

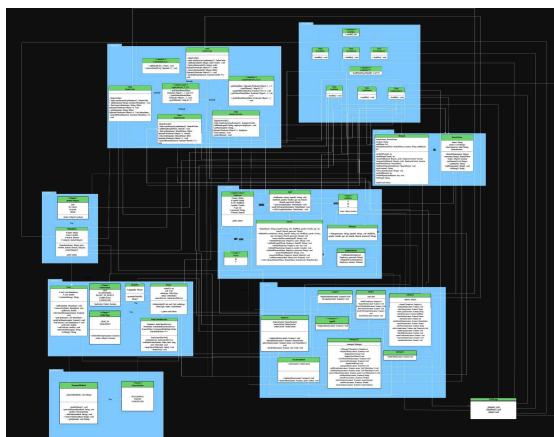
We hereby declare that the attached group assignment has been researched, undertaken, completed, and submitted as a collective effort by the group members listed below. We have honoured the principles of academic integrity and have upheld Student Code of Academic Conduct in the completion of this work. We understand that if plagiarism is found in the assignment, then lower marks or no marks will be awarded for the assessed work. In addition, disciplinary actions may be taken.

Name	Course	Lab Group	Signature/Date
Michael Yip Hoi Weng	DSAI	FDAC	Michael/ 25/04/2024
Muhammad Aidil Firdaus	DSAI	FDAC	Aidil/ 24/04/2024
Nathan Pua Jun Shen	DSAI	FDAC	Nathan/ 24/04/2024
Tay Kai Wen	DSAI	FDAC	Kai Wen/ 24/04/2024
Tan Yi Da	BS	FDAC	Yi Da/ 24/04/2024

## 1. Design Considerations

### 1.1 Approach

Our FOMSApp system implements a fast food management system for 4 users: Staff, Manager, Admin and Customer. The system uses polymorphism for the roles as there is some overlap in user options among the 4 users. A complex cache system is also implemented to store the data input by all the users (e.g. OrderCache, BranchCache etc.), and the memory is stored on system shutdown. A filtering system is also included to abstract relevant data based on the categories required by the users (e.g. filter staff by Branch). Overall, our system was designed with classes categorised to each package by their functional roles and designed to maximise encapsulation, reusability, readability of functions/method calls and a one-for-all filtering system provided by our cache system.



(UML also attached in submission folder)

## **1.2 Assumptions**

- 1) There is only one user accessing the system at a given time
- 2) We expect users to remove or redirect staff and menu items out of the closed branch as the compulsory and minimum requirement of this application. Closing a branch without removing its staff and menu items will result in boot up error when the Application is next-run.

## **1.3.1 SOLID Design Principles**

### **Single-Responsibility Principle**

#### **1) Separate classes for different users (Admin, Manager, Staff, Order)**

The different users have methods for different responsibilities.

#### **2) Separation of Order and Cart class**

Instead of treating the Order class as one class that handles the responsibility for both ordering and cart management, we create two classes that deal with each of these responsibilities separately.

#### **3) Separation of data storages (Menu, Branch, Employee, and Order caches)**

Each cache handles a single object type.

#### **4) Separation of data handlers for data persistence.**

Each loader and saver handles a specific file type respective to corresponding object type.

### **Open-Closed Principle**

#### **1) AppCache abstract class**

The AppCache abstract class is Open-Closed compliant as it is open for extension but closed for modification. To create cache classes like MenuCache and EmployeeCache, we inherit from the AppCache class providing three common functions, while allowing for additional child-specific implementations without modifying the base class or interface itself. This design consideration is implemented in other parts of the code too.

#### **2) DataUtilities**

LoadData and SaveData interface allows extension of data handling for new file type and objects in the future.

### **Liskov's Substitution Principle**

#### **1) User classes inherit from Employee Abstract Class**

Any object of Manager, Admin or Staff class can be treated as objects of Employee abstract class. This abstraction is used in various Admin methods such as editStaffName(Employee emp, String name). This allows us to change the name of any employee regardless of their actual class. This follows Liskov's substitution principle because objects of subclasses can replace objects of superclass without breaking the program.

### **Interface Segregation Principle(ISP)**

#### **1) CacheFetcher and CacheEditor / LoadData and SaveData interfaces**

Using ISP, we decided to create more specific cache and data utilities interfaces that will be implemented by different classes, instead of having a general Cache and datautilities interface. This allowed the classes to purely depend on the respective interfaces that they require, and helped us avoid FAT Interfaces and adhere to SRP principle.

### **Dependency Inversion Principle**

#### **1) All User Interfaces implement the AppUI interface**

Dependency Inversion Principles states that lower-level modules should depend on higher-level modules so that it is easier to change the implementation details of each lower-level module without affecting the higher-level modules. Our user interfaces all extend from the AppUI class such that we can modify existing user interface implementation or add new user interfaces without affecting the higher level AppUI class.

#### **2) All caches extend the AppCache abstract class which implements CacheFetcher and CacheEditor interfaces.**

They define the structure for all future expansion to the types of caches the App implements, so they will be compatible with how the app uniquely handles its data (retrieval, creation, deletion).

### **1.3.2 Additional Design Principles**

#### **Singleton Pattern**

- 1) Singleton design pattern has been considered for the custom Cache System that we implemented for Employee (Staff, Manager, Admin), Branches, Menus, Orders and custom services (AutoCancelService). This ensures that there is only one existence of the cache storage and so, only one instance of these objects exist throughout the program to prevent data corruption and ensure proper data handling.

- 2) We implemented a class (BranchName) that imitates an Enum class but with added features like allowing addition of new “Enum” constants during runtime. The singleton pattern is leveraged to initialise only one instance of each “Enum” constant stored using a List, to allow object comparison similar to Enum constants because we use Enum constants as filtering filters.

## 1.4 Difficulties Encountered

### 1) Collaborative Coding

Initially, we were unsure of how to collaborate and code together as a group and it was inefficient as only 1 person could code while the rest of us could access but not edit the code. This was incredibly inefficient progress so we needed to find a different way to proceed for project. After doing some research, we decided to use Visual Studio Live Share (workspace), where we could all hop on the platform and code together. This made collaborating much more efficient as we could split the workload equally, i.e. each person takes one package to code.

### 2) Cache System (Data Storage System)

We were unsure of how to retrieve data after storing data into our custom storage system until we did some research and learnt about ‘Stream’ in Java. We used Stream API as a way of expressing and processing the collections of objects that are created throughout the runtime of our programme(i.e creating new menu item). It also enabled us to perform operations like filtering, mapping and sorting which were crucial in our execution of the fast food application programme.

### 3) Object conversion -> multiple different object data type / object to handle

There were many times when we had object referencing errors. For example, under the cache package, method(s) is/are designed to allow input of arguments of different object types, while returning a completely different object type. We created a new method in each enum class that receives an ‘Object’ type then checks by (instanceof) and cast the corresponding object type (e.g. Branch, BranchName etc.) before passing back to the specified retrieval method of a Cache system to filter and retrieve the cached items.

### 4) Error handling

We noticed that when the user inputs a wrong value, it would cause our program to throw errors and crash. We solved this issue by including error handling for functions that required user input and used try-catch blocks to prompt for inputs and to catch errors such as

`InputMismatchException`, `IndexOutOfBoundsException`, etc. and handle them accordingly so that our program does not crash. Moreover, we also included a general catch (`Exception e`) block in our `StartupUI` class to handle any unexpected errors that we faced. This way, our program was able to run without crashing when unexpected errors or erroneous inputs are keyed in, and returns to main page for continuous running of the App.

### 5) Security

We provided added security to access rights by implementing different user interfaces for each user type (Staff, Manager, Admin). Upon logging in, users are directed to their respective role's UIs ensuring security as staff should not access Admin actions in `AdminUI`. Furthermore, the employee object was passed to the UI to personalise each instance of the management's UI to the logged in user. An error will occur if the user were to stumble upon into the wrong UI (e.g. Staff entering `AdminUI`) as the object will not have the methods called in the specific UI, the exception is caught by `StartupUI` effectively redirecting the user back to homepage.

### 6) Overlapping features causing extensiveness of codes

We inherited `StaffUI` in our `ManagerUI` class since they have overlapping functions and this made our design much cleaner and reduced redundancy of duplicated codes.

### 7) Handling extensibility

Initially, we used enum classes for our `BranchName` class and `PaymentMethod` class. However, we faced the issue of adding a new Branch or Payment method as we could not add in more enums during runtime. Hence, we had to convert these to concrete classes and use `ArrayLists` to store the values of `BranchNamees` and payment methods, while simultaneously allowing it to mimic an enum class each up to a different extend of mimication.

## 2. UML Class Diagram

<https://github.com/lycheeyd/SC2002-Object-Orientated-Programming-Project/blob/main/UML.svg> ←please click this link to download our UML Class Diagram!

## 3. Test Case Demonstration

No.	Test Case	Screenshot

1	<p>Add a new menu item with a unique name, price, description, and category. Verify that the menu item is successfully added.</p>	<pre>[==&gt;] Manager Interface [==&gt;] (1) Display New Orders (2) View Order Details (3) Process Order (4) Order is ready (5) Track Order (6) Display Staff List (7) Add/Edit/Remove Menu items/price/availability (8) Logout  Waiting for user input: 3 Enter order ID to process: 1001 Order not found or already processed.  [==&gt;] Manager Interface [==&gt;] (1) Display New Orders (2) View Order Details (3) Process Order (4) Order is ready (5) Track Order (6) Display Staff List (7) Add/Edit/Remove Menu items/price/availability 1&gt;2&gt; Enter category of new menu item: Available categories: SIDE SET_MEAL BURGER DRINK Enter category: DRINK 100 PLUS added to menu of NTU Menu item added successfully.  Joined Ⓜ 4</pre>
2	<p>Update the price and description of an existing menu item. Verify that the changes are reflected in the menu.</p>	<pre>Waiting for user input: 7 Select function (1) Add Menu Item (2) Edit Item (3) Remove Item (4) Exit 2  Available Menu Items: (1) chicken wing (2) 100 PLUS (3) 3PC set meal (4) chicken nugget Enter the index of the menu item to edit: 3 Enter new name of menu item: chicken nugget Enter new price of menu item: 2.99 Enter new category of menu item: Available categories: SIDE SET_MEAL BURGER DRINK Enter category: SIDE 100 PLUS added to menu of NTU Removed item: name='3PC set meal', price='9.9', branch='NTU', category='SET_MEAL' Menu item edited successfully.  Joined Ⓜ 4</pre>
3	<p>Remove an existing menu item. Verify that the menu item is no longer available</p>	<pre>[6] Display Staff List (1) Add/Edit/Remove Menu items/price/availability (8) Logout  Waiting for user input: 7 Select function (1) Add Menu Item (2) Edit Item (3) Remove Item (4) Exit 3  Enter the index of the menu item to remove: Removed item: name='chicken nugget', price='6.6', branch='NTU', category='SIDE' chicken nugget removed  [==&gt;] Manager Interface [==&gt;] (1) Display New Orders (2) View Order Details (3) Process Order (4) Order is ready (5) Track Order (6) Display Staff List (7) Add/Edit/Remove Menu items/price/availability (8) Logout  Waiting for user input: 7 Select function (1) Add Menu Item (2) Edit Item (3) Remove Item (4) Exit 1  Available Menu Items: (1) chicken wing, price='3.2', branch='NTU', category='SIDE' (2) name='100 PLUS', price='1.2', branch='NTU', category='DRINK' (3) name='Garlic bread', price='2.9', branch='NTU', category='SIDE' Enter menu item: Joined Ⓜ 4</pre>
4	<p>Place a new order with multiple food items, customise some items, and choose the takeaway option. Verify that the order is created successfully.</p>	<pre>[==&gt;] Fast Food Management System [==&gt;] (1) New Order (2) View Status / Collect Order (3) Staff Login (4) Shutdown  Waiting for user input: 1 [==&gt;] Order Interface [==&gt;] Available branches: NTU JP SF  Select branch: NTU Enter 1 for DINE_IN Enter 3 for TAKEAWAY Entered 3 for TAKEAWAY  Waiting for user input: 2 Menu List: (1) name='Chicken wing', price='3.2', branch='NTU', category='SIDE' (2) name='Onion rings', price='2.7', branch='NTU', category='SIDE' (3) name='100 PLUS', price='1.2', branch='NTU', category='DRINK' (4) name='Garlic bread', price='2.9', branch='NTU', category='SIDE' (5) Check out Select items or check outs: Chicken wing added to cart.  Menu List: (1) name='Chicken wing', price='3.2', branch='NTU', category='SIDE' (2) name='Onion rings', price='2.7', branch='NTU', category='SIDE' (3) name='100 PLUS', price='1.2', branch='NTU', category='DRINK' (4) name='Garlic bread', price='2.9', branch='NTU', category='SIDE' Joined Ⓜ 4</pre> <pre>(5) Check out. Select items or check out: 5 Enter remarks for your order (optional): Thank you [==&gt;] Check Out [==&gt;] Items in cart: name='Chicken wing', price='3.2', branch='NTU', category='SIDE' name='Onion rings', price='2.7', branch='NTU', category='SIDE' Total: \$5.9 Remarks: Thank you Total: \$5.99  Available payment methods: (1) CREDIT_CARD (2) DEBIT_CARD (3) PAYPAL (4) BANK_TRANSFER (5) Cancel Order.  Select payment method: 1 Payment successful.  Order 1002 successfully added.  [==&gt;] Receipt [==&gt;] orderID: 1002 orderType: TAKEAWAY cost: \$5.99 itemList: Chicken wing, Onion rings! remarks: Thank you status: NEW Joined Ⓜ 4</pre>

5	<p>Place a new order with dine-in option. Verify that the order is created with the correct preferences</p>	<pre>(2) name='chicken nugget', price='6.9', branch='JP', category='SIDE' (3) Check out.  Select items or check out: 3 Enter remarks for your order (optional): HI  [==&gt;] Check Out [==&gt;] Item in cart: name: [CAJUN FISH], price='5.0', branch='JP', category='BURGER' DINE_IN Remarks: HI Total: \$5.00  Available payment methods: (1) CREDIT_CARD (2) DEBIT_CARD (3) CASH (4) PAYPAL (5) NETS (6) Cancel Order.  Select payment method: 1 Processing payment with CREDIT_CARD Payment successful.  Order 1083 successfully added.  [==&gt;] Receipt [==&gt;] orderID: 1083 orderType: DINE_IN cost: \$5.00 cart: [CAJUN FISH] remarks: HI status: NEW</pre>
6	<p>Simulate a payment for using a credit/debit card. Verify that the payment is processed successfully.</p>	<pre>Available payment methods: (1) CREDIT_CARD (2) DEBIT_CARD (3) CASH (4) PAYPAL (5) NETS (6) Cancel Order.  Select payment method: 1 Processing payment with CREDIT_CARD Payment successful.  Order 1083 successfully added.  [==&gt;] Receipt [==&gt;] orderID: 1083 orderType: DINE_IN cost: \$5.6 cart: [CAJUN FISH] remarks: HI status: NEW</pre>
7	<p>Simulate a payment using an online payment platform (e.g., PayPal). Verify that the payment is processed successfully</p>	<pre>Available payment methods: (1) CREDIT_CARD (2) DEBIT_CARD (3) CASH (4) PAYPAL (5) NETS (6) Cancel Order.  Select payment method: 4 Processing payment with PAYPAL Payment successful.  Order 1084 successfully added.  [==&gt;] Receipt [==&gt;] orderID: 1084 orderType: DINE_IN cost: \$5.6 cart: [Chicken wing] remarks: YO status: NEW  joined ④ 4</pre>
8	<p>Track the status of an existing order using the order ID. Verify that the correct status is displayed.</p>	<pre>Order is in the kitchen! [==&gt;] Fast Food Management System [==&gt;] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Manager Login  Waiting for user input: 3 [==&gt;] Staff Login [==&gt;] Enter User ID: Alex1 Enter Password: mel  [==&gt;] Manager Interface [==&gt;]  (1) Display New Orders (2) View Order Details (3) Process Order (4) Display Staff (5) Track Order (6) Display Staff List (7) Add/Edit/Remove Menu items/price/availability (8) Logout  Waiting for user input: 2 Enter the order ID: 1084  orderID: 1084 orderType: DINE_IN cost: \$5.2 cart: [chicken wing] remarks: YO status: NEW  joined ④ 4</pre>

9	<p>Login as a staff member and display new orders. Verify that the staff can see all the new orders in the branch he/she is working.</p>	<pre> Waiting for user input: 3 [==&gt;] Staff Login [==&gt;] Enter User ID: mel Enter Password: mel  Your password is default password. Please update your password. Enter new password: mel Re-enter new password: mel [==&gt;] Staff Login [==&gt;] Enter User ID: mel Enter Password: mel  Waiting for user input: 1 Displaying new orders...  orderID: 1081 orderType: DONE_IN cost: \$30.00 cart: [SPC set meal, chicken nugget] remarks: 3030 status: NEW  orderID: 1082 orderType: TAKEAWAY cost: \$30.00 cart: [RIES] remarks: WAS status: NEW  Joined ④ 4 </pre>
10	<p>Process a new order, updating its status to "Ready to pickup". Verify that the order status is updated correctly.</p>	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <pre> (1) Display New Orders (2) View Order Details (3) Process Order (4) Order is ready (5) Track Order (6) Display Staff List (7) Add/Edit/Remove Menu items/price/availability (8) Logout </pre> </div> <div style="width: 45%;"> <pre> [==&gt;] Manager Interface [==&gt;] (1) Display New Orders (2) View Order Details (3) Process Order (4) Order is ready (5) Track Order (6) Display Staff List (7) Add/Edit/Remove Menu items/price/availability (8) Logout  Waiting for user input: 4 Enter order ID: 1081 Order status changed to READY_TO_PICKUP  Joined ④ 4 </pre> </div> </div>
11	<p>Login as a manager and display the staff list in the manager's branch. Verify that the staff list is correctly displayed.</p>	<pre> [==&gt;] Manager Interface [==&gt;] (1) Display New Orders (2) View Order Details (3) Process Order (4) Order is ready (5) Track Order (6) Display Staff List (7) Add/Edit/Remove Menu items/price/availability (8) Logout  Waiting for user input: 6 Displaying staff list... { name='kumar Blackmore', loginID='kumarB', role='S', gender='M', age='32', branch='NTU', password='password' }, { name='Alexei Ivanov', loginID='Alexei', role='M', gender='W', age='25', branch='NTU', password='mel' } </pre>
12	<p>A manager should be able to process order as described in Test Case 9</p>	<pre> Order 1084 successfully added.  [==&gt;] Receipt [==&gt;] orderID: 1084 orderType: TAKEAWAY cost: \$30.00 cart: [chicken nugget] remarks: YO status: NEW  [==&gt;] Fast Food Management System [==&gt;] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown  Waiting for user input: 3 [==&gt;] Staff Login [==&gt;] Enter User ID: Alexei Enter Password: mel  [==&gt;] Manager Interface [==&gt;] (1) Display New Orders (2) View Order Details (3) Process Order (4) Order is ready (5) Track Order (6) Display Staff List (7) Add/Edit/Remove Menu items/price/availability (8) Logout  Waiting for user input: 3 Enter order ID: 1084 Order status changed to IN_PROGRESS </pre>
13	<p>Close a branch. Verify that the branch does not display in Customer's Interface anymore.</p>	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <pre> [==&gt;] Admin Interface [==&gt;] (1) Add Staff Accounts (2) Edit Staff Accounts (3) Assign managers to branch (4) Promote a staff (5) Transfer a manager (6) Open/Close branch (11) Logout  Waiting for user input: 10 [==&gt;] Fast Food Management System [==&gt;] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown joined ④ 4  Available branches: JP JE  Select branch: NTU NTU is removed. </pre> </div> <div style="width: 45%;"> <pre> Waiting for user input: 1 [==&gt;] Order Interface [==&gt;]  Available branches: JP JE  Select branch: NTU Branch not found. Try again.  Available branches: JP JE  Select branch: ■ joined ④ 4 </pre> </div> </div>

14

Login as an admin and display the staff list with filters (branch, role, gender, age). Verify that the staff list is correctly filtered.

```
[==] Welcome to FMSApp [==]
[==] Fast Food Management System [==]
(1) New Order
(2) Display Status / Collect Order
(3) Staff Login
(4) Shutdown

Waiting for user input: 3
[==] Staff Login [==]
Enter Login ID: kumarB
Enter Password: kumarB

Available branches:
NTU
JP
JE

Select branch: NTU
1 name='Kumar Blackmore', loginID='kumarB', role='M', gender='M', age='32', branch='NTU', password='password'
2 name='Alice Ang', loginID='AliceA', role='M', gender='F', age='27', branch='JP', password='password'

[==] Admin Interface [==]
(1) Add Staff Accounts
(2) Remove Staff Accounts
(3) Edit Staff Accounts
(4) Display staff list
(5) Assign managers to branch
(6) Promote a staff
(7) Transfer a staff
(8) Transfer a manager
(9) Add/Delete payment method
(10) Open/Close branch
(11) Logout

Waiting for user input: 4
(1) Display all staffs
(2) to filter by branch
(3) to filter by role
(4) to filter by gender
(5) to filter by age
2

[==] Admin Interface [==]
(1) Add Staff Accounts
(2) Remove Staff Accounts
(3) Edit Staff Accounts
(4) Display staff list
(5) Assign managers to branch
(6) Promote a staff
(7) Transfer a staff
(8) Transfer a manager
(9) Add/Delete payment method
(10) Open/Close branch
(11) Logout

Waiting for user input: 4
(1) Display all staffs
(2) to filter by branch
(3) to filter by role
(4) to filter by gender
(5) to filter by age
2

[==] Admin Interface [==]
(1) Add Staff Accounts
(2) Remove Staff Accounts
(3) Edit Staff Accounts
(4) Display staff list
(5) Assign managers to branch
(6) Promote a staff
(7) Transfer a staff
(8) Transfer a manager
(9) Add/Delete payment method
(10) Open/Close branch
(11) Logout
```

15

Assign managers to branches with the quota/ratio constraint. Verify that managers are assigned correctly.

```
[==] Admin Interface [==]
(1) Add Staff Accounts
(2) Remove Staff Accounts
(3) Edit Staff Accounts
(4) Display staff list
(5) Assign managers to branch
(6) Promote a staff
(7) Transfer a staff
(8) Transfer a manager
(9) Add/Delete payment method
(10) Open/Close branch

Select branch: JP
1
JP
JE

Available branches:
NTU
JP
JE

Select branch: JP
1 name='temp', loginID='temp', role='S', gender='M', age='28', branch='JP', password='password'
2 name='Justin Loh', loginID='JustinL', role='S', gender='M', age='49', branch='JP'
3 name='Tom Chan ', loginID='TomC', role='M', gender='M', age='56', branch='JP'
4 name='Kumar Blackmore', loginID='KumarB', role='M', gender='M', age='32', branch='JP'
5 name='Mary Lee', loginID='MaryL', role='S', gender='F', age='44', branch='JP'
6 name='Alice Ang', loginID='AliceA', role='M', gender='F', age='27', branch='JP'
7 name='temp2', loginID='temp2', role='S', gender='M', age='28', branch='JP', password='temp2'
8 name='temp3', loginID='temp3', role='S', gender='F', age='21', branch='JP', password='temp3'

[==] Admin Interface [==]
(1) Add Staff Accounts
(2) Remove Staff Accounts
(3) Edit Staff Accounts
(4) Display staff list
(5) Assign managers to branch
(6) Promote a staff
(7) Transfer a staff
(8) Transfer a manager
(9) Add/Delete payment method
(10) Open/Close branch

Waiting for user input: 5
Enter Login ID: KumarB

Available branches:
NTU
JP
JE

Select branch: JP
1
JP
JE

Available branches:
NTU
JP
JE

Select branch: JP
1 name='temp', loginID='temp', role='S', gender='M', age='28', branch='JP', password='password'
2 name='Justin Loh', loginID='JustinL', role='S', gender='M', age='49', branch='JP'
3 name='Kumar Blackmore', loginID='KumarB', role='M', gender='M', age='32', branch='JP'
4 name='Tom Chan ', loginID='TomC', role='M', gender='M', age='56', branch='JP'
5 name='Mary Lee', loginID='MaryL', role='S', gender='F', age='44', branch='JP', password='password'
6 name='Alice Ang', loginID='AliceA', role='M', gender='F', age='27', branch='JP', password='password'

[==] Admin Interface [==]
(1) Add Staff Accounts
(2) Remove Staff Accounts
(3) Edit Staff Accounts
(4) Display staff list
(5) Assign managers to branch
(6) Promote a staff
(7) Transfer a staff
(8) Transfer a manager
(9) Add/Delete payment method
(10) Open/Close branch
(11) Logout
```

16

Promote a staff to a Branch Manager. Verify that the staff is promoted successfully.

```
[==] Admin Interface [==]
(1) Add Staff Accounts
(2) Remove Staff Accounts
(3) Edit Staff Accounts
(4) Display staff list
(5) Assign managers to branch
(6) Promote a staff
(7) Transfer a staff
(8) Transfer a manager
(9) Add/Delete payment method
(10) Open/Close branch
(11) Logout

Waiting for user input: 6
Enter Login ID: KumarB
Kumar Blackmore
Employee KumarB successfully added.
You have successfully promoted kumar Blackmore to Manager.

[==] Admin Interface [==]
(1) Add Staff Accounts
(2) Remove Staff Accounts
(3) Edit Staff Accounts
(4) Display staff list
(5) Assign managers to branch
(6) Promote a staff
(7) Transfer a staff
(8) Transfer a manager
(9) Add/Delete payment method
(10) Open/Close branch
(11) Logout

Waiting for user input: 4
Enter Staff Role [Staff(S) / Manager (M)]: M
1 name='Kumar Blackmore', loginID='kumarB', role='M', gender='M', age='32', branch='JP', password='password'
2 name='Tom Chan ', loginID='TomC', role='M', gender='M', age='56', branch='JP', password='password'
3 name='Alice Ang', loginID='AliceA', role='M', gender='F', age='27', branch='JP', password='password'
```

17	<p>Transfer a staff/manager among branches. Verify that the transfer is reflected in the system</p>	<pre> Waiting for user input: 4 (1) to filter by branch (2) to filter by role (3) to filter by gender (4) to filter by age (5) to filter by opt  Waiting for user input: 2 Available branches: NTU JP JE  Select branch: JP {   name='Tom Chan', loginID='TomC', role='M', gender='M', age='56', branch='JP', password='password'   {     name='Tom Chan', loginID='TomC', role='M', gender='M', age='56', branch='JP', password='password'   } }  Waiting for user input: 4 (1) to filter by branch (2) to filter by role (3) to filter by gender (4) to filter by age (5) to filter by opt  Waiting for user input: 2 Available branches: NTU JP JE  Select branch: NTU {   name='Justin Loh', loginID='JustinL', role='S', gender='M', age='49', branch='NTU', password='password'   {     name='Kumar Blackmore', loginID='kumarB', role='M', gender='M', age='32', branch='NTU', password='password'   }   {     name='Mary Lee', loginID='MaryL', role='S', gender='F', age='44', branch='NTU', password='password'   } }  Waiting for user input: 7 Enter login ID: JustinL Available branches: NTU JP JE  Select branch: ntu Justin Loh has been transferred successfully </pre>
18	<p>Place a new order, check the order status using the order ID, and Collect the food. Verify that the order status changes from "Ready to pickup" to "completed."</p>	<pre> [==&gt;] Fast Food Management System [==&gt;] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown  Waiting for user input: 1 [==&gt;] Order Interface [==&gt;]  Available branches: NTU JP JE  Select branch: NTU Enter 1 for DINE_IN Enter 2 for TAKEAWAY  Waiting for user input: 1 Menu List (1) name='FRIES', price='3.2', branch='NTU', category='SIDE' (2) name='3PC set meal', price='9.9', branch='NTU', category='SET_MEAL' (3) name='Chicken nugget', price='6.6', branch='NTU', category='SIDE' (4) Check out.  Select items or check out: 1 FRIES added to cart.  joined ④  [==&gt;] Order Interface [==&gt;]  Available payment methods: (1) CREDIT_CARD (2) DEBIT_CARD (3) CASH (4) PAYPAL (5) QR_CODE (6) Cancel Order.  Select payment method: 1 Processing payment with CREDIT_CARD Payment successful.  joined ④  [==&gt;] Order Interface [==&gt;]  (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown  Waiting for user input: 2 [==&gt;] Track Order Interface [==&gt;] Enter orderID: 1001  Order is ready! Do you want to collect your order? (Y/N): Y You have collected your order.  [==&gt;] Fast Food Management System [==&gt;] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown  Waiting for user input: 2 [==&gt;] Track Order Interface [==&gt;] Enter orderID: 1001  Order already completed! </pre>
19	<p>Attempt to add a menu item with a duplicate name. Verify that an appropriate error message is displayed.</p>	<pre> Enter User ID: Alexei Enter Password: mel  [==&gt;] Manager Interface [==&gt;]  (1) Display New Orders (2) View Order Details (3) Process Order (4) Add New Order (5) Track Order (6) Display Staff List (7) Add/Update/Remove Menu items/price/availability (8) Logout  Waiting for user input: 7 Select function: (1) Add Menu Item (2) Edit Item (3) Remove Item (4) Exit 1  Available Menu Items: (1) name='FRIES', price='3.2', branch='NTU', category='SIDE' (2) name='3PC set meal', price='9.9', branch='NTU', category='SET_MEAL' (3) name='chicken nugget', price='6.6', branch='NTU', category='SIDE' Enter new menu item: FRIES Menu item already exists.  joined ④ </pre>

20	<p>Attempt to process an order without selecting any items. Verify that an error message prompts the user to select items</p>	<pre>[=+=] Welcome to FOMApp [=+=] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown  Waiting for user input: 1  [=+=] Order Interface [=+=]  Available branches: NTU JP JE  Select branch: NTU Enter 1 for DINE_IN Enter 2 for TAKEAWAY</pre> <pre>Waiting for user input: 2  Menu List (1) name='FRIES', price='3.2', branch='NTU', category='SIDE' (2) name='3PC set meal', price='9.9', branch='NTU', category='SET_MEAL' (3) name='chicken nugget', price='6.6', branch='NTU', category='SIDE' (4) Check out.  Select items or check out: 4 Cannot check out with an empty cart. Please select at least one item.  Menu List (1) name='FRIES', price='3.2', branch='NTU', category='SIDE' (2) name='3PC set meal', price='9.9', branch='NTU', category='SET_MEAL' (3) name='chicken nugget', price='6.6', branch='NTU', category='SIDE' (4) Check out.  Select items or check out: 1 FRIES added to cart.</pre>
21	<p>Add a new payment method. Verify that the new payment method is successfully added.</p>	<pre>[=+=] Fast Food Management System [=+=] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown  Waiting for user input: 1  [=+=] Order Interface [=+=]  Available branches: NTU JP JE  Select branch: JP Enter 1 for DINE_IN Enter 2 for TAKEAWAY  Waiting for user input: 2  Menu List (1) name='CAJUN FISH', price='5.6', brnch='JP' (2) name='chicken nugget', price='6.9', brnch='JP' (3) Check out.  Select items on check out: 1 CAJUN FISH added to cart.</pre> <pre>[=+=] Check Out [=+=] Items in cart: name='CAJUN FISH', price='5.6', branch='JP' name='chicken nugget', price='6.9', branch='JP' TAKEAWAY Remarks: ve hugs Total: \$12.50  Available payment methods: (1) CREDIT_CARD (2) DEBIT_CARD (3) CASH (4) PAYPAL (5) NETS (6) Hugs (7) Cancel Order.  Select payment method: 6 Processing payment with Hugs Payment successful.  Order 1002 successfully added.  [=+=] Receipt [=+=] orderID: 1002 orderType: TAKEAWAY cost: \$12.50 cart: [CAJUN FISH, chicken nugget] remarks: ve hugs status: NEW</pre>
22	<p>Open a new branch. Verify that the new branch is added without affecting existing functionalities.</p>	<pre>[=+=] Admin Interface [=+=] (1) Add Staff Accounts (2) Remove Staff Accounts (3) Edit Staff Accounts (4) Display staff list (5) Assign managers to branch (6) Promote staff (7) Transfer a staff (8) Transfer a manager (9) Add/Remove payment method (10) Open/Close branch (11) Logout  Waiting for user input: 10 (0) To close a branch (1) To open a branch  Waiting for user input: 1  Enter new branch name: NUS Enter new branch location:clementi Enter new branch staff quota:6 Branch NUS successfully added.  [=+=] Admin Interface [=+=] (1) Add Staff Accounts (2) Remove Staff Accounts (3) Edit Staff Accounts joined ④ ⑤ Live Share Chat: 7 new</pre> <pre>[=+=] Admin Interface [=+=] (4) Display staff list (5) Assign managers to branch (6) Promote staff (7) Transfer a staff (8) Transfer a manager (9) Add/Remove payment method (10) Open/Close branch (11) Logout  Waiting for user input: 11 Logging Out... [=+=] Fast Food Management System [=+=] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown  Waiting for user input: 1  [=+=] Order Interface [=+=]  Available branches: NTU NUS JP JE  Select branch: NUS Enter 1 for DINE_IN joined ④ ⑤ Live Share Chat: 7 new</pre>
23	<p>Place a new order and let it remain uncollected beyond the specified Timeframe. Verify that the order is automatically canceled and removed from the "Ready to pickup" list.</p>	<pre>(2) View Order Details (3) Process Order (4) Order is ready (5) Track Order (6) Display Staff List (7) Add/Edit/Remove Menu items/price/availability (8) Logout  Waiting for user input: 3  Enter order ID: 1001 Order status changed to IN_PROGRESS  [=+=] Manager Interface [=+=]  (1) Display New Orders (2) View Order Details (3) Process Order (4) Order is ready (5) Track Order (6) Display Staff List (7) Add/Edit/Remove Menu items/price/availability (8) Logout  Waiting for user input: 4  Enter order ID: 1001 Order status changed to READY_TO_PICKUP</pre> <pre>[=+=] Manager Interface [=+=] (1) Display New Orders (2) View Order Details (3) Process Order (4) Order is ready (5) Track Order (6) Display Staff List (7) Add/Edit/Remove Menu items/price/availability (8) Logout  Waiting for user input: Order 1001 automatically cancelled. (Uncollected beyond specified time) 8 Logging out... [=+=] Fast Food Management System [=+=] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown  Waiting for user input: 2  [=+=] Track Order Interface [=+=] Enter orderID: 1001 Order cancelled or expired!</pre>

24	<p>Attempt to log in with incorrect credentials as a staff member.</p> <p>Verify that an appropriate error message is displayed.</p>	<pre>[==&gt;] Fast Food Management System [==&gt;] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown  Waiting for user input: 3 [==&gt;] Staff Login [==&gt;] Enter User ID: Alexia Enter Password: wrong  Wrong User ID or Password! (1) Try again. (Any) Return to previous page.  Waiting for user input: ■</pre>
25	<p>Log in as a staff member, change the default password, and log in again with the new password.</p> <p>Verify that the password change functionality works as expected.</p>	<pre>[==&gt;] Staff Login [==&gt;] Enter User ID: kumarB Enter Password: password  Your password is default password. Please update your password. Enter new password: mel Re-enter new password: mel  [==&gt;] Staff Interface [==&gt;] (1) Display New Orders (2) View Order Details (3) Process Order (4) Order is Ready (5) Track Order (6) Logout  Waiting for user input: 6 Logging out... [==&gt;] Fast Food Management System [==&gt;] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown  Waiting for user input: 3 [==&gt;] Staff Login [==&gt;] Enter User ID: kumarB Enter Password: mel joined ④ ④ Live Share Chat: 7 new  [==&gt;] Staff Interface [==&gt;] (1) Display New Orders (2) View Order Details (3) Process Order (4) Order is Ready (5) Track Order (6) Logout  Waiting for user input: ■</pre>
26	<p>Upload a staff list file during system initialization. Verify that the staff list is correctly initialized based on the uploaded file.</p>	<pre>Branch NTU successfully added. Branch JP successfully added. Branch JE successfully added. pensi added to menu of JE FRIED CHICKEN added to menu of NTU CAJUN FISH added to menu of JP COLE SLAW added to menu of JE 3PC set meal added to menu of NTU 3PC set meal added to menu of JE chicken nugget added to menu of JP chicken nugget added to menu of NTU Employee boss successfully added. Employee kumarB successfully added. Employee Justin successfully added. Employee Alice successfully added. Employee MaryL successfully added. Employee Alexei successfully added.  [==&gt;] Welcome to FOMSapp [==&gt;]</pre>
27	<p>Perform multiple sessions of the application, adding, updating, and removing menu items. Verify that changes made in one session persist and are visible in subsequent sessions</p>	<pre>[==&gt;] Welcome to FOMSapp [==&gt;] [==&gt;] Fast Food Management System [==&gt;] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown  Waiting for user input: 3 [==&gt;] Staff Login [==&gt;] Enter User ID: boss Enter Password: meow  [==&gt;] Admin Interface [==&gt;] (1) Add Staff Accounts (2) Remove Staff Accounts (3) Edit Staff Accounts (4) Display staff list (5) Assign managers to branch (6) Promote staff (7) Transfer a staff (8) Transfer a manager (9) Add/Remove payment method (10) Open/Close branch (11) Logout  Waiting for user input: 3 Enter Login ID: boss joined ④ ④ Live Share Chat: 7 new  [==&gt;] Admin Interface [==&gt;] (1) Edit staff name (2) Edit staff loginID (3) Edit staff gender (4) Edit staff age (5) Edit staff password ... Enter staff password: a Account edited successfully!  [==&gt;] Admin Interface [==&gt;] (1) Add Staff Accounts (2) Remove Staff Accounts (3) Edit Staff Accounts (4) Display staff list (5) Assign managers to branch (6) Promote staff (7) Transfer a staff (8) Transfer a manager (9) Add/Remove payment method (10) Open/Close branch (11) Logout  Waiting for user input: 11 Logging Out... [==&gt;] Fast Food Management System [==&gt;] (1) New Order (2) Check Order Status / Collect Order (3) Staff Login (4) Shutdown joined ④ ④ Live Share Chat: 7 new  Waiting for user input: 4 [==&gt;] Terminating system processes... [==&gt;]  [==&gt;] Saving files... [==&gt;]  Branch data successfully saved. Menu data successfully saved. Employee data successfully saved.  [==&gt;] Shutting down... [==&gt;] PS C:\Users\User\Desktop\NTU\Yr2 Sem2\SC2002 joined ④ ④ Live Share Chat: 7 new</pre>