

# PROJECT 2

*A Secure Teller Terminal System  
COMP 2710-001 (SPRING 2017)*

## OVERVIEW

---

An analysis of *design and implementation of teller terminals* for Tiger Bank. An application which handles local bank transactions and integrable with other Tiger Bank branches across the state.



### *Sections:*

- (i) Analysis: Use Case Interactions and Program Flow*
  - (ii) Design: System Objects and Encapsulated Data*
  - (iii) Testing: Finding Bugs and Detecting Regressions*
-

## *(i) Analysis::*

# USE CASE ELABORATION

<b>USE CASE 1: LOGIN TO SECURE TELLER TERMINAL SYSTEM</b>	
<b><u>SUMMARY:</u></b>	An idle teller terminal displays a login menu. It invites a branch employee to enter a userid and password and either (i) to manage client and account information or (ii) add/delete a user (branch staff) if the user is a system administrator or (iii) to change his/her password (in which case, the teller is first authenticated using the current password, then the password is updated, and finally a session is initiated with the authenticated teller). In all the cases, the information provided is checked for validity. If the user name and password are valid, then the teller terminal system is placed in the active state and a session is started.
<b><u>ACTOR(S):</u></b>	SystemMenu, User
<b><u>PRECONDITION:</u></b>	<ul style="list-style-type: none"><li><b><u>(i)</u></b> SystemMenu prompts user with welcome message</li><li><b><u>(ii)</u></b> User sees option to either 1) Login or 2) Quit</li><li><b><u>(iii)</u></b> User is prompted to input an option</li></ul>
<b><u>CONTROL FLOW:</u></b>	<ul style="list-style-type: none"><li><b><u>(1)</u></b> User enters in an option (1 or 2)<ul style="list-style-type: none"><li><b><u>(1.1)</u></b> Option 1) displays Login to access the system</li><li><b><u>(1.2)</u></b> Option 2) exits and terminates the program</li></ul></li><li><b><u>(2)</u></b> User is prompted for existing User Name</li><li><b><u>(3)</u></b> User name is stored in a temp string variable</li><li><b><u>(4)</u></b> User prompted for password</li><li><b><u>(5)</u></b> User password is stored in a temp variable</li><li><b><u>(6)</u></b> Temp username and temp password are compared against all usernames and passwords in the file containing staff member attr's.<ul style="list-style-type: none"><li><b><u>(6.1)</u></b> If both username and password exists and match on the same input stream line (i.e. getline() ), user is redirected to appropriate menu display based on their roles.</li><li><b><u>(6.2)</u></b> If both equalities are not satisfied, SystemMenu outputs appropriate cerr message, stating username or password was incorrect and then repeats prompt for username and password.</li></ul></li></ul>
<b><u>ALTERNATE FLOW:</u></b>	<ul style="list-style-type: none"><li><b><u>(1.3)</u></b> User enters anything besides 1 or 2 displays error message, followed by the option display, and then re-prompted for option. (optional: consecutive attempts can cause program termination).</li></ul>
<b><u>POST CONDITIONS:</u></b>	<ul style="list-style-type: none"><li><b><u>(6.1.1)</u></b> If user role is system admin, admin menu displays</li><li><b><u>(6.1.2)</u></b> If user role is branch staff only, branch staff menu displays</li></ul>

**USE CASE 2:****ADD A BRANCH STAFF MEMBER****SUMMARY:**

A system administrator can add new users (branch staff members) to the teller terminal system. When a new user is added into the teller system by the system administrator, user name and password of the new staff member must be initialized. Empty value of the user name or password is not acceptable.

**ACTOR(S):**

SystemMenu, AdminMenu, User

**PRE-  
CONDITIONS:**

- (i) User must have already logged in and in an 'active' menu
- (ii) User must be in the "AdminMenu" and see displayed admin menu options
- (iii) User info (name, pw, and role) must be populated in the appropriate user struct for the current user
- (iv) User is prompted for option and has entered in option 2) Add a branch staff member

**CONTROL  
FLOW:**

- (1) User is prompted to enter in a new user name
- (2) User enters in the desired username to console
- (3) User is then prompted to enter in a desired password
- (4) User enters in desired password for new user to console
- (5) Console displays option to enter in a role for the new branch employee 1) Administrator or 2) Branch Staff
- (6) User enters in 1 or 2 to specify staff type
- (7) Console displays option to confirm or cancel creation of employee and prompts user to 1) Confirm or 2) Cancel
- (8) User confirms creation of branch staff employee or administrator and a new employee is created with an employee record created

**ALTERNATE  
FLOW:**

- (2.1) If user enters user name with Non-alpha numeric characters or spaces, user will be prompted to enter in a valid entry and allowed to try again from the system administrator menu
- (2.2) If user enters in a pre-existing user name, user will be prompted that the action is not allowed because the user already exists and will be allowed to try again from the system administrator menu
- (4.2) If user enters in a password that consists of spaces or Non-alpha numeric characters, user will be prompted that the action is not allowed, and the user will be allowed to try again from the system administrator menu
- (6.1) If user enters in any value other than 1 or 2, console will display invalid entry and user will be allowed to try again until success. (Optional: more than 3 failures will exit the program)
- (7.1) If user enters in any value other than 1 or 2, console will display invalid entry, a new user will not be created, and user will be allowed to try again until success. (Optional: more than 3 failures will exit the program)

**POST  
CONDITIONS:**

- (8) User is returned to the appropriate teller terminal system menu.

**USE CASE 3:     ADD A CLIENT****SUMMARY:**

From the client and account management menu, if “Add a client” is selected, the new client's name, address, social security number, employer, and annual income must be entered. For simplicity, we assume that client names are unique, meaning that we can use client names as client identifiers.

**ACTOR(S):**

ClientMenu, Client

**PRE-  
CONDITION:**

- (i)** User must be logged in as branch staff employee or system admin
- (ii)** User must have selected option 1) from the Teller system active menu
- (iii)** User must be in the client and account management menu and have selected option 1) to add a client

**CONTROL  
FLOW:**

- (1)** Console informs user that a new client will be added and then prompts the user for client's name
- (2)** User enters client name
- (3)** System asks for user to enter client's address and user enters in an address
- (4)** System prompts user for client's SSN and user enters in client's SSN
- (5)** System prompts user for client's employer and user enters in client's employer
- (6)** System prompts user for client's annual income and user enters in client's annual income
- (7)** System reports a client was added and a client record is created

**ALTERNATE  
FLOW:**

- (2.1)** If user enters a name that contains numbers or special characters user will be prompted with the specified error and allowed to reenter the client's name.
- (2.2)** If user enters in a name that contains special characters (excluding hyphen), user will be prompted with the specified error and allowed to reenter a valid username.
- (2.3)** If user enters in more than two fields (first name + last name) than user will be asked for only client's first and last name and allowed to reenter client's name.
- (3.1)** If the client's zip/postal code is not equal to 5 characters, than user will be prompted to reenter the address with a valid zip/postal code.
- (3.2)** If the client's address contains ASCII characters 33, 34, 36 - 43, 47, 58-64, 91 - 96, or 124 - 126, user will be asked to reenter client's address without illegal address characters.
- (4.1)** If user enters more than 9 numeric characters, user will be asked to reenter client's SSN with only 9 numeric characters excluding the hyphen.
- (4.2)** If user enters non-numeric characters values, user will be asked to enter only numeric values.
- (6.1)** If user enters non-numeric characters values, user will be asked to enter only numeric values.

**POST  
CONDITIONS:**

- (8)** User is returned to the client and account management menu

**USE CASE 4:****ADD AN ACCOUNT****SUMMARY:**

If “Add an account” is selected, a client's name must be entered first. If the client is not found in the system, an error message will pop up. If the client is in the system, then the branch staff member has to enter account number, account type, and account balance.

**ACTOR(S):**

ClientMenu, Client, Account

**PRECONDITION:**

- (i)** User must be logged in as branch staff employee or system admin
- (ii)** User must have selected option 1) from the Teller system active menu
- (iii)** User must be in the client and account management menu and have selected option 2) to add an account

**CONTROL  
FLOW:**

- (1)** User will be prompted to first enter in the name of a current client
- (2)** User enters in the name of the client and system prompts user that a client account will be created
- (3)** User will be prompted to enter in a unique client account number
- (4)** User will enter in the unique client account number and then will be prompted to enter in the account type.
- (5)** User enters in either "Checking" or "Savings" account and then is prompted to enter in a balance amount.
- (6)** System prompts that an account was created for the specified client and then pauses the command line so user may enter any key to continue.

**ALTERNATE  
FLOW:**

- (2.1)** If client entered does not exist, user will be prompted to reenter client name
- (4.1)** If the unique account number already exists on file, user will be asked to enter in a different account number
- (5.1)** If user enters in an invalid account type (anything besides checking and savings), user will be prompted to reenter a valid account type.

**POST  
CONDITIONS:**

- (7)** Continuation returns user to client and account management menu.

**USE CASE 5:****WITHDRAW FUNDS FROM MANAGE ACCOUNT****SUMMARY:**

If "Manage an account" is selected, an account number will be entered. If the account does not exist in the system, an error message will pop up. The format of the error message is given: "Error – Account <Account\_Number> is not in the system!" After the account is chosen, the staff can either deposit or withdraw funds by choosing a menu option. Thus, if the account exists in the system, the following menu will appear:

Manage account <Account\_Number> for <Client Name> ...

- 1) Deposit
- 2) Withdraw
- 3) Cancel

**ACTOR(S):**

ClientMenu, Client, Account

**PRECONDITION:**

- (i) User must be logged in as branch staff employee or system admin
- (ii) User must have selected option 1) from the Teller system active menu
- (iii) User must be in the client and account management menu and have selected option 4) Manage an account

**CONTROL FLOW:**

- (1) Once user specifies that they will be managing an account, the system will ask user for an existing account number to manage
- (2) User will enter in an existing account number
- (3) System will display account management options 1) Deposit 2) Withdraw and 3) Cancel
- (4) User will specify account withdrawal with option number 2
- (5) System will display "Withdraw Amount: " and then prompt user for an amount to withdraw
- (6) User will enter in an amount to withdraw

**ALTERNATE FLOW:**

- (2.1) If account number entered does not exist
- (4.1) If user enters an invalid number, system will inform user they have entered in an invalid option and allow user to reenter the appropriate option
- (4.2) If user enters in an invalid character, system will report user is not entering in a valid option and allow user to reenter a valid option
- (4.3) (Optional) If user enters in an invalid option more than once, user will be returned to the manage account option menu
  - (4.3.1) If user chooses option 1, user will initiate account deposit
  - (4.3.2) If user chooses option 3, user will be returned to the client and account management menu
- (6.1) If the amount of withdrawal is more than the client's available balance, the system will inform the teller that the client's balance cannot be adjusted due to insufficient funds the manage account option menu
  - (6.1.1) If user chooses option 1, user will initiate account deposit
  - (6.1.2) If user chooses option 3, user will be returned to the client and account management menu
- (6.2) If the user enters in an alphanumeric value, the system will inform the user to enter in only numbers and user will be allowed to reenter the withdrawal amount.

**POST CONDITIONS:**

- (7) Once the user has successfully entered in a withdrawal amount, the system will output message containing the account modified and the new balance after the account has been adjusted.
- (8) User will then be returned to the manage account option menu until they select cancel. Once user selects option 3) to cancel, user will be returned the client and account management menu

*(ii) Design::*

## CLASSES AND STRUCTS

<u>CLASS</u>	<u>SystemMenu</u>
<u>DESCRIPTION</u>	SystemMenu is the primary menu class comprising of all other menu class objects it will call upon. It is responsible for transitioning the user between "idle" and "active" menus and facilitates user login. It is the class which handles password changes for existing TigerBank staff.
<u>DEPENDENCIES</u>	SystemMenu has no dependencies of its own. Other classes, such as ClientMenu and AdminMenu have strong associative relationships with SystemMenu because their instances are called within the scope of the SystemMenu class. SystemMenu handles all base functions a branch staff member would, minus admin privileges.
<u>ADDT'L INFO</u>	<u>See class diagram below...(Fig 1.1)</u>

<u>CLASS</u>	<u>AdminMenu</u>
<u>DESCRIPTION</u>	AdminMenu is a friend of SystemMenu and can access SystemMenu's functions and member variables. It does that so the user can change their password from a single menu as opposed to multiple function calls from different classes, possibly duplicating info. AdminMenu is also responsible for allowing the user to add and delete staff from the staff records. AdminMenu contains file I/O functionality to maintain record of all staff members.
<u>DEPENDENCIES</u>	<u>An instance of AdminMenu is used and created within the SystemMenu class.</u>
<u>ADDT'L INFO</u>	<u>See class diagram below...(Fig 1.1)</u>

<b><u>CLASS</u></b>	<b>ClientMenu</b>
<b><u>DESCRIPTION</u></b>	ClientMenu is a friend of SystemMenu and handles all attributes manipulation of Client and Account structures contained within. Furthermore, ClientMenu can add new clients to the bank and modify client data to include depositing and withdrawing client funds. ClientMenu manages file I/O manipulation for client records.
<b><u>DEPENDENCIES</u></b>	ClientMenu is a friend of SystemMenu, thereby inheriting all its functions and member variables. An instance of ClientMenu is contained in SystemMenu.
<b><u>ADDT'L INFO</u></b>	<u>See class diagram below...(Fig 1.1)</u>

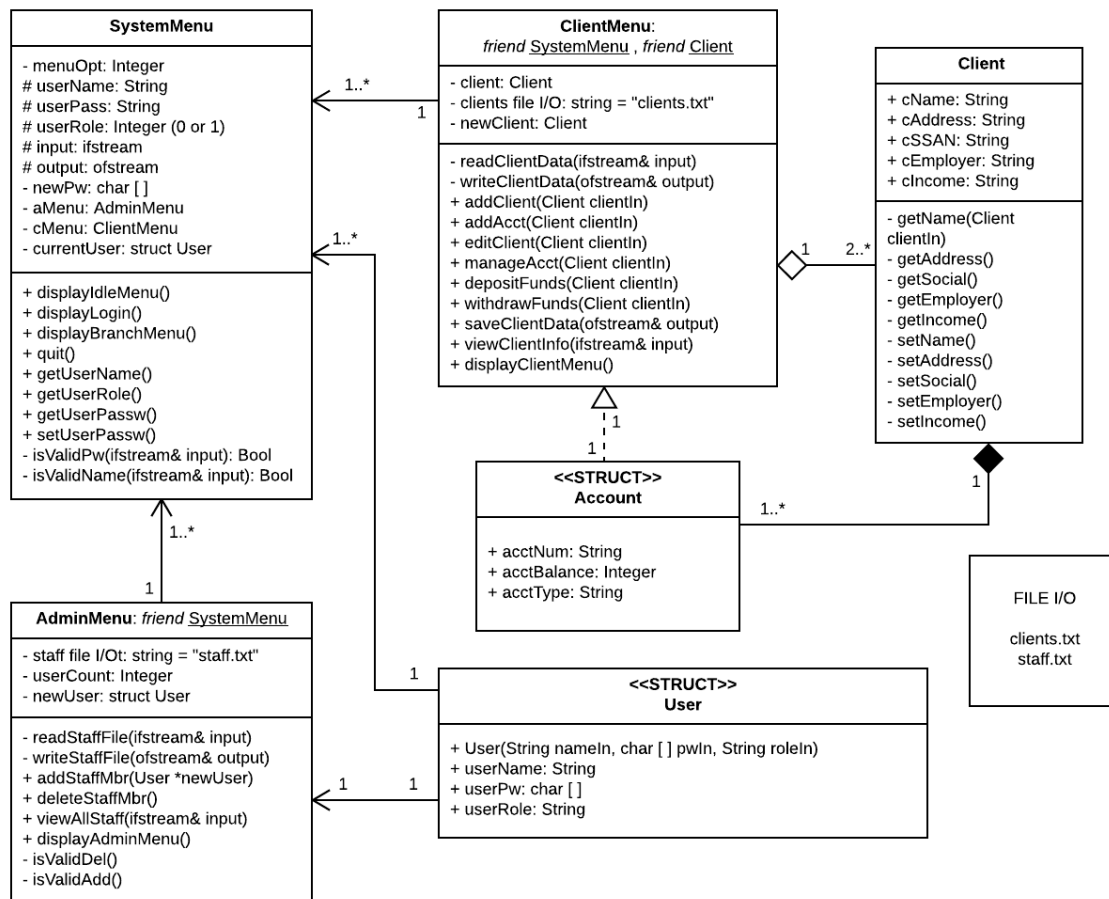
<b><u>&lt;&lt;STRUCT&gt;&gt;</u></b>	<b>User</b>
<b><u>DESCRIPTION</u></b>	User is a struct which represents one of two types of Terminal System end users--System Admin and Branch Staff Member. User has no functionality and is only instantiated to retain user information until such time User's values need to be recorded or retrieved for authentication purposes.
<b><u>DEPENDENCIES</u></b>	An instance of User is created as a newUser and declared in AdminMenu based on the fact AdminMenu <i>can</i> create staff members. A separate instance of User is also declared in SystemMenu to handle all attribute verification for the current teller system user.
<b><u>ADDT'L INFO</u></b>	<u>See class diagram below...(Fig 1.1)</u>

<b><u>CLASS</u></b>	<b>Client</b>
<b><u>DESCRIPTION</u></b>	Client is a class which contains member variables associated with bank clientele. There are two instances of Client in ClientMenu--one for data on current client data being manipulated and the other is for new clients being created.
<b><u>DEPENDENCIES</u></b>	Client is initialized in the ClientMenu class and is used to retain information on clients for attribute manipulation or comparison. Client contains: client name, address, SSAN, employer, and income. It also contains a struct Account since client accounts and information are associated.
<b><u>ADDT'L INFO</u></b>	<u>See class diagram below...(Fig 1.1)</u>



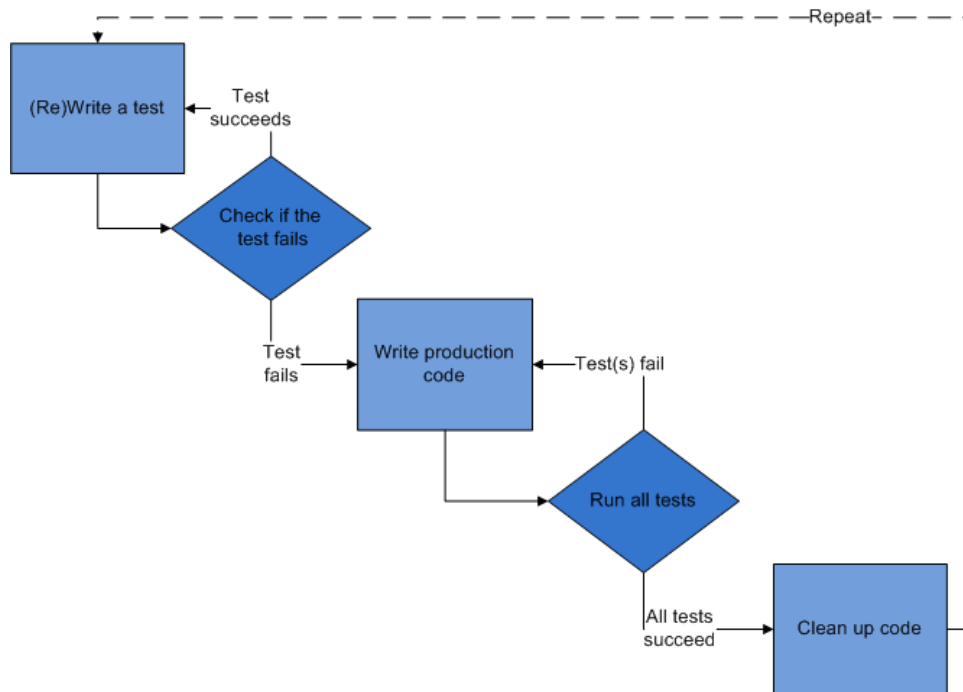
<b>&lt;&lt;STRUCT&gt;&gt;</b>	<b>Account</b>
<b>DESCRIPTION</b>	The struct Account contains information about client accounts and is maintained separate from client data. It includes account number, account balance, and account type.
<b>DEPENDENCIES</b>	Two instances of Account appear in ClientMenu for the purposes of creating new accounts for current new clients. Account data is manipulated and retrieved from the ClientMenu.
<b>ADDT'L INFO</b>	See class diagram below...(Fig 1.1)

Figure 1.1 : UML CLASS DIAGRAM



### (iii) Testing::

## CANDIDATE TEST CASES



Test Case 1	Password Masking
<b>Test Case Summary</b>	This test case will examine password masking which is a vital component of the program during login and password changes.
<b>Test Case Requirements</b> [Requirements needed to run the test suite]	For this test, there will need to be a text file containing branch staff employee data. The user will need to be at the login menu for the teller terminal system (SystemMenu obj created).
<b>Expected Test Result(s)</b>	The expectation is that when a user enters in their password, each character of the password entered in will match an asterisk character.
<b>Optional Details</b>	Potential challenges could come from the use of spaces or indexing out of bounds, and the input string length.  [Environmental considerations such as negative values, loops, index bounds, or other points of failure]

<b>Test Case 2</b>	<b>Valid user name creation</b>
<b>Test Case Summary</b>	Tests system administrator is not entering in a username that pre-exists, during branch staff employee creation.
<b>Test Case Requirements</b> [Requirements needed to run the test suite]	This test requires the user to be at the Administrator Menu from the system menu, and be at the branch staff member creation. It will require access to the staff records (staff.txt) file.
<b>Expected Test Result(s)</b>	The test will make sure that the user enters in only a first and last name, non-alpha numeric characters, pre-existing usernames, and prohibit the use of spaces.
<b>Added Details</b>	<p>This test will need to evaluate special characters and ensure the username is of a specific length. If a c-string is used, the test will need to ensure there is no illegal index used.</p> <p>[Environmental considerations such as negative values, loops, index bounds, or other points of failure]</p>

<b>Test Case 3</b>	<b>Valid password</b>
<b>Test Case Summary</b>	Tests user is not entering in their old password during password change.
<b>Test Case Requirements</b> [Requirements needed to run the test suite]	This test will require the user to be logged in through the system menu and then have selected the option to change their password. The Branch staff member or system administrator will need to enter in the the current user's new password.
<b>Expected Test Result(s)</b>	The expected result is that password masking works for each entered character during user input, and that the staff record is appropriately updated. The result should also take into consideration that the password is NOT the same as the user's previous password.
<b>Added Details</b>	[Environmental considerations such as negative values, loops, index bounds, or other points of failure]

Test Case 4	Test for valid Client Address
<b>Test Case Summary</b>	This test will examine client address is entered in and stored correctly during client creation and because the client's address has multiple fields to be stored during entry.
<b>Test Case Requirements</b> [Requirements needed to run the test suite]	This test requires that the branch staff member or system administrator are logged in and in the client management menu and have selected the option to add a client. This test will also require access to the file storing client information.
<b>Expected Test Result(s)</b>	The expected test result is that client information will be updated with a valid address, and that the address is correctly indexed and written to the client info file.
<b>Added Details</b>	<p>From (i) Analysis::USE CASE 3: Add a Client ...</p> <p><b>(3.1)</b> If the client's zip/postal code is not equal to 5 characters, than user will be prompted to reenter the address with a valid zip/postal code.</p> <p><b>(3.2)</b> If the client's address contains ASCII characters 33, 34, 36 - 43, 47, 58-64, 91 - 96, or 124 - 126, user will be asked to reenter client's address without illegal address characters.</p> <p>Address also must be limited to a valid length with less than 100 characters. The address will also need to be delimited by commas.</p> <p>[Environmental considerations such as negative values, loops, index bounds, or other points of failure]</p>