held by a particular staff member. Also can be used to determine level of access in being able to open/close accounts, as well as work with loan information.
Salary	Salary earned by a particular staff member in USD.

Branch Data	
Address	Street, city, and state in which a particular branch is located.
Number of Employees	The current quantity of employees held at the particular branch.
Branch Status	Identifies as to whether the branch is currently in operation or has been closed.
Cash Held	The current total cash held by the bank in USD

Account Data	
Account Number	Unique numeric identifier to a bank account in which a customer possesses.
Account type	The type of the account possessed by a customer. EX: checking or saving
Account Balance	The total amount of money had within a particular account
Account Status	Current status of the account. EX: active or terminated.
Customer ID	Unique customer ID referenced to relate a customer to a particular bank account.

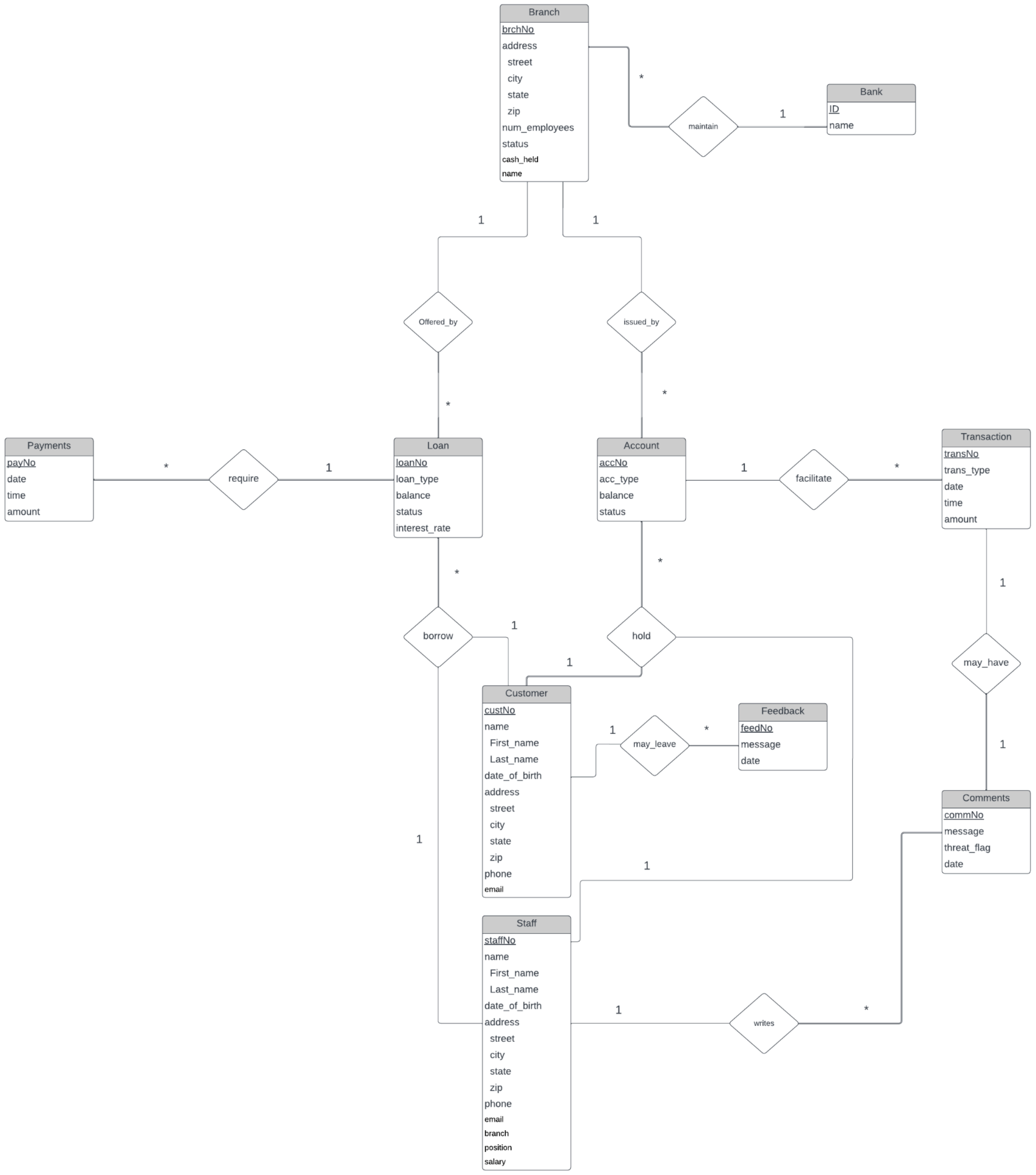
Loan Data	
Account Number	Unique numeric identifier to a loan account in which a customer possesses.
Account type	Specifies that the account is a loan account.
Loan Amount	The total amount of money loaned out.
Interest Rate	Federal interest rate applied to loan - to calculate interest to be paid by customer
Payments to Date	Any payments back on a loan that a customer has made.
Account Status	Current status of the account. EX: active or terminated.
Customer ID	Unique customer ID referenced to relate a customer to a particular bank account.

Transaction Data	
Account Number	Unique account number for bank account being accessed during a particular transaction
Customer ID	Unique customer identifier used to relate a customer to a particular bank account.
Transaction Type	Type of transaction that occurred. EX: deposit, withdrawal, loan payment.
Transaction Date	Date in which the transaction occurred (mm/dd/yyyy).
Transaction Time	Time in which the transaction occurred.

Transaction Comment Data	
Account Number	Unique account number reference for bank account for which a comment can be written.
Customer ID	Unique customer identifier used to relate a customer to a particular note - if left by customer.
Message	The actual string content of the comment left for a transaction if created.
Threat Flag	Boolean variable to denote when leaving transaction notes as to if the transaction was suspicious in any manner. EX: True or False.
Staff ID	Unique numeric identifier per staff member used to reference which staff member left a comment - if left by staff.
Date Written	(mm/dd/yyyy)

Customer Assistance Data	
Help Message	The actual string content that can be provided by a customer for a bank staff member to later view.
Customer ID	Unique customer identifier used to relate a customer to a particular help message - if left by customer.
Date Written	(mm/dd/yyyy)

ER Diagram:


Branch
brchNo
address
street
city
state
zip
num_employees
status
cash_held
name
maintain
Bank
ID
name
Offered_by
issued_by
Payments
payNo
date
time
amount
require
Loan
loanNo
loan_type
balance
status
interest_rate
Account
accNo
acc_type
balance
status
facilitate
Transaction
transNo
trans_type
date
time
amount
borrow
hold
may_have
Customer
custNo
name
First_name
Last_name
date_of_birth
address
street
city
state
zip
phone
email
may_leave
Feedback
feedNo
message
date
Comments
commNo
message
threat_flag
date
Staff
staffNo
name
First_name
Last_name
date_of_birth
address
street
city
state
zip
phone
email
branch
position
salary
writes
1
*

Schema Definitions:

- Bank(ID, name)
- Branch(brchNo,

  Street, City, Zip, State,

  num_employees, status, cash_held, name, *B_ID*) **B_ID is the foreign key of Bank(ID)*
- Loan(loanNo, loan_type, balance, status, interest_rate, *Brch_ID*) **Brch_ID is the foreign key of Branch(brchNo)*
- Customer_Borrows(*Loan_ID*, *Cust_ID*)**Loan_ID is the foreign key of Loan(loanNo) | *Cust_ID is the foreign key of Customer(custNo)*
- Staff_Borrows(*Loan_ID*, *Staff_ID*)**Loan_ID is the foreign key of Loan(loanNo) | *Staff_ID is the foreign key of Staff(staffNo)*
- Payments(payNo, date, time, amount, *Loan_ID*)**Loan_ID is the foreign key of Loan(loanNo)*
- Account(accNo, acc_type, balance, status, *Brch_ID*)**B_ID is the foreign key of Bank(ID)*
- Customer_Holds(Acc_ID, Cust_ID) **Acc_ID is the foreign key of Account(accNo) | *Cust_ID is the foreign key of Customer(custNo)*
- Staff_Holds(Acc_ID, Staff_ID) **Acc_ID is the foreign key of Account(accNo) | *Staff_ID is the foreign key of Staff(staffNo)*
- Customer(custNo,

  first_name, last_name,

  Street, City, Zip, State,

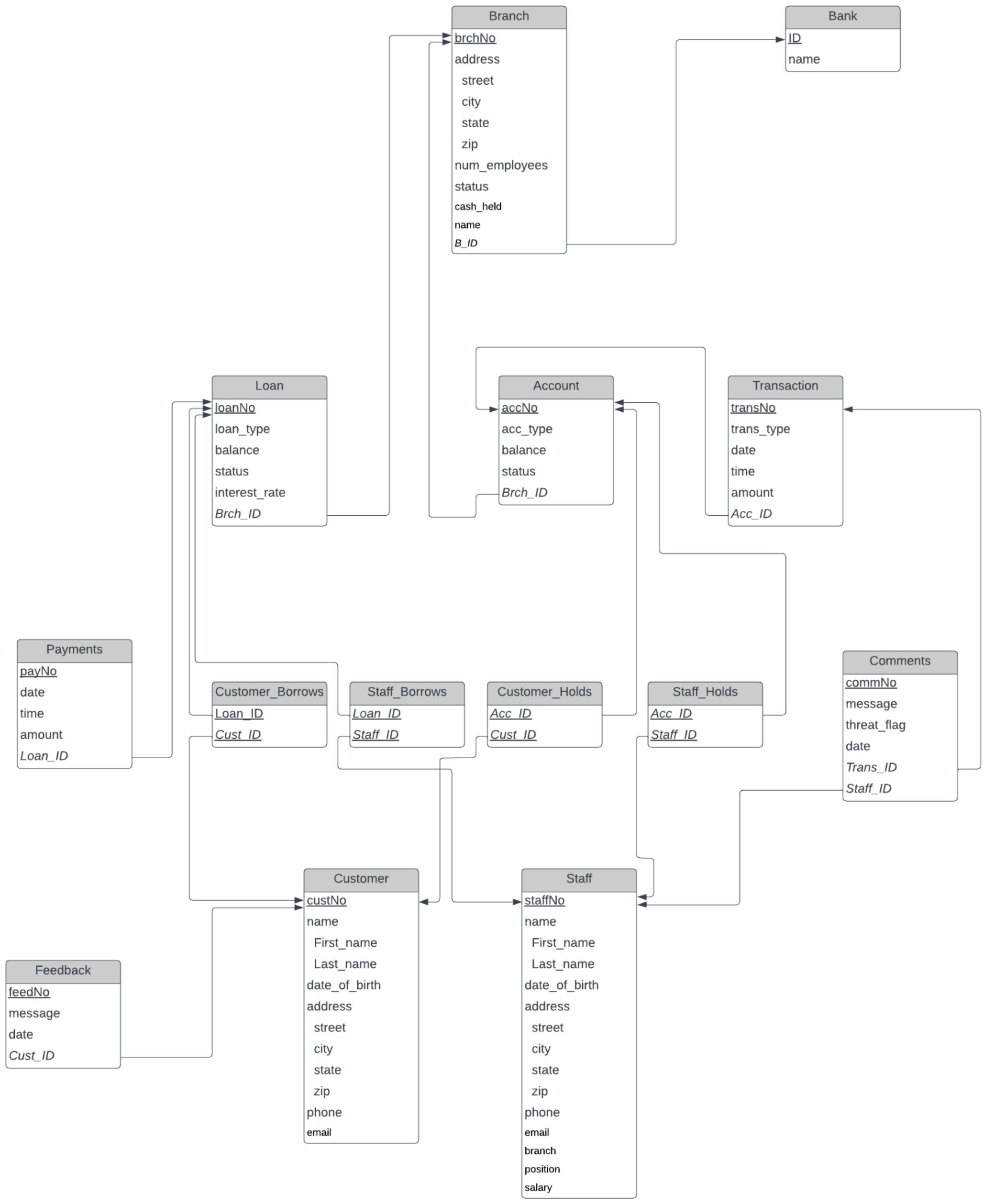
  date_of_birth, phone, email)
- Staff (staffNo,

  first_name, last_name,

  Street, City, Zip, State,

  date_of_birth, phone, email, branch, position, salary)
- Transaction(transNo, trans_type, date, time, amount, *Acc_ID*)**Acc_ID is the foreign key of Account(accNo)*
- Comments(commNo, message, threat_flag, date, *Staff_ID, Trans_ID*)**Staff_ID is the foreign key of Staff(staffNo) | *Trans_ID is the foreign key of Transaction(transNo)*
- Feedback(feedNo, message, date, *Cust_ID*)**Cust_ID is the foreign key of Customer(custNo)*

Schema Diagram:



SQL Queries:

Staff Perspective:

To manage (add, delete, update) data on customer bank accounts.
Add Account:
INSERT INTO Account (acc_type, balance, status, Brch_ID) VALUES (acc_type, balance, ‘active’, Brch_ID);
SELECT brchNo FROM Branch WHERE name = ‘EX_STRING’);
Delete Account:
DELETE FROM Account WHERE accNo = #;
Update Account Information:
UPDATE Account SET status = EX_STATUS WHERE accNo = #;

To manage (add, delete, update) data on given loans (amount, interest, paid to date).
Add Loan:
INSERT INTO Loan (loan_type, balance, status, interest_rate, Brch_ID) VALUES (loan_type, balance, ‘active’, interest_rate, Brch_ID);
SELECT brchNo FROM Branch WHERE name = ‘EX_STRING’);
Delete Loan:
DELETE FROM Loan WHERE loanNo = #;
UPDATE Loan SET status = EX_STATUS WHERE loanNo = #;
UPDATE Loan SET interest_rate = EX_IR WHERE loanNo = #;

Allow for basic transaction services: withdrawal, deposits, loans.

Account Withdrawals:

```
INSERT INTO Transaction (trans_type, date, time,amount,Acc_ID) VALUES ('Withdrawal',
date, time, amount, Acc_ID);

UPDATE Account SET balance = balance - EX_AMOUNT where accNO = (
SELECT ACC_ID FROM Transaction WHERE Acc_ID = # AND date = 'EX_DATE' AND
time = 'EX_TIME');
```

Account Deposits:

```
INSERT INTO Transaction (trans_type, date, time, amount, Acc_ID) VALUES('Deposit',
date, time, amount, Acc_ID);

UPDATE Account SET balance = balance + EX_AMOUNT where accNO = (
SELECT ACC_ID FROM Transaction WHERE Acc_ID = # AND date = 'EX_DATE' AND
time = 'EX_TIME');
```

Loan Payments:

```
INSERT INTO Payments (date, time, amount, Loan_ID) VALUES (date, time, amount,
Loan_ID);

UPDATE Loan SET balance = balance - (SELECT amount FROM Payments WHERE
Loan_ID = # AND date = # AND time = #);
```

Perform searches on branches (address, managers, operating hours).

General Info:

```
SELECT * FROM Branch WHERE brchNo = #;

SELECT * FROM Branch WHERE name = EX_NAME; (name is unique)
```

Address:

```
SELECT street,city,state,zip FROM Branch WHERE brchNo = #;

SELECT street,city,state,zip FROM Branch WHERE name = EX_NAME; (name is unique)
```

Perform searches on transactions (type, customer, account number, amount, date, additional notes).
SELECT t.trans_type, t.date, t.time, t.amount, c.message FROM Transaction AS t JOIN Comments AS c ON t.transNo = c.Trans_ID JOIN JOIN Account AS a ON t.transNo = a.accNo WHERE a.accNo = #;

To track bank cash flow (total physical monies in house, manager access).

SELECT cash_held FROM Branch WHERE brchNo = #;
 	
SELECT cash_held FROM Branch WHERE name = EX_NAME; *(name is unique)*

To track transaction security (flagging of suspicious transactions, information on said transaction).
SELECT t.transNo, t.Acc_ID, t.trans_type, c.message, s.name as "written by",c.staff_ID FROM transaction as t join Comments as c ON t.transNo = c.Trans_ID join Staff as s ON c.Staff_ID = staffNo WHERE c.threat_flag = 1; *(1 = threat | 0 = no threat)*

Report on client (loan status, account(s) status, customer comments - help notifications).

For Customers:

SELECT * FROM Account AS a JOIN Customer_Holds AS ch ON a.accNo = ch.Acc_ID JOIN Customer AS c ON ch.Cust_ID = c.custNo WHERE c.custNo = #;

SELECT * FROM Loan AS l JOIN Customer_Borrows AS cb ON l.loanNo = cb.Loan_ID JOIN Customer AS c ON cb.Cust_ID = c.custNo WHERE c.custNo = #;

For Staff:

SELECT * FROM Account AS a JOIN Staff_Holds AS sh ON a.accNo = sh.Acc_ID JOIN Staff AS s ON sh.Staff_ID = s.staffNo WHERE s.staffNo = #;

SELECT * FROM Loan AS l JOIN Staff_Borrows AS sb ON l.loanNo = sb.Loan_ID JOIN Staff AS s ON sb.Staff_ID = s.staffNo WHERE s.staffNo = #;

For Staff - Review Feedback From Customers Left In Customer Portal

SELECT f.message, f.date, c.First_name, c.Last_name, c.custNo FROM Feedback AS f JOIN Customer AS c ON f.Cust_ID = c.custNo; *(a general search that shows all feedback left in feedback portal for a staff member to review)*

SELECT f.message, f.date, c.First_name, c.Last_name, c.custNo FROM Feedback AS f JOIN Customer AS c ON f.Cust_ID = c.custNo WHERE c.custNo = #; *(specific search for all the feedback left in the feedback portal by a specific customer that can be accessed by providing the customer's unique identification number)*

Add data on bank customers.

```
INSERT INTO Customer
        VALUES(12345, 'Bill', 'Bob', '03/14/2015', '123 Dill Rd.', 'Detroit', 'MI', 45689,
                        123456789, 'bill_bob2@aol.com');
```

Update data on bank customers.

UPDATE Customer
SET phone = 987654321
WHERE custNo = 12345;

Delete data on bank customers.

All customer data:
 	
DELETE FROM Customer
WHERE custNo = 12345;

Specific customer data: DELETE FROM Customer WHERE custNo IN (SELECT custNo FROM Customer WHERE *(add specific data that is to be deleted)*;

Add data on bank staff.

INSERT INTO Staff
VALUES(54321, ‘Frank’, ‘East’, ’01/05/1970’, ‘321 West Rd.’, ‘Portland’, ‘ME’, 87543, 342567432, ‘East_West3@hotmail.com', ‘Main St. Branch’, ‘Teller’, 65000);

Update data on bank staff.

UPDATE Staff
SET salary = 70000
WHERE staffNo = 54321;

Delete data on bank staff.

All staff data:
 	
DELETE FROM Staff
WHERE staffNo = 54321;

Specific staff data: DELETE FROM Staff WHERE staffNo IN (SELECT staffNo FROM Staff WHERE *(add data to be deleted)*;

Add data on bank branches.

INSERT INTO Branches
VALUES(98765, ‘421 End St.’, ‘Monroe’, CT, 07432, 6, ‘open’, 250000, ‘Monroe Branch’, 09876);

Update data on bank branches.

UPDATE Branch
SET num_employees = 8
WHERE brchNo = 98765;

Delete data on bank branches.

All branch data:
 	
DELETE FROM Branch
WHERE brchNo = 98765;

Specific branch data: DELETE FROM Branch WHERE brchNo IN (SELECT brchNo FROM Branch WHERE *(add data to be deleted)*;

Perform searches on customers.

Customer data:
 	
SELECT email
FROM Customer
WHERE custNo = 12345; *(insert information you wish to access)*

Customer account data:
 	
SELECT balance *(insert account information you need)*
FROM Account, Customer_Holds, Customer
WHERE Account.accNo = Customer_Holds.accNo AND Customer.custNo = Customer_Holds.custNo;

Perform searches on specific accounts.

All account data:
 	
SELECT acc_type
FROM account
WHERE accNo = 34521;

All loan data:
 	
SELECT loan_type
FROM Loan
WHERE loanNo = 89043;

Amount of an transaction:
 	
SELECT amount
FROM Transaction, Account
WHERE transNo = 32214 AND Transaction.Acc_ID = Account.accNo;

Transaction message:
 	
SELECT message
FROM Transaction, Account, Comments
WHERE transNo = 32214 AND Transaction.Acc_ID = Account.accNo AND Transaction.transNo = Comments.Trans_ID;

Perform searches on staff.

All staff data:

```
SELECT position
FROM Staff
WHERE staffNo = 54321;
```

Staff account data:

```
SELECT balance (insert account information you need)
FROM Account, Staff_Holds, Staff
WHERE Account.accNo =  Staff_Holds.accNo AND Staff.custNo = Staff_Holds.custNo;
```

To track the status of interest rates.

```
UPDATE Loan
SET interest_rate = 0.026
WHERE loanNo = 23410;
```

Report on staff.

```
SELECT position, salary, branch
FROM Staff
WHERE staffNo = 54321;
```

Customer Perspective:

Access active account(s) - view transaction history and staff notifications.
Transaction History: SELECT * FROM Transaction AS t JOIN Account AS a ON t.Acc_ID = a.accNo JOIN Customer_Holds AS ch ON a.accNo = ch.Acc_ID JOIN Customer AS c ON ch.Cust_ID = c.custNo WHERE c.custNo = #;
Leave Message In Message Portal: INSERT INTO Feedback (message, date, Cust_ID) VALUES (STRING_MSG, date, Cust_ID);

Manage customer accounts (email, phone numbers, address).

Email:
 	
SELECT email FROM Customer where custNo = #;
 	
UPDATE Customer SET email = EX_EMAIL where custNO = #;

Phone:
 	
SELECT phone FROM Customer where custNo = #;
 	
UPDATE Customer SET phone = EX_PHONE where custNO = #;

Address:
 	
SELECT street,city,state,zip FROM Customer where custNo = #;
 	
UPDATE Customer SET street = EX_ST, city = EX_CTY, state = EX_STE, zip = EX_ZIP where custNO = #;

Initiate transactions.

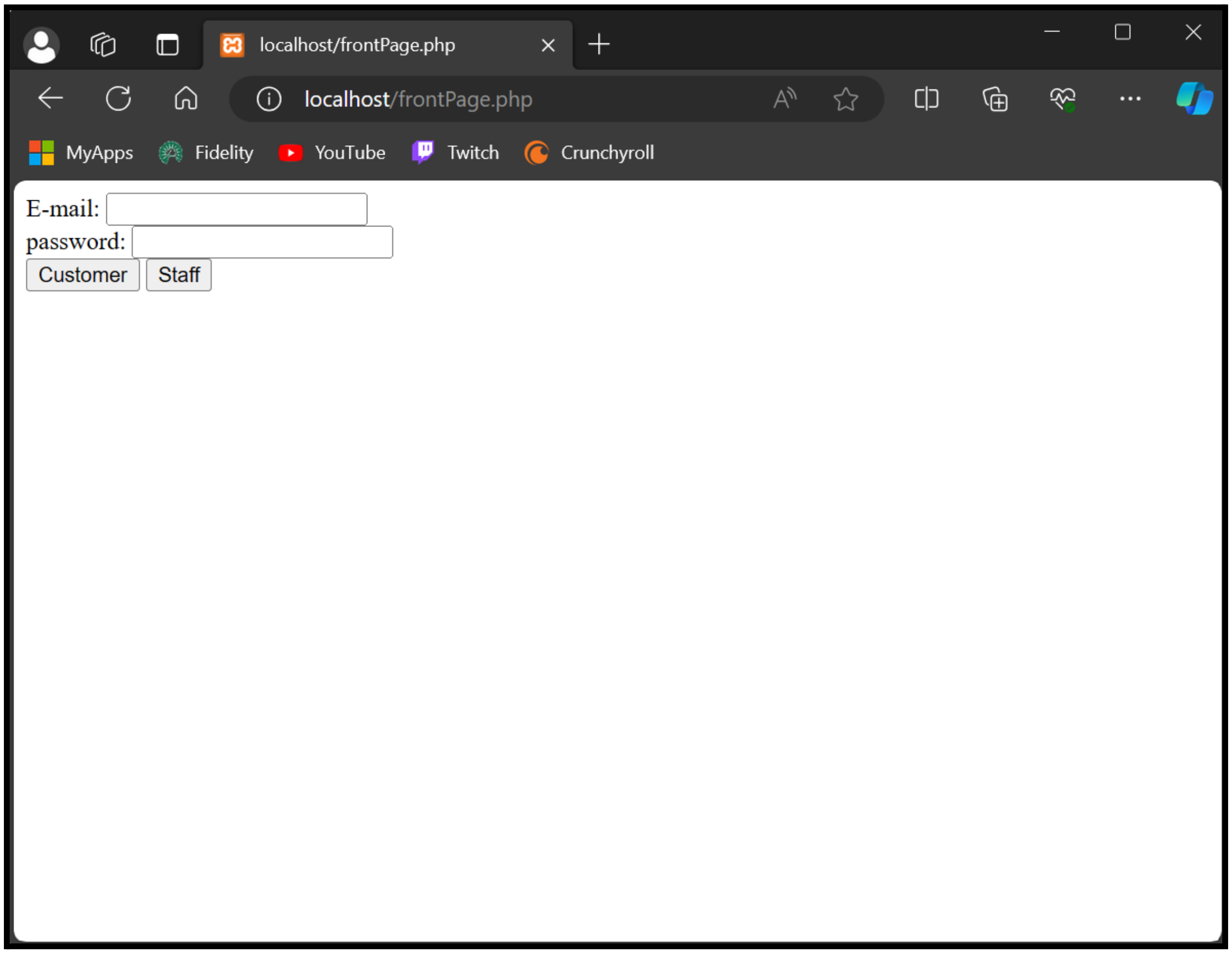
Account transactions (deposit, withdraw):
 	
INSERT INTO Transaction
        VALUES(32456, ‘deposit’, ’12/03/2020’, 19.30, 100.00, 34521);

Loan payments:
 	
INSERT INTO Payments
        VALUES(65789, ’11/02/22’, 10.40, 500.00, 23410);

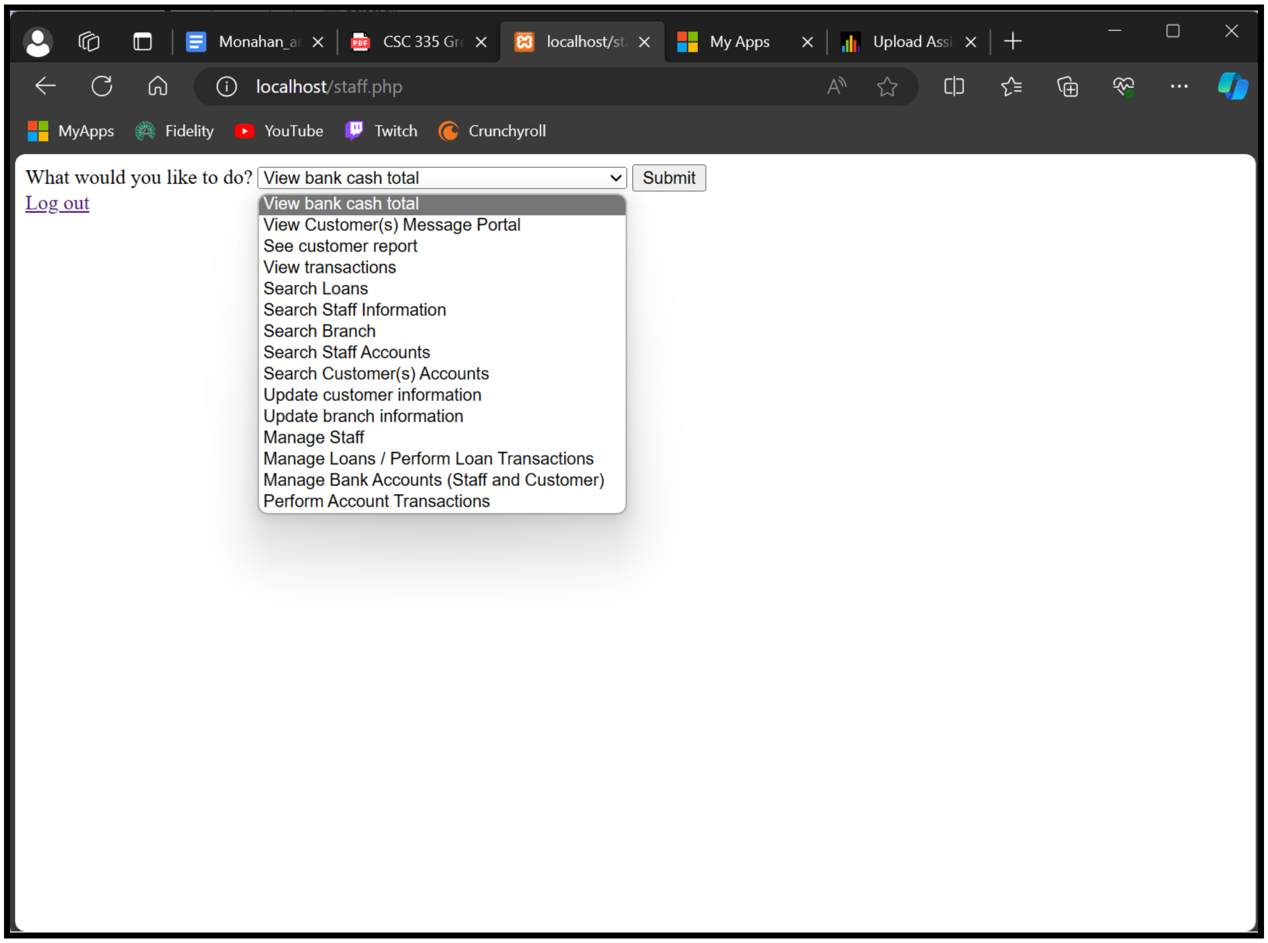
Request customer assistance - leave comments for bank staff to see.

INSERT INTO Feedback
        VALUES(93210, ‘Help message’, ‘03/20’, 12345);

Web Page Screenshots:



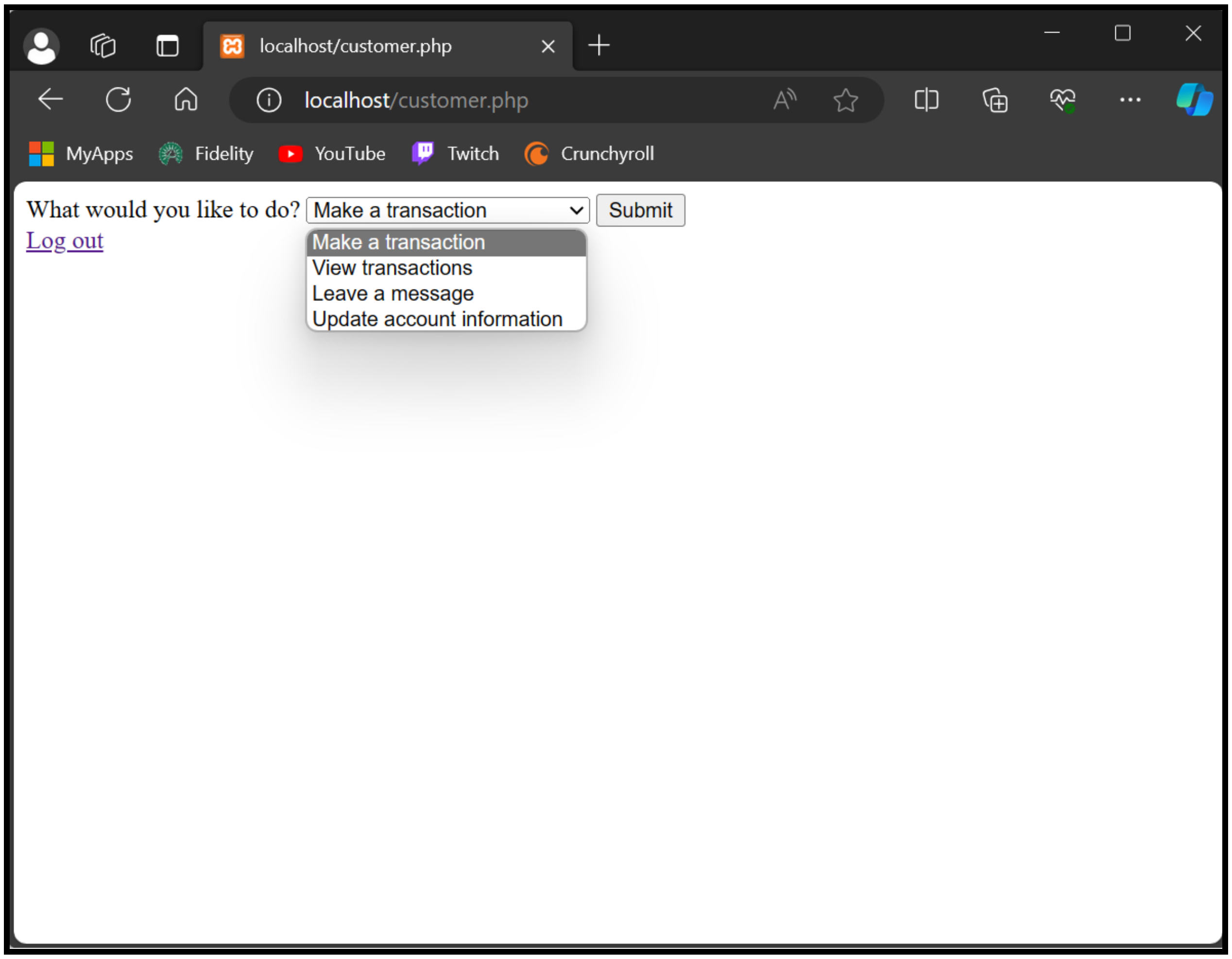
E-mail:
password:



What would you like to do? View bank cash total Submit

Log out

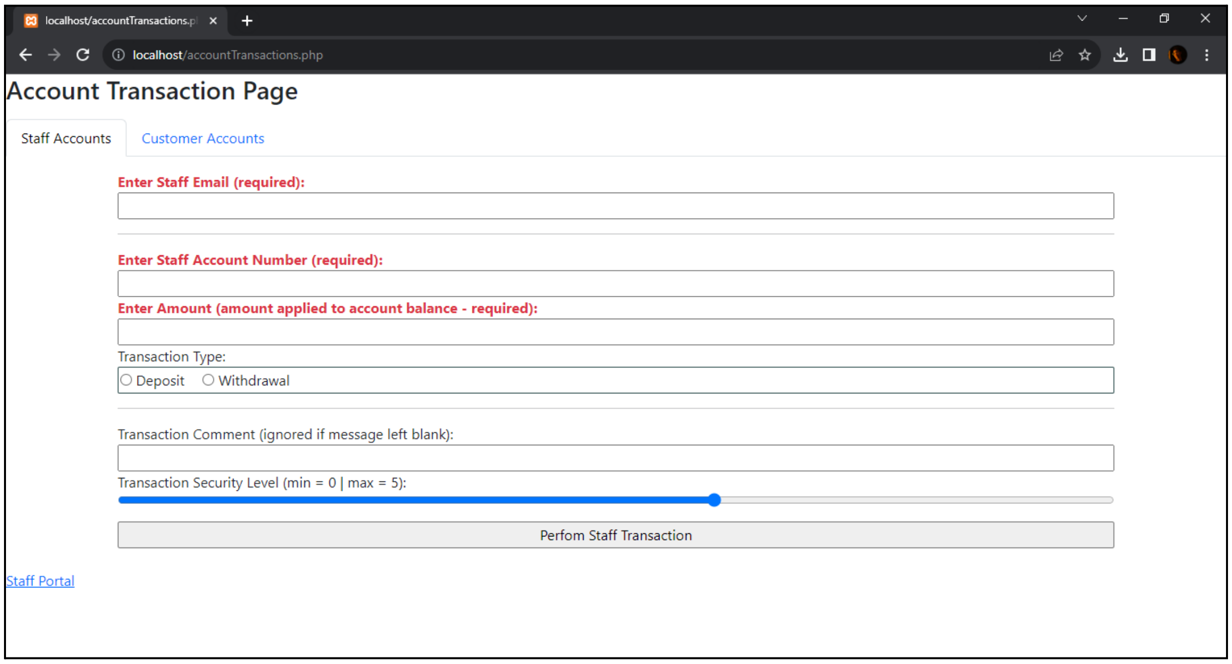
View bank cash total
View Customer(s) Message Portal
See customer report
View transactions
Search Loans
Search Staff Information
Search Branch
Search Staff Accounts
Search Customer(s) Accounts
Update customer information
Update branch information
Manage Staff
Manage Loans / Perform Loan Transactions
Manage Bank Accounts (Staff and Customer)
Perform Account Transactions



What would you like to do? Make a transaction Submit

Log out

Make a transaction
View transactions
Leave a message
Update account information



Account Transaction Page

Staff Accounts | Customer Accounts

Enter Staff Email (required):

Enter Staff Account Number (required):

Enter Amount (amount applied to account balance - required):

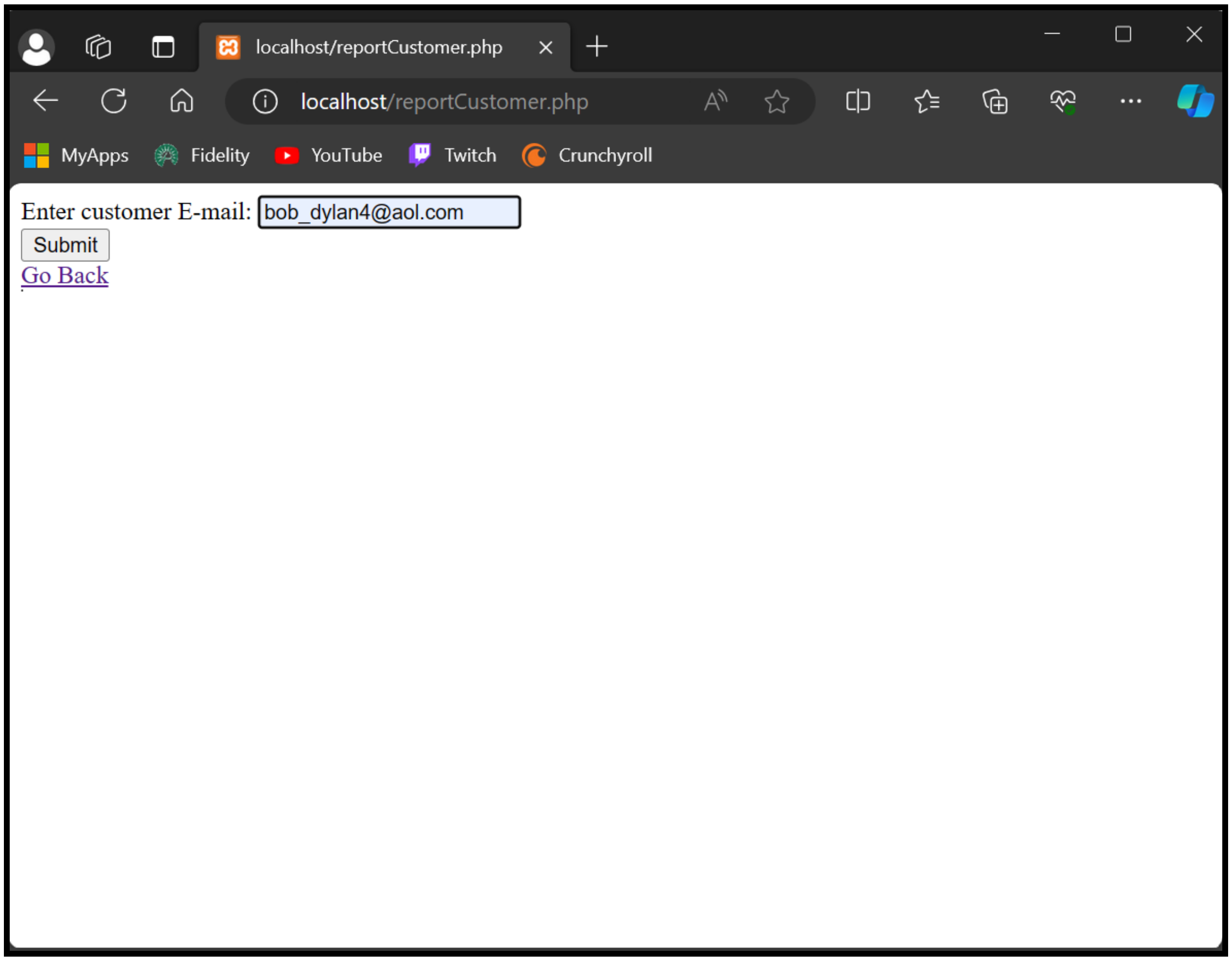
Transaction Type:
○ Deposit ○ Withdrawal

Transaction Comment (ignored if message left blank):

Transaction Security Level (min = 0 | max = 5):

Perfom Staff Transaction

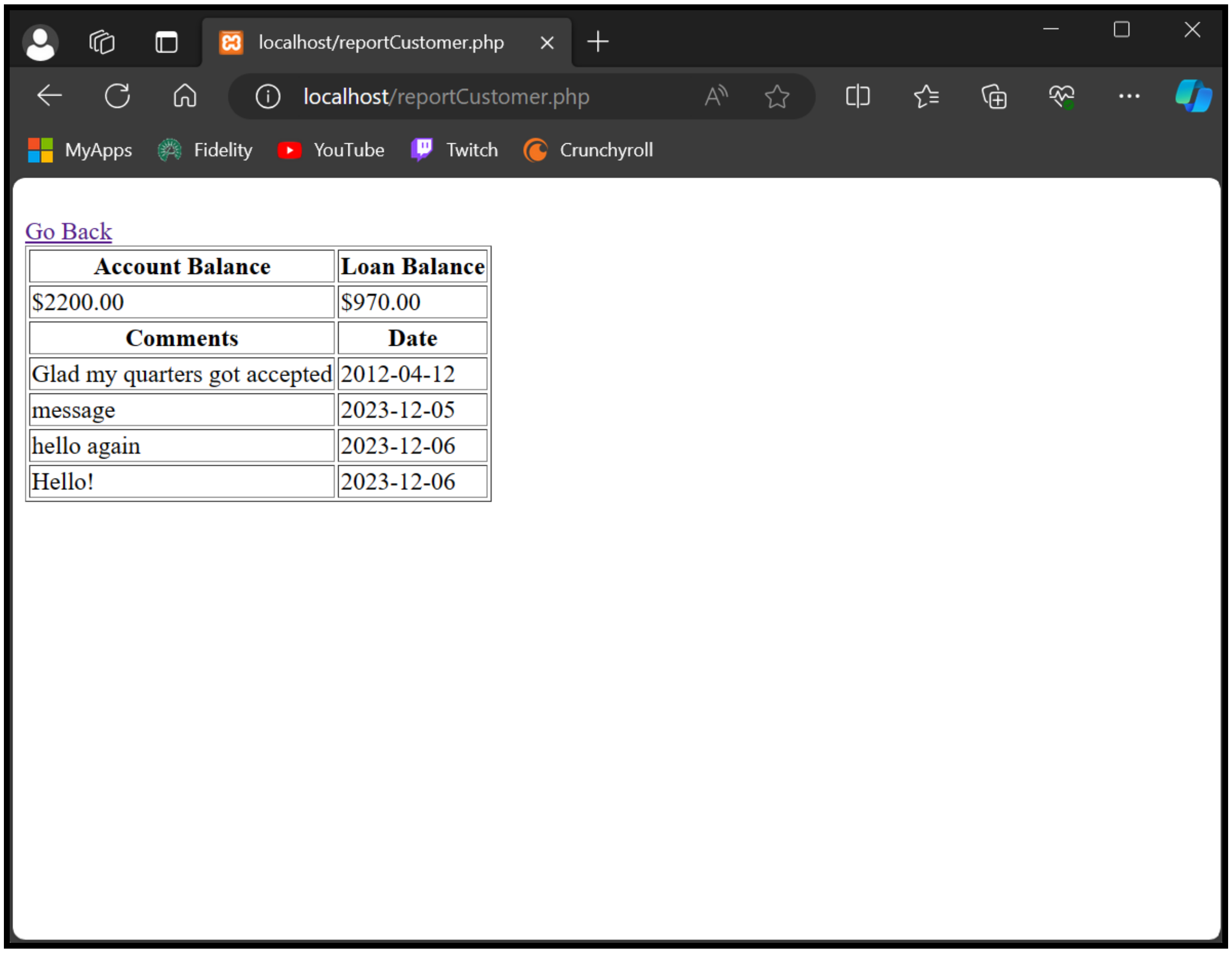
Staff Portal



Enter customer E-mail: bob_dylan4@aol.com

Submit

Go Back

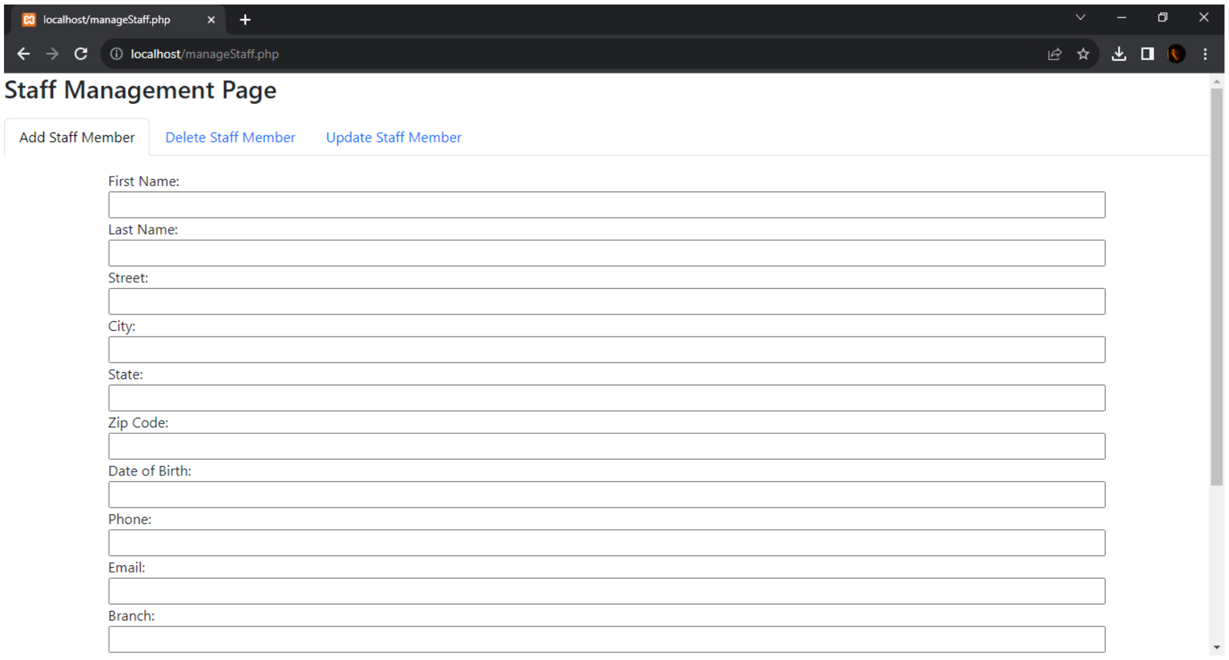


Go Back

Account Balance	**Loan Balance**
$2200.00	$970.00
Comments	**Date**
Glad my quarters got accepted	2012-04-12
message	2023-12-05
hello again	2023-12-06
Hello!	2023-12-06

Manage Staff Example (*These two images count as one example*):

View When Manager (Access Granted):



Staff Management Page

First Name:

Last Name:

Street:

City:

State:

Zip Code:

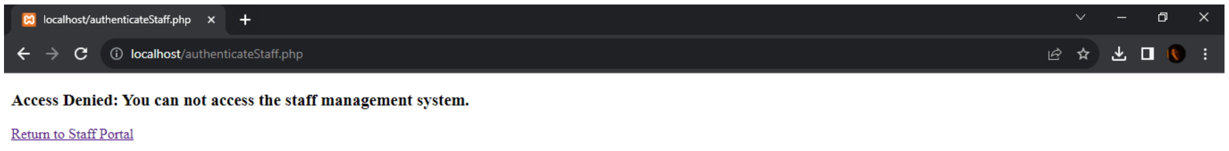
Date of Birth:

Phone:

Email:

Branch:

View When Not Manager (Access Denied):

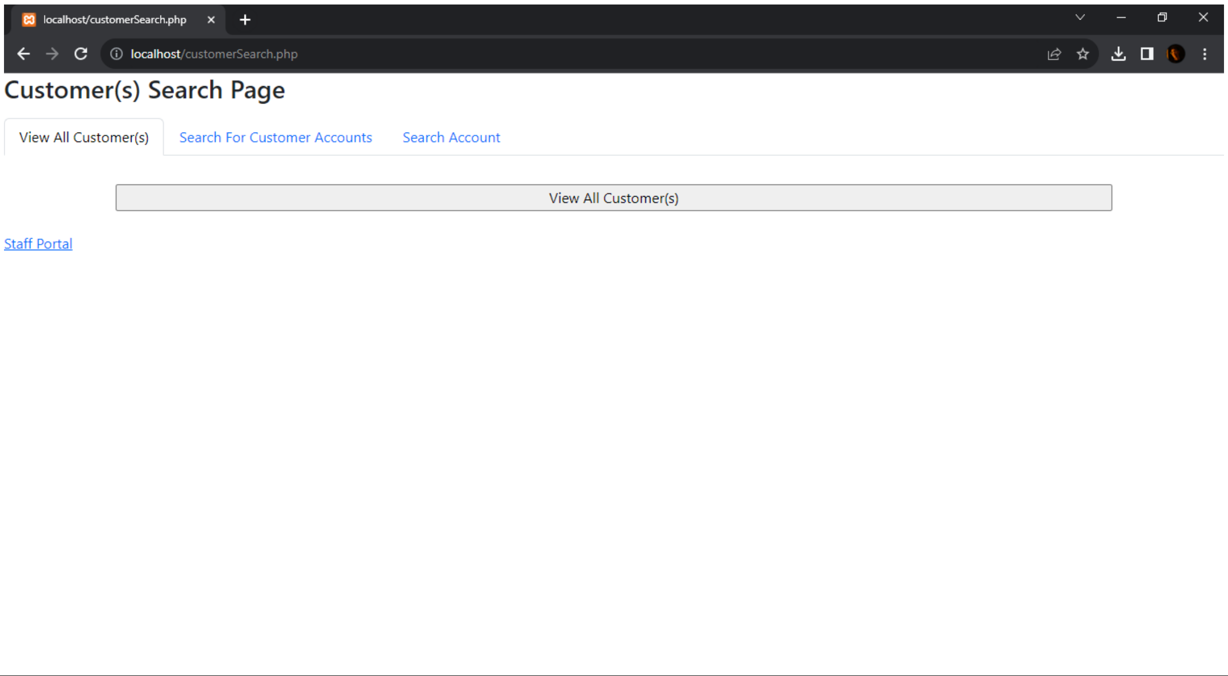

localhost/authenticateStaff.php
localhost/authenticateStaff.php

Access Denied: You can not access the staff management system.

Return to Staff Portal

Customer Search (*These two images count as one example*):

Staff View of Customer Search Page:



localhost/customerSearch.php

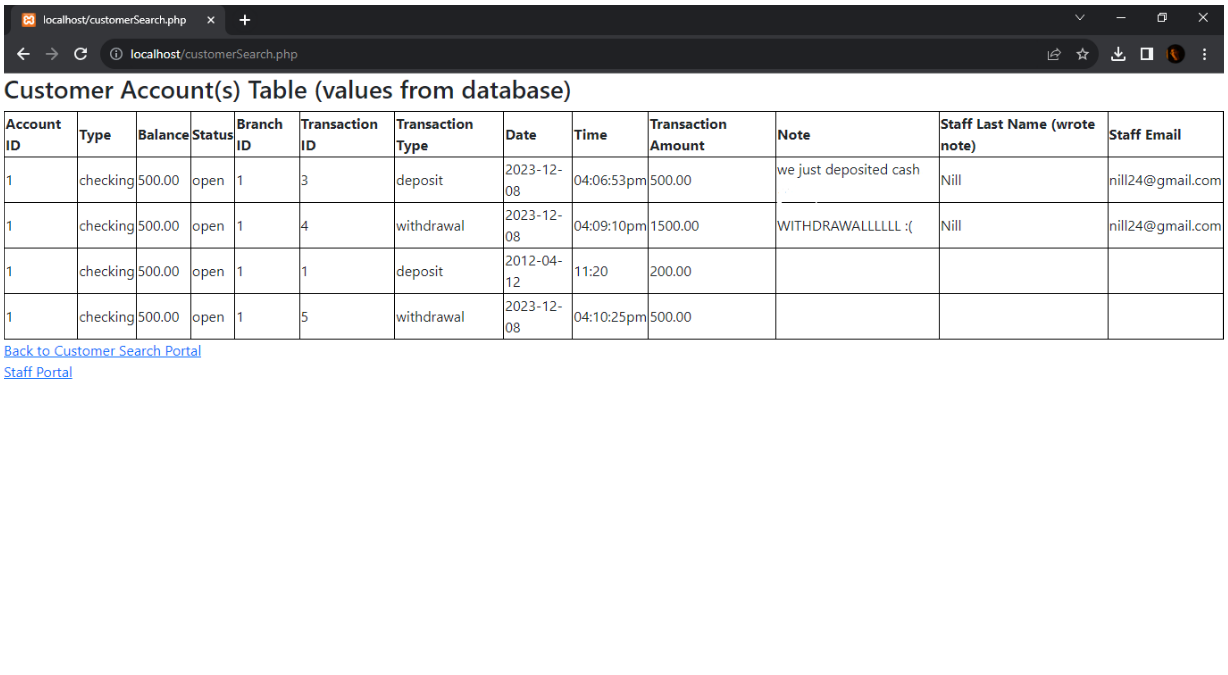
Customer(s) Search Page

View All Customer(s) | Search For Customer Accounts | Search Account

View All Customer(s)

Staff Portal

Result of Viewing A Specific Customer Account (Account ID = 1, Values From Tests):

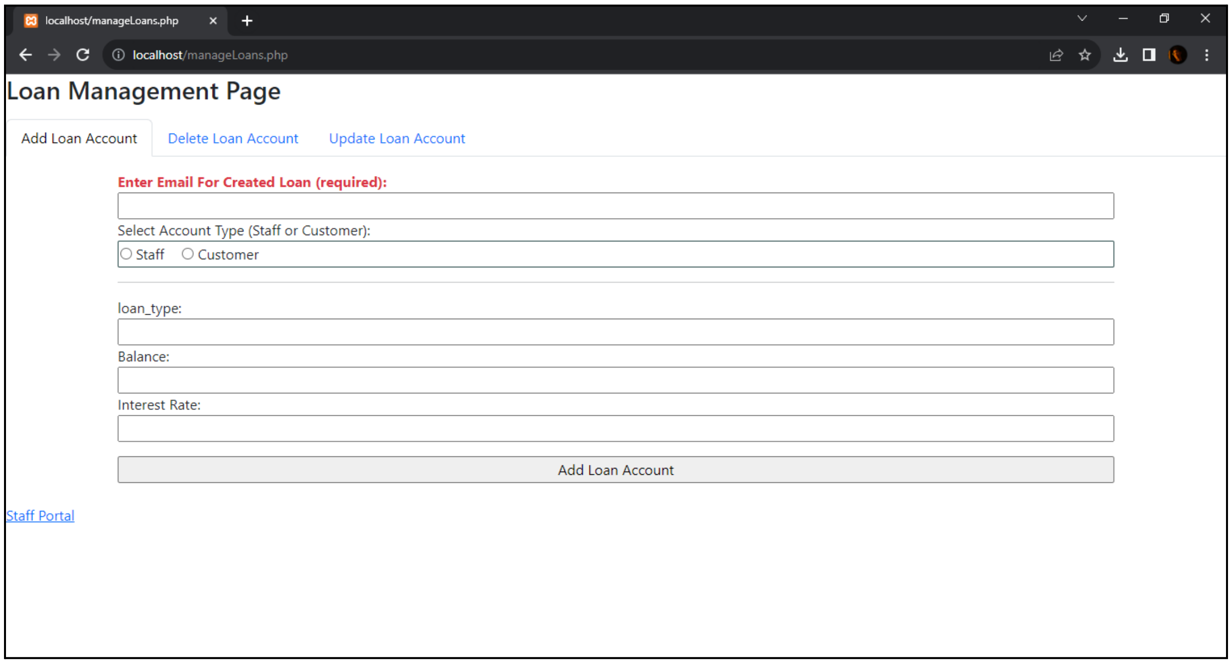


localhost/customerSearch.php

Customer Account(s) Table (values from database)

Account ID	Type	Balance	Status	Branch ID	Transaction ID	Transaction Type	Date	Time	Transaction Amount	Note	Staff Last Name (wrote note)	Staff Email
1	checking	500.00	open	1	3	deposit	2023-12-08	04:06:53pm	500.00	we just deposited cash	Nill	nill24@gmail.com
1	checking	500.00	open	1	4	withdrawal	2023-12-08	04:09:10pm	1500.00	WITHDRAWALLLLLL :(	Nill	nill24@gmail.com
1	checking	500.00	open	1	1	deposit	2012-04-12	11:20	200.00			
1	checking	500.00	open	1	5	withdrawal	2023-12-08	04:10:25pm	500.00			

Back to Customer Search Portal
Staff Portal



localhost/manageLoans.php

Loan Management Page

Add Loan Account | Delete Loan Account | Update Loan Account

Enter Email For Created Loan (required):

Select Account Type (Staff or Customer):
○ Staff ○ Customer

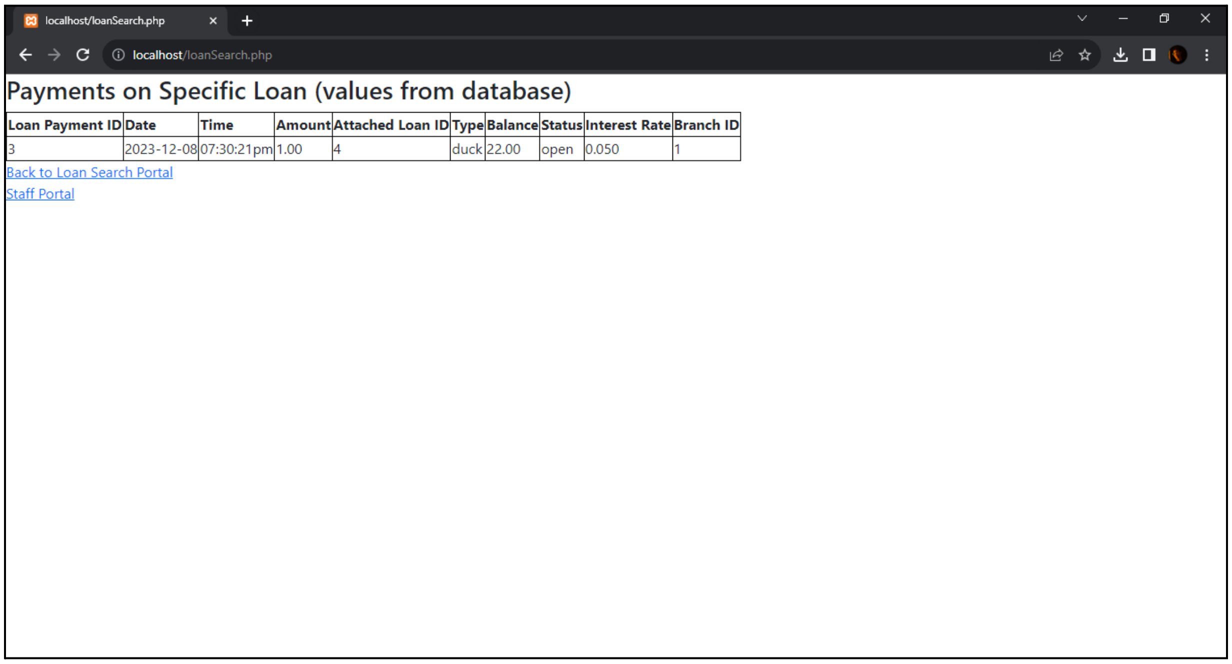
loan_type:

Balance:

Interest Rate:

Add Loan Account

Staff Portal



localhost/loanSearch.php

Payments on Specific Loan (values from database)

Loan Payment ID	Date	Time	Amount	Attached Loan ID	Type	Balance	Status	Interest Rate	Branch ID
3	2023-12-08	07:30:21pm	1.00	4	duck	22.00	open	0.050	1

Back to Loan Search Portal
Staff Portal

Test Cases		
Test	**Expected Outcome**	**Actual Outcome**
Login with correct email and password (Staff & Customer)	User is sent to either staff.php or customer.php	User was sent to appropriate web page
Login with incorrect or no email and password (Staff & Customer)	Web page reloads with no sql or post method sent	Web page reloads and works as if first time using page
In customer.php user selects a task from the select menu and clicks submit	User is sent to appropriate web page based on selected task	User was sent to correct web page for selected task
In customer.php logout link is clicked	User is sent back to frontPage.php as if using this web page for the first time	Link sends user to correct page and acts as if first time going to said page
Clicking go back link in all web pages it is used	Sends user back to previous web page with continued session	Returns user to last used web page in a continued session
In transactionCustomer.php user enters a number value and selects a task button	Web page performs task from clicked button using sql queries and returns users new balance	Web page performs task as asked and returns results to user
In message.php user enters information in text box and clicks submit	Web page sends a sql query to the server and enters a new row in the feedback table. Then returning message sent for the user to see	Web page preforms sql query as expected and returns success message to user
In updateAccountCustomer.php user enters information into text box and clicks button for task they want to have done	Web page performs task selected with sql queries and returns success message	Web page sends and completes sql queries and returns success message
In viewCustomer.php user loads the web page	Web page collects all user transactions and loan payments to be displayed in a table format	Web page performs sql queries to gather user transaction and loan payment data and then displays information in a table format
In staff.php user selects a task	User is sent to appropriate	User was sent to correct web

from the select menu and clicks submit	web page based on selected task	page for selected task
In staff.php logout link is clicked	User is sent back to frontPage.php as if using this web page for the first time	Link sends user to correct page and acts as if first time going to said page
In cash.php web page is loaded	Web page collects branch cash total from database and displays information for user to see	Web page performs sql queries and gather branch cash data and then displays for user to see
In reportCustomer.php user enters customer email in system and clicks submit	Web page gathers information about customers based off of user entered email then displays account balance, loan balance, and any feedback messages customer sent.	Web page performs sql queries and displays customer data as excepted
In viewTransactions.php user clicks a task button	Web page collects either all transactions or transactions with comments and displays information in a table format for the user	Web page performs sql queries and displays information asked for in a table format
In updateCustomer.php user enters new information and customer email into text box and clicks button for task they want to have done	Web page performs task selected with sql queries and returns success message	Web page sends and completes sql queries and returns success message
In updateBranch.php user enters new information into text box and clicks button for task they want to have done	Web page performs task selected with sql queries and returns success message	Web page sends and completes sql queries and returns success message
In customerMessagePortal.php, staff users can view all feedback messages left by bank customers entered at their end of the application.	Staff can view all bank user messages in two different ways: 1. Viewing all messages 2. Viewing messages left by a specific customer using their email as an identifier.	Web page sends and completes sql queries and successfully displays feedback messages.
In loanSearch.php, staff users	Staff can search for held loan	Web page sends and

can search for loans held by staff members or customers. Further, specific loans can be accessed for information on any payments made towards them.	accounts via email for both staff and customers. Staff can use a loan ID number to access loan information and any payments made on a specific loan.	completes necessary sql queries to return any loans held by staff or bank customers, as well as any payments made towards a specific loan.
In staffSearch.php, staff users can search through all staff member information, and search for specific staff.	Staff can search generally through a display of all staff, or retrieve specific information by entering a staff member's email.	Web page sends and completes necessary sql queries to return all staff information and display to the user (general and specific).
In branchSearch.php, staff users can search for all branch information, as well as the information for a specific branch.	Staff can search generally through a display of all 'Generic Bank's' branches or retrieve the specific information for a selected branch by entering the branch's name.	Web page sends and completes necessary sql queries to return all branch information and display to the user (general and specific).
In staffAccountSearch.php, staff users can search for all bank accounts owned by staff members, as well as search for any transactions underwent by a specific account.	Staff can search in two ways: 1. A general search of staff members and the accounts they own. 2. A specific search of a staff account with information on all of its transaction history.	Web page sends and completes necessary sql queries to return the general searched staff account information, or the specific account (and transaction history) of a staff member's account.
In customerSearch.php, staff members can search for customer information, customer accounts, and transactions underwent on specific customer accounts.	Staff can search in three ways: 1. A general search of all customers and customer information. 2. A search on a specific customer that returns the accounts they hold. 3. A search on a specific customer's accounts and any account transaction history.	Web page sends and completes necessary sql queries to return the needed information for all of the three different search options.
In manageStaff.php, access is granted to staff members only to manage staff: add staff,	An authentication process is undergone that redirects managers to the staff	Web page sends and completes necessary sql queries to facilitate all

delete staff, and update staff information.	managing portal, and tellers back to the staff general search page. Managers can add, delete, and update staff information.	managing operations (for managers only).
In authenticateStaff.php, authorization status is evaluated for staff members, and staff management access is only allowed for managers.	This is a background redirect process not seen by users. A staff member's status ('Manager' or 'Teller') is checked, and managers are redirected to the staff management portal, whereas tellers are redirected to the staff general search page.	Web page sends and completes necessary sql queries to authenticate staff members and redirect the user to the appropriate landing page.
In manageLoans.php, Staff members can create loans for staff and customers, delete specific loan accounts, and update loan accounts (which includes updating the payments made towards paying them off).	Staff can: 1. Add loans 2. Delete loans 3. Update loan type, status, interest rate, and payments. Updating payments deducts from the standing loan balance.	Web page sends and completes necessary sql queries to facilitate all loan managing operations.
In manageBankAccounts.php, staff members can add bank accounts for both staff and customers, delete bank accounts, and update bank account information.	Staff can: 1. Add bank accounts 2. Delete bank accounts 3. Update bank account type, status, and balance. Updating the balance manually overrides the current account balance, and automatically attaches a transaction history note to the account stating that a staff member manually altered the account.	Web page sends and completes necessary sql queries to facilitate all bank account management operations.
In accountTransactions.php,	Staff can perform bank	Web page sends and

staff members can perform bank account transactions on both staff and customer accounts.	account transactions for both staff and customer accounts. Transactions will only be allowed if the selected account has the balance required for the transaction (doesn't allow the bank account to over withdrawal).	completes sql queries needed to facilitate the account transaction operations.

Key:
- ☐ Test cases written by John Rosso
- ☐ Test cases written by Cornelius Monahan

Contributions:

- Mission Statement: Team Effort
- Mission Objectives: Team Effort
- System Boundary Diagram: John Rosso
- List of Data Items: Cornelius Monahan
- ER Diagram: Team Effort
- Schema Definitions: Team Effort
- Schema Diagram: Team Effort
- SQL Queries: Team Effort
- Snapshots of webpages (UI): Team Effort
- Test Cases: Team Effort
- Web App Pages:
    - frontPage.php: John Rosso
    - customer.php: John Rosso
    - staff.php: John Rosso
    - cash.php: John Rosso
    - message.php: John Rosso
    - reportCustomer.php: John Rosso
    - transactionCustomer.php: John Rosso
    - updateAccountCustomer.php: John Rosso
    - updateBranch.php: John Rosso
    - updateCustomer.php: John Rosso
    - viewCustomer.php: John Rosso
    - viewTransactions.php: John Rosso

    - customerMessagePortal.php: Cornelius Monahan
    - loanSearch.php: Cornelius Monahan
    - staffSearch.php: Cornelius Monahan
    - branchSearch.php: Cornelius Monahan
    - staffAccountSearch.php: Cornelius Monahan

- customerSearch.php: Cornelius Monahan
- manageStaff.php: Cornelius Monahan
- authenticateStaff.php: Cornelius Monahan
- manageLoans.php: Cornelius Monahan
- manageBankAccounts.php: Cornelius Monahan
- accountTransactions.php: Cornelius Monahan