CST 338
Project 2 (Part 2) – Test Case Development
Due: 11/20/2016 (11:55 PM)

1. For the Project 2, you will develop an Android app for "Book Rental System for CSUMB Library." For the detailed description of the project, see the use case model presented on the next page.

Now, for Part 2 of Project 2, let's assume that you are a TA of this course and will grade the project 2 implementation of other students. To grade the implementation, you should create test cases. Thus, you should **create 15 sample test cases for Project 2**.

For each test case, you should present the title of the test case (such as test case 1, test case 2, test case 3, etc.), purpose of your test case, a sample input scenario, and the expected result for the sample input. The following is a sample test case called **test case 0**. Similarly, you should develop **your own 15 test cases**.

Title: Test Case 0

Purpose: Fail to create a new account with incorrect username/password format.

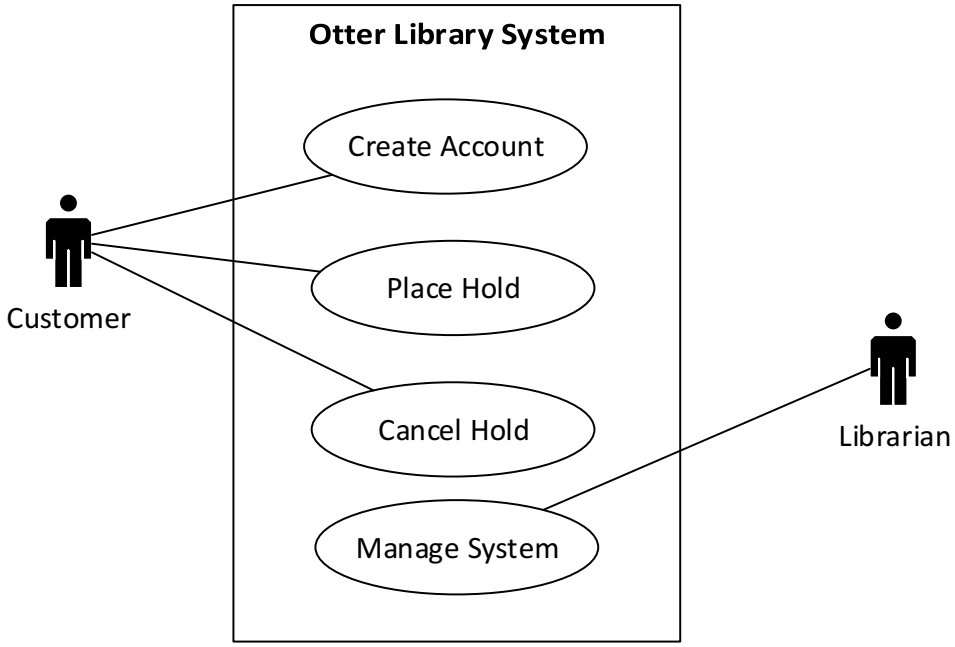
Input Scenario:

1. Selects the option of "Create Account".
2. Type username: byun7
3. Type password: !byun7
4. The System should display an error message for the incorrect format.
5. Type username: byun7!
6. Type password: 7byun

Expected Result:

a) The System should display an error message at the step 4.
b) The System should display an error message after the step 6. After that, it should display the main menu.

Use Case Diagram



Use Case – Create Account

1. Basic Flow

{Display Main Menu}

(1) The use case begins when the System prompts the Customer with options of "Create Account", "Place Hold", "Cancel Hold", and "Manage System".

(2) The Customer selects the "Create Account" option.

{Create Account}

(3) The Customer enters a username and a password.

(4) The System verifies that the username and password are valid. In the project, the username and password should have at least one special symbol (!, @, #, or $), one number, and three alphabetic letters. Note that **!admin2** is reserved for the Librarian. Thus, the Customer can't use it as his/her username. Additionally, we assume that the System already has three usernames as below:

Username	Password
a@lice5	@csit100
$brian7	123abc##
!chris12!	CHRIS12!!

(5) The System informs the Customer that his/her account has been created successfully.

{Log the Transaction}

(6) The System records the transaction information that includes a transaction type (= new account), the Customer's username, and current date/time (= hour and minute).

{Use Case Ends}

(7) The System displays the main menu.
(8) The use case ends.

2. Alternative Flows

2.1 Handle Incorrect Username/Password

At {Create Account}, if the Customer's username and/or password do not meet the required criteria (= at least one special symbol (!, @, #, or $), one number, and three alphabet letters),

1. The System informs the Customer that the username and/or password are not in the correct format.
2. The Customer is prompted to enter new username and password to create an account.
3. If the Customer fails again,
   a. The System displays an error message.
   b. The System displays the main menu.
   c. The use case ends.

2.2 Handle Duplicate Account

At {Create Account}, if the Customer's username already exists in the System,

1. The System informs the Customer that the username already exists.
2. The Customer is prompted to enter new username and password to create an account.
3. If the Customer fails again,
   a. The System displays an error message.
   b. The System displays the main menu.
   c. The use case ends.

Use Case – Place Hold

1. Basic Flow

{Display Main Menu}
(1) The use case begins when the System prompts the Customer with options of "Create Account", "Place Hold", "Cancel Hold", and "Manage System".
(2) The Customer selects the "Place Hold" option.

{Enter Pickup and Return Information}
(3)The System prompts the Customer to enter the date and hour information he/she wants to rent out and return a book.
(4) The Customer picks the date and hour information that he/she wants to pick-up and return the book. In the project, you can't rent a book more than 7 days from the date of pick-up. For example, if the Customer picks up a book on May 22 at 10:00 (AM), the Customer should return it at least by May 29 10:00 (AM).

{Select Book}
(5) Based on the dates and hours selected by the Customer, the System displays all books available for the reservation. In our case, we assume that there are three books in the library as below when the System starts:

Book Title	Author	ISBN	Fee Per Hour
Hot Java	S. Narayanan	123-ABC-101	$0.05
Fun Java	Y. Byun	ABCDEF-09	$1.00
Algorithm for Java	K. Alice	CDE-777-123	$0.25

(6) The Customer picks a book that he/she would like to rent out.

{Login to System}
(7) The System prompts the Customer to enter his/her username and password.
(8) The Customer enters the username and password.
(9) The System confirms that the Customer information is valid.

{Confirmation}
(10) The System displays the Customer username, pickup date/hour, return date/hour, book title, reservation number, and total amount owed.
(11) The System prompts the Customer if the information is correct.
(12) The Customer indicates that the information is correct.

{Log the Transaction}
(13) The System records the transaction information that includes a transaction type (= place hold), the customer username, pickup date/hour, return date/hour, book title, a reservation number, a total amount, and current date/time.

{Use Case Ends}
(14) The System displays the main menu.
(15) The use case ends.

2. Alternative Flows

2.1 Handle Invalid Username or Password

At {Login to System}, if the Customer's login information is not valid,

1. The System informs the Customer that the information entered is not valid.
2. The Customer is prompted to re-enter the username and password.
3. If the Customer fails again,
   a. The System displays an error message.
   b. The System displays the main menu.
   c. The use case ends.

2.2 Handle More than 7 Days Rental

At {Enter Pickup and Return Information}, if the Customer requests more than 7 days rental,

1. The System informs the Customer that the books can't be reserved due to rental restriction.
2. The Customer confirms the error message.
3. The use case returns to {Enter Pickup and Return Information} to prompts again the Customer to enter the date and hour he/she wants to rent out and return a book.

2.3 Handle No Book Available

At {Select Book}, if there's no book available for the pickup and return dates and hours,

1. The System informs the Customer that there is no book available for the dates/hours entered.
2. The System prompts the Customer to exit the program.
3. The Customer clicks the "Exit" and the System displays the main menu.
4. The use case ends.

Use Case – Cancel Hold

1. Basic Flow

{Display Main Menu}
(1) The use case begins when the System prompts the Customer with options of "Create Account", "Place Hold", "Cancel Hold", and "Manage System".
(2) The Customer selects the "Cancel Hold" option.

{Login to System}
(3) The System prompts the Customer to enter his/her username and password.
(4) The Customer enters the username and password.
(5) The System confirms that the Customer information is valid.

{Display Reservations}
(6)The System displays all reservations the Customer made with the reservation number, pickup date/hour, return date/hour, and book title.

{Cancel Hold}
(7) The Customer selects a reservation to cancel.
(8) The System asks the Customer if he/she really wants to cancel the reservation.
(9) The Customer confirms the cancellation.

{Log the Transaction}
(10) The System records the transaction information that includes the Customer's username, book title, pickup date/hour, return date/hour, transaction type (=cancelation), and current date/time.

{Use Case Ends}
(11) The System displays the main menu.
(12) The use case ends.

2. Alternative Flows

2.1 Handle No Reservation
At {Display Reservations}, if the Customer doesn't have any reservation at the moment,
1. The System informs the Customer that there is no reservation with the username.
2. The Customer clicks the OK button to continue.
3. The System displays the main menu.
4. The use case ends.

2.2 Handle Invalid Username or Password
At {Login to System}, if the Customer's login information is not valid,
1. The System informs the Customer that the information entered is not valid.
2. The Customer is prompted to re-enter the username and password.
3. If the Customer fails again,
   a. The System displays an error message.
   b. The System displays the main menu.
   c. The use case ends.

Use Case – Manage System

1. Basic Flow

{Display Main Menu}
(1) The use case begins when the System prompts the Customer with options of "Create Account", "Place Hold", "Cancel Hold", and "Manage System".
(2) The Librarian selects the "Manage System" option.

{Login to System}
(3) The System prompts the Librarian to enter his/her username and password.
(4) The Librarian enters the username and password. In the project, we assume that the Librarian uses **!admin2** as a username and **!admin2** as a password.
(5) The System confirms that the Librarian information is valid.

{Display Log Information}
(6) The System displays all transaction logs.
(7) The Librarian selects the OK button to continue.

{Add New Book Information}
(8) The System asks the Librarian if he/she has a new book to add to the System.
(9) If the Librarian selects the NO option, the use case moves to {Use Case Ends}. However, if the Librarian selects the YES option, the System lets the Librarian enter the book's title, author, ISBN, and fee information.
(10) The System displays the information entered and asks if the information is correct.
(11) The Librarian confirms it.

{Use Case Ends}
(8) The System displays the main menu.
(9) The use case ends.

2. Alternative Flows

2.1 Handle Invalid Username or Password
At {Login to System}, if the Librarian's login information is not valid,

1. The System informs the Librarian that the information entered is not valid.
2. The Librarian is prompted to re-enter the username and password.
3. If the Librarian fails again,
   a. The System displays an error message.
   b. The System displays the main menu.
   c. The use case ends.

2.2 Handle Invalid Book Information
At {Add New Book Information}, if the Librarian missed any information or the book already exists in the Library System,

1. The System informs the Librarian that the information entered is not valid.
2. The Librarian confirms the error message.
3. The System displays the main menu.
4. The use case ends.