CSIT314 Project Test Plan & Test Cases

[Document subtitle]

DAMIAN YEO KUN HAN

Contents

[Test Plan Introduction 3](#_Toc102471550)

[Test Strategy 3](#_Toc102471551)

[Test Objectives 3](#_Toc102471552)

[Test Assumptions 3](#_Toc102471553)

[Scope of Testing 3](#_Toc102471554)

[Front End Testing 3](#_Toc102471555)

[Unit Testing 3](#_Toc102471556)

[Defect Severity Categorisation 4](#_Toc102471557)

[Front End Testing for User Stories 5](#_Toc102471558)

[Sprint 1 5](#_Toc102471559)

[User Story #1 5](#_Toc102471560)

[User Story #2 6](#_Toc102471561)

[User Story #13 7](#_Toc102471562)

[User Story #14 8](#_Toc102471563)

[User Story #20 9](#_Toc102471564)

[User Story #21 10](#_Toc102471565)

[User Story #35 11](#_Toc102471566)

[User Story #36 12](#_Toc102471567)

[Sprint 2 13](#_Toc102471568)

[User Story #3 and #8 13](#_Toc102471569)

[User Story #4 and #9 14](#_Toc102471570)

[User Story #5 and #10 15](#_Toc102471571)

[User Story #6 and #11 16](#_Toc102471572)

[User Story #7 and #12 17](#_Toc102471573)

[User Story #27 18](#_Toc102471574)

[User Story #28 19](#_Toc102471575)

[User Story #29 20](#_Toc102471576)

[User Story #30 21](#_Toc102471577)

[User Story #31 22](#_Toc102471578)

[User Story #32 23](#_Toc102471579)

[User Story #33 24](#_Toc102471580)

[User Story #34 25](#_Toc102471581)

[Staff class 26](#_Toc102471582)

[User\_Admin class 27](#_Toc102471583)

# Test Plan Introduction

The objective of this test plan is to show how testing will be done for the Hopium program.

# Test Strategy

## Test Objectives

The objective of testing is to ensure that the program works as intended and that there are no critical errors that will affect the end user.

## Test Assumptions

Key Assumptions

* Testing Data will include a small amount of data for initial testing and a larger production like amount of data for final testing
* Performance will not be considered for testing
* Any defects will be reported to the developer team to fix

## Scope of Testing

### Front End Testing

**Purpose:** Front end testing will focus on what the end user will see and experience as they use the program.

**Scope:** Testing will be mainly focused on GUI elements

**Method:** Testing will be done manually on the program’s GUI based on what the user will do for each user story. Results will be compared to expected results and any bugs will be reported to the programming team

**Frequency of Tests:** These tests will be done whenever a new feature is implemented into the program to test the new functionality and to make sure it doesn’t not break existing code.

### Unit Testing

**Purpose:** Testing will be performed to check the functions of the program. Testing is done by feeding the functions inputs and checking the output of the functions.

**Scope:** Testing will mainly be done on the Controller and Entity classes to check if they can retrieve the correct information from the database.

**Method:** Tests will be mainly performed using Junit 5 scripts to run automated tests on a per class basis. Any errors will be reported to the programming team

**Frequency of Tests:** These tests can be run more often than Front End Testing, so this test can be run whenever code is adjusted or new functionality is added.

# Defect Severity Categorisation

Defects will be given a severity level to show the severity of the defect and the priority in fixing the defect.

|  |  |  |
| --- | --- | --- |
| Severity | Priority | Impact |
| Critical | 1 | Defect that can crash the program |
| High | 2 | Defect that disables a function of the program |
| Medium | 3 | Defect that disables a function of the program but has a workaround |
| Low | 4 | Defect that has minimal impact on functionality or purely cosmetic |

# Front End Testing for User Stories

## Sprint 1

### User Story #1

As a User Admin, I want to login so that I can manage user accounts

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 1.1 | Login with valid User Admin credentials | Test if user admin can login with valid credentials |
| 1.2 | Login with invalid User Admin credentials | Test if user admin can login without valid credentials |
| 1.3 | Login with valid User Admin credentials but wrong profile | Test if user admin can login if choosing the wrong profile |

|  |  |
| --- | --- |
| **Test Case 1.1 – Login with valid User Admin credentials** | |
| **Steps** | 1. Start application 2. Click ‘Staff’ button 3. Type in valid credentials (user\_admin1 for both fields) 4. Select ‘User Admin’ in profile dropdown menu 5. Click ‘Login’ button |
| **Expected Results** | Logged in and User Admin homepage displayed |

|  |  |
| --- | --- |
| **Test Case 1.2 – Login with invalid User Admin credentials** | |
| **Steps** | 1. Start application 2. Click ‘Staff’ button 3. Type in invalid credentials (user\_admin0 for both fields) 4. Select ‘User Admin’ in profile dropdown menu 5. Click ‘Login’ button |
| **Expected Results** | Failed Login |

|  |  |
| --- | --- |
| **Test Case 1.3 – Login with valid User Admin credentials but wrong profile** | |
| **Steps** | 1. Start application 2. Click ‘Staff’ button 3. Type in valid credentials (user\_admin1 for both fields) 4. Select wrong profile in the profile dropdown menu (Restaurant Manager selected) 5. Click ‘Login’ button |
| **Expected Results** | Failed Login |

### User Story #2

As a User Admin, I want to logout so that other people can login and for security purposes

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 2.1 | Log out from the User Admin account | Test if the user admin can log out of the account |

|  |  |
| --- | --- |
| **Test Case 2.1 – Log out from the User Admin account** | |
| **Steps** | 1. While in the User Admin homepage, click the log out button |
| **Expected Results** | Logged out and returned to the login page |

### User Story #13

As a Restaurant Manager, I want to login so that I can create/modify menu items, coupons, etc.

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 13.1 | Login with valid Restaurant Manager credentials | Test if restaurant manager can login with valid credentials |
| 13.2 | Login with invalid Restaurant Manager credentials | Test if restaurant manager can login without valid credentials |
| 13.3 | Login with valid Restaurant Manager credentials but wrong profile | Test if restaurant manager can login if choosing the wrong profile |

|  |  |
| --- | --- |
| **Test Case 13.1 – Login with valid Restaurant Manager credentials** | |
| **Steps** | 1. Start Application 2. Click 'Staff' button 3. Type in valid username and password (manager1 for both fields) 4. Select Restaurant Manager in the profile login dropdown menu 5. Click 'Login' button |
| **Expected Results** | Logged in and Restaurant Manager homepage displayed |

|  |  |
| --- | --- |
| **Test Case 13.2 – Login with invalid Restaurant Manager credentials** | |
| **Steps** | 1. Start Application 2. Click 'Staff' button 3. Type in invalid username and password (manager0 for both fields) 4. Select Restaurant Manager in the profile login dropdown menu 5. Click 'Login' button |
| **Expected Results** | Failed Login |

|  |  |
| --- | --- |
| **Test Case 13.3 – Login with valid Restaurant Manager credentials but wrong profile** | |
| **Steps** | 1. Start Application 2. Click 'Staff' button 3. Type in valid username and password (manager1 for both fields) 4. Select wrong profile in the profile login dropdown menu (User Admin selected) 5. Click 'Login' button |
| **Expected Results** | Failed Login |

### User Story #14

As a Restaurant Manager, I want to logout so that other people can login and for security purposes

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 14.1 | Log out from the Restaurant Manager account | Test if the restaurant manager can log out of the account |

|  |  |
| --- | --- |
| **Test Case 14.1 – Log out from the Restaurant Manager account** | |
| **Steps** | 1. While in the Restaurant Manager homepage, click the log out button |
| **Expected Results** | Logged out and returned to the login page |

### User Story #20

As a Restaurant Staff, I want to login so that I can manage and fulfil orders

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 20.1 | Login with valid Restaurant Staff credentials | Test if restaurant staff can login with valid credentials |
| 20.2 | Login with invalid Restaurant Staff credentials | Test if restaurant staff can login without valid credentials |
| 20.3 | Login with valid Restaurant Staff credentials but wrong profile | Test if restaurant staff can login if choosing the wrong profile |

|  |  |
| --- | --- |
| **Test Case 20.1 – Login with valid Restaurant Staff credentials** | |
| **Steps** | 1. Start application 2. Click ‘Staff’ button 3. Type in valid credentials (user1 for both fields) 4. Select ‘Restaurant Staff’ in profile dropdown menu 5. Click ‘Login’ button |
| **Expected Results** | Logged in and Restaurant Staff homepage displayed |

|  |  |
| --- | --- |
| **Test Case 20.2 – Login with invalid Restaurant Staff credentials** | |
| **Steps** | 1. Start application 2. Click ‘Staff’ button 3. Type in invalid credentials (user0 for both fields) 4. Select ‘Restaurant Staff’ in profile dropdown menu 5. Click ‘Login’ button |
| **Expected Results** | Failed Login |

|  |  |
| --- | --- |
| **Test Case 20.3 – Login with valid Restaurant Staff credentials but wrong profile** | |
| **Steps** | 1. Start application 2. Click ‘Staff’ button 3. Type in valid credentials (user1 for both fields) 4. Select wrong profile in the profile dropdown menu (User Admin selected) 5. Click ‘Login’ button |
| **Expected Results** | Failed Login |

### User Story #21

As a Restaurant Staff, I want to logout so that other people can login and for security purposes

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 21.1 | Log out from the Restaurant Staff account | Test if the restaurant staff can log out of the account |

|  |  |
| --- | --- |
| **Test Case 21.1 – Log out from the Restaurant Staff account** | |
| **Steps** | 1. While in the Restaurant Staff homepage, click the log out button |
| **Expected Results** | Logged out and returned to the login page |

### User Story #35

As a Restaurant Owner, I want to login to my account, so that I can view and manage the data.

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 35.1 | Login with valid Restaurant Owner credentials | Test if restaurant owner can login with valid credentials |
| 35.2 | Login with invalid Restaurant Owner credentials | Test if restaurant owner can login without valid credentials |
| 35.3 | Login with valid Restaurant Owner credentials but wrong profile | Test if restaurant owner can login if choosing the wrong profile |

|  |  |
| --- | --- |
| **Test Case 35.1 – Login with valid credentials** | |
| **Steps** | 1. Start application 2. Click ‘Staff’ button 3. Type in valid credentials (owner1 for both fields) 4. Select ‘Restaurant Owner’ in profile dropdown menu 5. Click ‘Login’ button |
| **Expected Results** | Logged in and Restaurant Owner homepage displayed |

|  |  |
| --- | --- |
| **Test Case 35.2 – Login with invalid credentials** | |
| **Steps** | 1. Start application 2. Click ‘Staff’ button 3. Type in invalid credentials (owner0 for both fields) 4. Select ‘Restaurant Owner’ in profile dropdown menu 5. Click ‘Login’ button |
| **Expected Results** | Failed Login |

|  |  |
| --- | --- |
| **Test Case 35.3 – Login with valid credentials but wrong profile** | |
| **Steps** | 1. Start application 2. Click ‘Staff’ button 3. Type in valid credentials (owner1 for both fields) 4. Select wrong profile in the profile dropdown menu (User Admin selected) 5. Click ‘Login’ button |
| **Expected Results** | Failed Login |

### User Story #36

As a Restaurant Owner, I want to logout of my account, so that other people can login and for security purposes.

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 36.1 | Log out from the Restaurant Owner account | Test if the restaurant owner can log out of the account |

|  |  |
| --- | --- |
| **Test Case 36.1 – Log out from the Restaurant Owner account** | |
| **Steps** | 1. While in the Restaurant Owner homepage, click the log out button |
| **Expected Results** | Logged out and returned to the login page |

## Sprint 2

### User Story #3 and #8

As a User Admin, I want to create a user account so that I can create an account for a new employee As a User Admin, I want to create a user profile so that I can create a profile for a new employee

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 3.1 | Create with new credentials | Test if the program can create a new account |
| 3.2 | Create with existing credentials | Test if the program can create a duplicate account |
| 3.3 | Create with missing credentials | Test if the program will warn the user of empty fields |

|  |  |
| --- | --- |
| **Test Case 3.1 – Create with new credentials** | |
| **Steps** | 1. Login to User Admin account 2. Click the ‘Create’ button 3. Type in new user account details 4. Click the ‘Create Account’ button |
| **Expected Results** | The program should show a window telling the user a new account is created |

|  |  |
| --- | --- |
| **Test Case 3.2 – Create with existing credentials** | |
| **Steps** | 1. Login to User Admin account 2. Click the ‘Create’ button 3. Type in duplicate user account details 4. Click the ‘Create Account’ button |
| **Expected Results** | The program should show a window telling the user that the account already exists |

|  |  |
| --- | --- |
| **Test Case 3.3 – Create with missing credentials** | |
| **Steps** | 1. Login to User Admin account 2. Click the ‘Create’ button 3. Type only username or password or neither 4. Click the ‘Create Account’ button |
| **Expected Results** | The program should show a window telling the user that fields should not be left empty |

### User Story #4 and #9

As a User Admin, I want to edit user accounts, so that I can change and update the user account information

As a User Admin, I want to edit user profiles, so that I can change and update the user profile information

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 4.1 | Edit User Account | Test if the program can edit a user’s account details |
| 4.2 | Edit without selecting user | Test if the program can warn that no user is selected |
| 4.3 | Edit with empty fields | Test if the program can warn that there are empty fields |

|  |  |
| --- | --- |
| **Test Case 4.1 – Edit User Account** | |
| **Steps** | 1. Login to User Admin account 2. Click ‘Edit’ button 3. Select a User Account 4. Modify the details in the fields 5. Click ‘Edit Changes’ button |
| **Expected Results** | The program should show a window telling the user that the edits are saved |

|  |  |
| --- | --- |
| **Test Case 4.2 – Edit details without selecting user** | |
| **Steps** | 1. Login to User Admin account 2. Click ‘Edit’ button 3. Modify the details without selecting user 4. Click ‘Edit Changes’ button |
| **Expected Results** | The program should warn the user that the account does not exist |

|  |  |
| --- | --- |
| **Test Case 4.3 – Edit with empty fields** | |
| **Steps** | 1. Login to User Admin account 2. Click ‘Edit’ button 3. Select a User Account 4. Clear the username, password, user profile fields 5. Click ‘Edit Changes’ button |
| **Expected Results** | The program should warn the user that there are empty fields |

### User Story #5 and #10

As a User Admin, I want to be able to search for user accounts so that I can look for specific user accounts

As a User Admin, I want to be able to search for user profiles so that I can look for specific user profiles

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 5.1 | Search for existing user by Username | Test if the program can search for an existing user by username |
| 5.2 | Search for non-existent user by Username | Test if the program can tell the admin the user does not exist by username |
| 5.3 | Search for existing user by User Profile | Test if the program can search for an existing user by user profile |
| 5.4 | Search for non-existent user by User Profile | Test if the program can tell the admin the user does not exist by user profile |

|  |  |
| --- | --- |
| **Test Case 5.1 – Search for existing user by Username** | |
| **Steps** | 1. Login to User Admin account 2. Click ‘View’ button 3. Type an existing username 4. Click ‘Search by Username’ button |
| **Expected Results** | The program should show the list of user accounts that match the criteria |

|  |  |
| --- | --- |
| **Test Case 5.2 – Search for non-existent user by Username** | |
| **Steps** | 1. Login to User Admin account 2. Click ‘View’ button 3. Type a non-existing username 4. Click ‘Search by Username’ button |
| **Expected Results** | The program should show an empty list |

|  |  |
| --- | --- |
| **Test Case 5.3 – Search for existing user by Username** | |
| **Steps** | 1. Login to User Admin account 2. Click ‘View’ button 3. Type an existing username 4. Click ‘Search by User Profile’ button |
| **Expected Results** | The program should show the list of user accounts that match the criteria |

|  |  |
| --- | --- |
| **Test Case 5.4 – Search for non-existent user by Username** | |
| **Steps** | 1. Login to User Admin account 2. Click ‘View’ button 3. Type a non-existing username 4. Click ‘Search by User Profile’ button |
| **Expected Results** | The program should show an empty list |

### User Story #6 and #11

As a User Admin, I want to view user accounts so that I can retrieve user account information

As a User Admin, I want to view user profiles so that I can retrieve user profile information

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 6.1 | View user account information | Test if the program can display user account details |

|  |  |
| --- | --- |
| **Test Case 6.1 – View all accounts** | |
| **Steps** | 1. Login to User Admin account 2. Click ‘View’ button |
| **Expected Results** | The program should show the list of user accounts |

### User Story #7 and #12

As a User Admin, I want to be able to suspend user accounts, so that I can suspend their user account if they leave the company

As a User Admin, I want to be able to suspend user profiles, so that I can suspend their user profile if they leave the company

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 7.1 | Suspend user | Test if the program can suspend an active user |
| 7.2 | Unsuspend user | Test if the program can un-suspend an inactive user |

|  |  |
| --- | --- |
| **Test Case 7.1 – Suspend user** | |
| **Steps** | 1. Login to User Admin account 2. Click ‘Suspend’ button 3. Select an active user 4. Click ‘Suspend/Un-suspend Account’ button |
| **Expected Results** | The program should show that the user account is suspended |

|  |  |
| --- | --- |
| **Test Case 7.2 – Un-suspend user** | |
| **Steps** | 1. Login to User Admin account 2. Click ‘Suspend’ button 3. Select an inactive user 4. Click ‘Suspend/Un-suspend Account’ button |
| **Expected Results** | The program should show that the user account is un-suspended |

|  |  |
| --- | --- |
| **Test Case 7.3 – Suspend/Un-suspend without selecting user** | |
| **Steps** | 1. Login to User Admin account 2. Click ‘Suspend’ button 3. Click ‘Suspend/Un-suspend Account’ button |
| **Expected Results** | The program should tell the user that no user is selected |

### User Story #27

As a Customer, I want to Insert Table Number, so that I can start placing order under my table number for identification

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 27.1 | Insert table number | Test if the program will allow the customer to start placing orders |
| 27.2 | Leave table field empty | Test if the program will warn the user that the table number field is empty |

|  |  |
| --- | --- |
| **Test Case 27.1 – Insert table number** | |
| **Steps** | 1. At the main page, click the ‘Customer’ button 2. Enter the table number in the table number field 3. Click ‘View Menu’ |
| **Expected Results** | The program should open the customer home page |

|  |  |
| --- | --- |
| **Test Case 27.1 – Insert table number** | |
| **Steps** | 1. At the main page, click the ‘Customer’ button 2. Leave the table number field empty 3. Click ‘View Menu’ |
| **Expected Results** | The program should warn the user that no table number is entered. |

### User Story #28

As a Customer, I want to browse the menu, so that I can choose what menu items I want

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 28.1 | View Menu | Test if the program will allow the customer to view the menu |

|  |  |
| --- | --- |
| **Test Case 27.1 – View Menu** | |
| **Steps** | 1. At the Customer Homepage, click ‘View Menu’ |
| **Expected Results** | The program should display the menu |

### User Story #29

As a Customer, I want to add menu items to cart, so that I can order menu items

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 29.1 | Add menu item to cart | Test if the program will allow the customer to add items to cart |

|  |  |
| --- | --- |
| **Test Case 29.1 – Add menu item to cart** | |
| **Steps** | 1. At the Customer Homepage, click ‘View Menu’ 2. Select menu item 3. Change the quantity to desired amount 4. Click ‘Add To Cart’ |
| **Expected Results** | The program should display a message telling the customer the item is added to cart |

### User Story #30

As a Customer, I want to edit cart items, so that I can change the quantity of the items in my cart

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 30.1 | Edit cart item quantity | Test if the program will allow the customer to edit the cart item quantity |

|  |  |
| --- | --- |
| **Test Case 30.1 – Remove menu item from cart** | |
| **Steps** | 1. At the Customer Homepage, click ‘Edit Cart’ 2. Select menu item 3. Type in new item quantity 4. Click ‘Edit’ |
| **Expected Results** | The program should display a message telling the customer the item quantity is updated |

### User Story #31

As a Customer, I want to search different categories of menu items (main course, drinks etc), so that I can easily find the type of food I wish to order.

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 31.1 | Categorize menu items by main course | Test if the program will show only main course items |
| 31.2 | Categorize menu items by drinks | Test if the program will show only drink items |

|  |  |
| --- | --- |
| **Test Case 31.1 – Categorize menu items by main course** | |
| **Steps** | 1. At the view menu, click the ‘Main Course’ button |
| **Expected Results** | The program should display only main course items |

|  |  |
| --- | --- |
| **Test Case 31.2 – Categorize menu items by drinks** | |
| **Steps** | 1. At the view menu, click the ‘Drinks’ button |
| **Expected Results** | The program should display only drink items |

### 

### User Story #32

As a Customer, I want to view cart items, so that I can verify the items that I want to order

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 32.1 | View cart items | Test if the program will allow the customer to view the cart |

|  |  |
| --- | --- |
| **Test Case 32.1 – View cart items** | |
| **Steps** | 1. At the Customer Homepage, click ‘View Cart’ |
| **Expected Results** | The program should display a list of all items in the cart |

### User Story #33

As a Customer, I want to delete cart items, so that I can remove items from cart which I have incorrectly added/do not want anymore

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 33.1 | Delete cart items | Test if the program will allow the customer to delete cart items |

|  |  |
| --- | --- |
| **Test Case 33.1 – Delete cart items** | |
| **Steps** | 1. At the Customer Homepage, click ‘View Cart’ |
| **Expected Results** | The program should display a list of all items in the cart |

### User Story #34

As a Customer, I want to do e-payment, so that I can make payment from the table

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 34.1 | Perform e-payment | Test if the program will allow the customer make e-payment |

|  |  |
| --- | --- |
| **Test Case 34.1 – Delete cart items** | |
| **Steps** | 1. At the Customer Homepage, click ‘Payment’ 2. Enter phone number/email 3. Click ‘Make Payment’ |
| **Expected Results** | The program should show the user a message that payment has been completed |

Unit Testing for Classes

## Staff class

The Staff Class is responsible for communicating with the database and retrieving the relevant information from the database whenever the program requires it.

Junit Test file name : StaffTest.java

**List of Unit Test cases for Staff class (Updated 9/5/22)**

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Test Description | Tested Function | Expected Result |
| Staff\_1 | Login with valid User Admin credentials | login() | True |
| Staff\_2 | Login with invalid User Admin credentials | login () | False |
| Staff\_3 | Login with valid User Admin but wrong profile | login () | False |
|  |  |  |  |
| Staff\_4 | Login with valid Manager credentials | login () | True |
| Staff\_5 | Login with invalid Manager credentials | login () | False |
| Staff\_6 | Login with valid Manager but wrong profile | login () | False |
|  |  |  |  |
| Staff\_7 | Login with valid Staff credentials | login () | True |
| Staff\_8 | Login with invalid Staff credentials | login () | False |
| Staff\_9 | Login with valid Staff but wrong profile | login () | False |
|  |  |  |  |
| Staff\_10 | Login with valid Owner credentials | login () | True |
| Staff\_11 | Login with invalid Owner credentials | login () | False |
| Staff\_12 | Login with valid Owner but wrong profile | login () | False |

## User\_Admin class

The User\_Admin class is responsible for User Admin functions such as creating and editing accounts and retrieving user info for viewing.

Junit Test file name : User\_AdminTest.java

**List of Unit Test cases for User\_Admin class (Updated 9/5/22)**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Case | Tested Function | Expected Result |
| User\_Admin\_1 | Create new account | createAccount() | No exceptions |
| User\_Admin\_2 | Edit an account | editAccount() | No exceptions |
| User\_Admin\_3 | Get information from database | getUserInfoFromDB() | No exceptions |
| User\_Admin\_4 | Search for accounts | searchAccount() | No exceptions |
| User\_Admin\_5 | Suspend an account | suspendAccount() | No exceptions |

## Customer class

9/5/22 – For now all functions in customer class are just calling the Cart class functions. Therefore, testing the Cart class is sufficient.

## Cart class

The Cart class is responsible for Cart functions such as adding and removing items from the customer’s cart.

Junit Test file name : CartTest.java

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Case | Tested Function | Expected Result |
| Cart\_1 | Get transaction id from cart object | getTransaction\_idTest() | No exceptions |
| Cart\_2 | Generate transaction id for table number | setTransaction\_idTest() | No exceptions |
| Cart\_3 | View items in cart | viewCart() | No exceptions |
| Cart\_4 | Add item to cart | addToCart() | No exceptions |
| Cart\_5 | Edit quantity in cart | editCart() | No exceptions |
| Cart\_6 | Delete item from cart | deleteFromCart() | No exceptions |
| Cart\_7 | Get total price of cart | getTotalPrice() | Float Value |
| Cart\_8 | Apply discount on price | applyDiscount() | True |
| Cart\_9 | Make payment | makePayment() | True |

## Restaurant\_Owner class

The Restaurant Owner class is responsible for functions such as generating daily, weekly and monthly reports.

Junit Test file name : Restaurant\_OwnerTest.java

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Case | Tested Function | Expected Result |
| Restaurant\_Owner\_1 | Get daily spending report | dailySpending() | String |
| Restaurant\_Owner\_2 | Get weekly spending report | weeklySpending() | String |
| Restaurant\_Owner\_3 | Get monthly spending report | monthlySpending() | String |
| Restaurant\_Owner\_4 | Get daily frequency report | dailyFrequency() | String |
| Restaurant\_Owner\_5 | Get weekly frequency report | weeklyFrequency() | String |
| Restaurant\_Owner\_6 | Get monthly frequency report | monthlyFrequency() | String |
| Restaurant\_Owner\_7 | Get daily preference report | dailyPreference() | String |
| Restaurant\_Owner\_8 | Get weekly preference report | weeklyPreference() | String |
| Restaurant\_Owner\_9 | Get monthly preference report | monthlyPreference() | String |
| Restaurant\_Owner\_10 | Get all reports in a 2D array | getReport() | String [] [] |

## Menu\_Item class

The Menu Item class is responsible for displaying the menu to the customer

Junit Test file name : Menu\_ItemsTest.java

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Case | Tested Function | Expected Result |
| Menu\_Item\_1 | Get all menu items | viewMenu() | String [] [] |
| Menu\_Item\_2 | Get pasta menu items | viewMenu() | String [] [] |
| Menu\_Item\_3 | Get pizza menu items | viewMenu() | String [] [] |
| Menu\_Item\_4 | Get baked rice menu items | viewMenu() | String [] [] |
| Menu\_Item\_5 | Get drink menu items | viewMenu() | String [] [] |

## Restaurant\_Manager class

The restaurant manager class is responsible for functions such as creating, editing, searching and deleting menu items/coupons

Junit Test file name : Restaurant\_ManagerTest.java

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Case | Tested Function | Expected Result |
| Restaurant\_Manager\_1 | Get transaction id from cart object | checkMenuItemExistence() | True |
| Restaurant\_Manager\_2 | Generate transaction id for table number | createMenuItems() | True |
| Restaurant\_Manager\_3 | View items in cart | checkCouponExistence() | True |
| Restaurant\_Manager\_4 | Add item to cart | createCoupon() | True |
| Restaurant\_Manager\_5 | Edit quantity in cart | edditMenuItems() | True |
| Restaurant\_Manager\_6 | Delete item from cart | searchMenuItems() | True |
| Restaurant\_Manager\_7 | Get total price of cart | viewMenuItems() | True |
| Restaurant\_Manager\_8 | Apply discount on price | deleteMenuItems() | True |

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Case** | **Purpose** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

|  |  |
| --- | --- |
| **Test Case** | |
| **Steps** |  |
| **Expected Results** |  |

|  |  |  |
| --- | --- | --- |
| ID | Test Case | Tested Function |
|  |  |  |

|  |  |
| --- | --- |
| **Test Case** | |
| **Expected Results** |  |
| **Actual Results** |  |
| **Pass/Fail** |  |