**CSCI222 Assignment Test Cases**

**Test Case #1 – Display Login GUI**

a) Startup your program

**Expected Outcome:**

Program should display a login GUI

**< Paste Screen Capture below >**

**Test Case #2a – Testing security functions**

1. User enter wrong username or password

**Expected Outcome:**

Program should display account does not exist

**< Paste Screen Capture below >**

**Test Case #2b – Testing lock account functions**

a) User attempts to log in a specific account three times with wrong passwords.

**Expected Outcome:**

The account with the username user enter should be locked out.

**< Paste Screen Capture below >**

**Test Case #2c – Login and display menu**

a) User log in with correct credentials

**Expected Outcome:**

After successful login program should display menu which should at least show the following

1. ***Add new stock***
2. ***Remove stock***
3. ***Edit stock item***
4. ***Search stock item***
5. ***Daily stock summary report***
6. ***Weekly stock summary report***
7. ***Monthly stock summary report***
8. ***Yearly stock summary report***
9. ***Quit***

**< Paste a screen capture >**

**Test Case #3a – Admin menu**

a) Log in to a admin account

**Expected Outcome :**

Your program should display exactly the following **data** :

Upon successful login to admin account, program should display menu which should have the following

1. ***Add new stock***
2. ***Remove stock***
3. ***Edit stock item***
4. ***Search stock item***
5. ***Daily stock summary report***
6. ***Weekly stock summary report***
7. ***Monthly stock summary report***
8. ***Yearly stock summary report***
9. ***Create Account***
10. ***Delete Account***
11. ***View Account***
12. ***Reset Account***
13. ***Quit***

**< Paste a screen capture >**

**Test Case #4a – View account**

a) Goto to Main Menu, choose k) View Account

**Expected Outcome :**

1. Program should output all data in account files

**< Paste a screen capture >**

**Test Case #4b – Create account (Non existing account)**

a) Goto to Main Menu, choose i) create account

b) User attempts to create account that is not in data file as shown is test case 4a

**< Paste a screen capture (1) – Showing your data entry >**

c) Press enter

**Expected Outcome :**

Program should output “Account successfully created”

**< Paste a screen capture (2) - message and view account >**

**Case #4c – Create account (Existing account)**

a) Goto to Main Menu, choose i) create account

b) user attempts to create an account which already exist in data file

**< Paste a screen capture (1) – Showing your data entry >**

c) Press enter

**Expected Outcome:**

Program should output “account already exist”

**< Paste a screen capture (2) - message and view account summary >**

**Test Case #4d – Delete account (Non existing account)**

a) Goto to Main Menu, choose j) Delete account

b) User attempts to delete an account which exist in data file

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “Account deleted successfully”

**< Paste a screen capture (2) – message and view account >**

**Test Case #4e – delete account (existing account)**

a) Goto to Main Menu, choose j) Delete account

b) user attempts to delete an account which does not exist

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “Account does not exist”

**< Paste a screen capture (2) – Message and view account >**

**Test Case #4f – reset account**

a) Goto to Main Menu, choose l) Reset account

b) Choose the account which was locked out in test case #2b

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “Account reset successfully”

**< Paste a screen capture (2) – Message >**

**Test Case #5a – Daily summary report**

1. Go to Main Menu, choose e) Daily stock summary report

**Expected Outcome:**

Program should output all stock transected on the date based on user input

**< Paste a screen capture – Output of Daily Summary Report >**

**Test Case #5b – Weekly Summary Report**

a) Go to Main Menu, choose f) Weekly stock summary report

**Expected Outcome:**

Program should output all stock transected on the week based on user input

**< Paste a screen capture – Output of Weekly Summary Report >**

**Test Case #5c – Monthly Summary Report**

a) Go to Main Menu, choose g) Monthly stock summary report

**Expected Outcome:**

Program should output all stock transected in the month based on user input

**< Paste a screen capture – Output of Monthly Summary Report >**

**Test Case #5d – Yearly Summary Report**

a) Go to Main Menu, choose h) Yearly stock summary report

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output all stock transected in the year based on user input

**< Paste a screen capture – Output of Yearly Summary Report >**

**Test Case #6a – Add new stock (Non existing stock)**

a) Go to Main Menu, choose a) Add new stock

b) User add in a new stock item which is not present in data file

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

c) Go back to main menu and choose e) Daily summary report

**< Paste a screen capture (2) – Daily summary report >**

**Test Case #6b – Add new stock (existing stock)**

a) Go to Main Menu, choose a) Add new stock

b) User add in a new stock item already present in data file

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “stock item already present in data file”

**< Paste a screen capture (2) – Showing your error message >**

**Test Case #6c – Add new stock (Non existing stock but existing stock ID)**

a) Go to Main Menu, choose a) Add new stock

b) User add in a new stock item with a stock ID already present in data file but item description is not present

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “Stock ID already exist”

**< Paste a screen capture (2) – Showing your error message >**

**Test Case #6d – Add new stock (invalid amount per unit)**

a) Go to Main Menu, choose a) Add new stock

b) User add new stock but does not enter numbers for amount per unit

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome :**

**< Paste a screen capture (2) – Message >**

**Test Case #6e – Add new stock (invalid date format)**

a) Go to Main Menu, choose a) Add new stock

b) User add new stock but for transacted date user enter wrong date format (dd-mmm-yy)

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome :**

Program should output “wrong date format”.

**< Paste a screen capture (2) – Message >**

**Test Case #6f – Add new stock (Invalid quantity)**

a) Go to Main Menu, choose a) Add new stock

b) User add new stock but for quantity enter decimal numbers

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome :**

Program should output “invalid input”

**< Paste a screen capture (2) – Message >**

**Test Case #7a – Remove stock (Non existing stock)**

a) Go to Main Menu, choose b) Remove stock

b) User enter a stock ID which is present in data file

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome :**

Program should output “Stock with all “ID” enter is deleted”

**< Paste a screen capture (2) –message >**

**Test Case #7b – Remove stock (existing stock)**

a) Go to Main Menu, choose b) Remove stock

b) User enter a stock ID which is not present in data file

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “No stock with “ID” is found”

**< Paste a screen capture (2) –message >**

**Test Case #8a – Search stock (existing item ID)**

a) Go to Main Menu, choose d) Search stock item

b) User enter existing item id

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output following stock item with that correspond to the item ID user enter

**< Paste a screen capture (2) – message>**

**Test Case #8b – Search stock (Non existing item ID)**

a) Go to Main Menu, choose d) Search stock item

b) User enter non existing item id

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “No stock with this ID is found”

**< Paste a screen capture (2) – message>**

**Test Case #8c – Search stock (existing item description)**

a) Go to Main Menu, choose d) Search stock item

b) User enter non existing item description

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output following stock item with that correspond to the item description user enter

**< Paste a screen capture (2) – message>**

**Test Case #8d – Search stock (non-existing item description)**

a) Go to Main Menu, choose d) Search stock item

b) User enter non existing item description

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “No stock with that description is found”

**< Paste a screen capture (2) – message>**

**Test Case #8e – Search stock (existing item category)**

a) Go to Main Menu, choose d) Search stock item

b) User enter existing item category

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output following stock item with that correspond to the item category user enter

**< Paste a screen capture (2) – message>**

**Test Case #8f – Search stock (non-existing item category)**

a) Go to Main Menu, choose d) Search stock item

b) User enter non existing item category

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “No stock with that category is found”

**< Paste a screen capture (2) – message>**

**Test Case #8g – Search stock (existing item sub-category)**

a) Go to Main Menu, choose d) Search stock item

b) User enter existing item sub-category

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output following stock item with that correspond to the item sub-category user enter

**< Paste a screen capture (2) – message>**

**Test Case #8h – Search stock (non-existing item sub-category)**

a) Go to Main Menu, choose d) Search stock item

b) User enter non existing item sub-category

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “No stock with that sub-category is found”

**< Paste a screen capture (2) – message>**

**Test Case #9a – edit stock (Non existing stock)**

a) Go to Main Menu, choose c) edit stock item

b) User enter stock ID of which user wishes to update on

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “No stock with this ID found”

**< Paste a screen capture (2) – message>**

**Test Case #9b – edit stock incoming (Existing Stock)**

a) Go to Main Menu, choose c) edit stock item

b) User enters stock ID that exists in data file and updates incoming stock and enters the date of transection.

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “Stock successfully updated”

**< Paste a screen capture (2) – message>**

c) User proceeds to main menu and choose e) Daily summary report

d) User enter the date of stock transection

**< Paste a screen capture (3) – Daily summary report>**

**Test Case #9c – edit stock outgoing (Existing Stock)**

a) Go to Main Menu, choose c) edit stock item

b) User enters stock ID that exists in data file and updates outgoing stock and enters the date of transection.

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “Stock successfully updated”

**< Paste a screen capture (2) – message>**

c) User proceeds to main menu and choose e) Daily summary report

d) User enter the date of stock transection

**< Paste a screen capture (3) – Daily summary report>**

**Test Case #9c – edit stock outgoing (invalid input)**

a) Go to Main Menu, choose c) edit stock item

b) User enters stock ID that exists in data file and updates outgoing stock but not in correct format (example with decimal points)

**< Paste a screen capture (1) – Showing your data entry >**

**Expected Outcome:**

Program should output “invalid input”

**< Paste a screen capture (2) – message>**