CS3343 Software Engineering Practice

Project title:

Service Plan Searching Assistant - Search It(SI)

Test Report

Group Member:

Ng Chi Kin

Wu Wei Jian

Tsui Ka Kin

Tam Kwan Leong

Chan Ching Yi

Table of content

[**Testing Strategy**](#_a4s3bquz8i3o) **3**

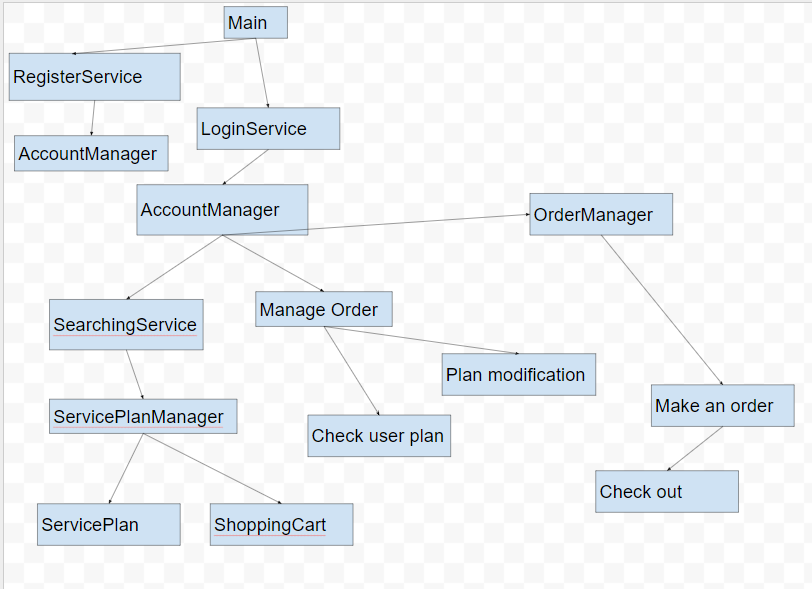
[**Bug report**](#_r0fa4k20i8v5) **14**

[**Code refactoring**](#_83d1zcssdvt2) **16**

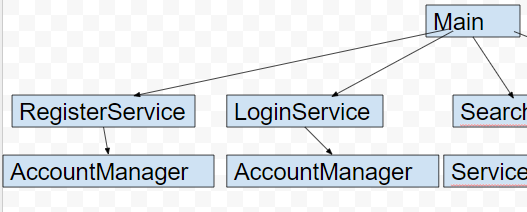
# Testing Strategy

In the project, we have used the bottom-up approach as our testing strategy, since it tests from the bottom root, it doesn’t need any test stub to test the program. Also, we separate the project into a few of part to test. After the unit test, we will integrate each member’s testing progress and combine to test the whole program.

Hierarchy Diagram



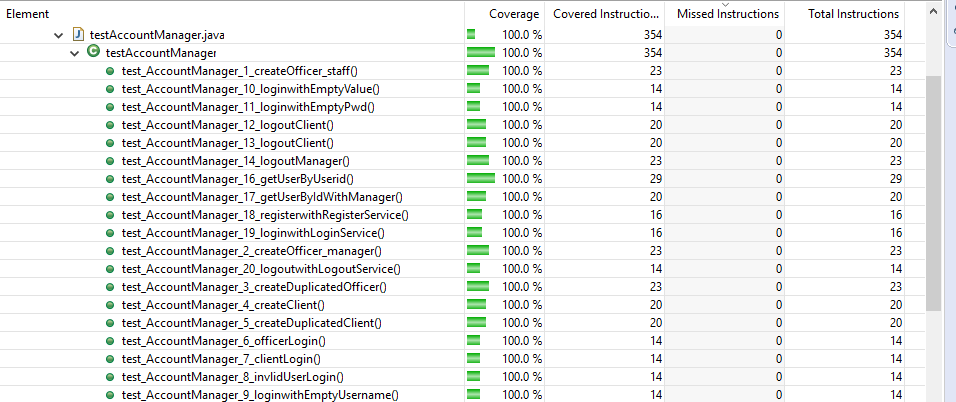
Testing with account registration



* Unit testing:

Unit testing 1: AccountManager

→ testAccountManager.java



1. test\_Account\_Manager\_1\_createOfficer\_staff()
   1. Purpose: testing creating an officer(staff) account function
2. test\_Account\_Manager\_2\_createOfficer\_manager()
   1. Purpose: testing creating an officer(manager) account function
3. test\_Account\_Manager\_3\_createDuplicatedOfficer()
   1. Purpose: testing creating a duplicated officer account result
4. test\_Account\_Manager\_4\_createClient()
   1. Purpose: testing creating a client account function
5. test\_Account\_Manager\_5\_createDuplicatedClient()
   1. Purpose: testing creating a duplicated client account result

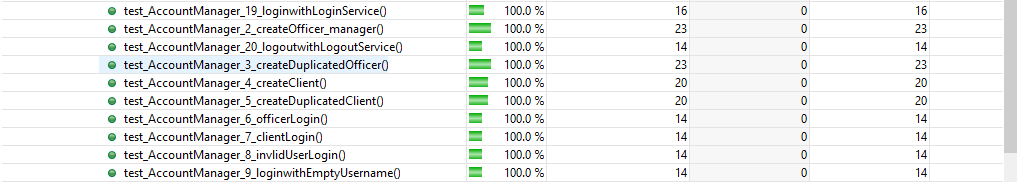
* Integration testing

Integration testing 1: RegisterService + AccountManager

* System testing

Main → registerService → Accountmanager

Testing with the login function



* Unit testing:

Unit testing 1: LoginService

Purpose: testing the login service if it works. Typing the username and password.

Unit testing 2: AccountManager

Purpose: testing any function relate with login service

* Integration testing

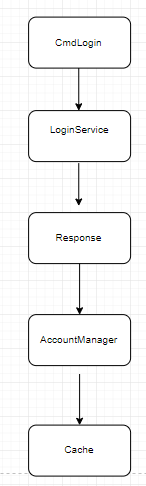
Integration testing 1: loginwithLoginService + registerService

Purpose: register an account and use it to sign in the system

Integration testing 2: officerLogin + registerService

Purpose: register an officer account and use it to sign in

Integration testing 3: clientLogin + registerService

Purpose: register a client account and use it to sign in

Integration testing 4: loginwithLoginService + registerService + accountManager

Purpose: register ab account and use it to sign in the system. Also, compare the user input information which is match with the existing user list.

Integration testing 5: officerLogin + registerService + accountManager

Purpose: register an officer account and sign in the system, and compare the user input.

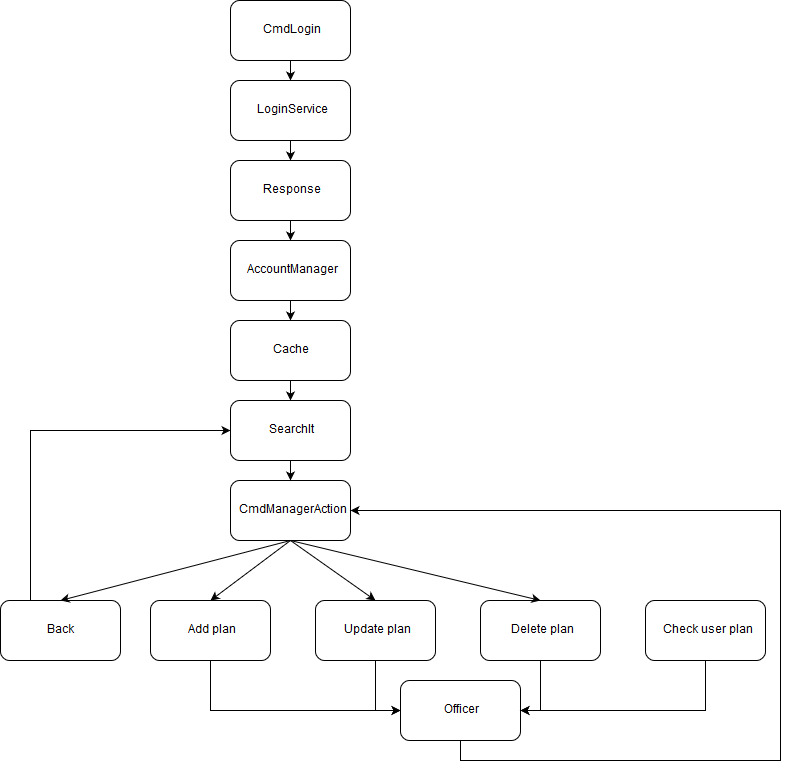
Integration testing 5: clientLogin + registerService + accountManager

Purpose: register a client account and sign in the system, and compare the user input when it is matched with the user list

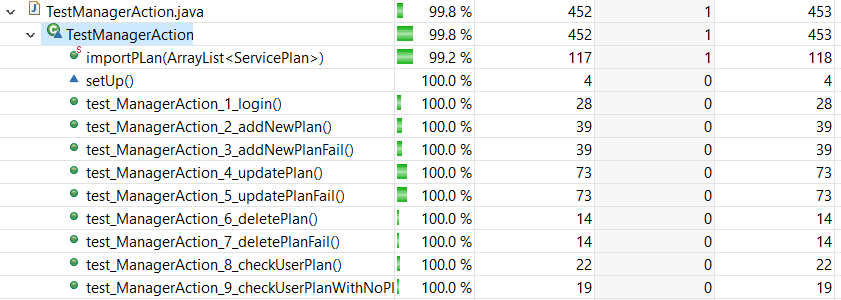
* System testing

CmdLogin → LoginService → Response → accountManager → Cache

Testing with the manager action

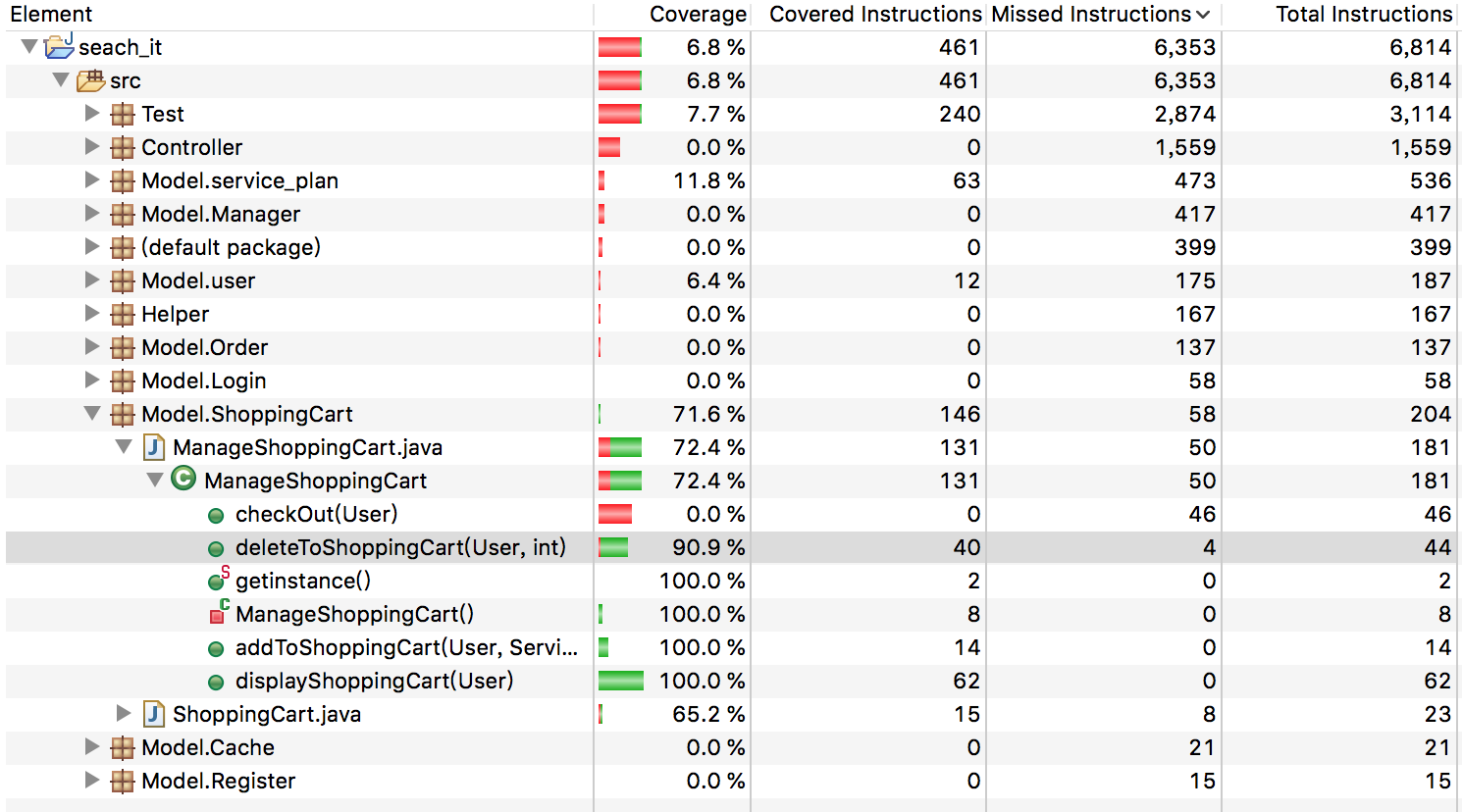


* Unit testing:



* + Unit testing 1: CmdLogin
  + Unit testing 2: LoginService
  + Unit testing 3: Response
  + Unit testing 4: AccountManager
  + Unit testing 5: Cache
  + Unit testing 6: SearchIt
  + Unit testing 7: Back
  + Unit testing 8: CmdManagerAction
  + Unit testing 9: Add plan
  + Unit testing 10: Update plan
  + Unit testing 11: Delete plan
  + Unit testing 12: Check user plan
  + Unit testing 13: Officer
* Integration testing:
  + Integration testing 1: [CmdManagerAction] + [Back]
  + Integration testing 2: [CmdManagerAction] + [Add plan]
  + Integration testing 3: [CmdManagerAction] + [Update plan]
  + Integration testing 4: [CmdManagerAction] + [Delete plan]
  + Integration testing 5: [CmdManagerAction] + [Check user plan]
  + Integration testing 2: [Add plan] + [Officer]
  + Integration testing 3: [Update plan] + [Officer]
  + Integration testing 4: [Delete plan] + [Officer]
  + Integration testing 5: [Check user plan] + [Officer]
* System testing: [Main] + login function + [CmdManagerAction] + [Back] + [Add plan] + [Update plan] + [Delete plan] + [Officer]

Testing with a shopping cart



Unit testing :

1. addToShoppingCart
2. deleteToShoppingCart
3. displayShoppingCart

Integration testing:

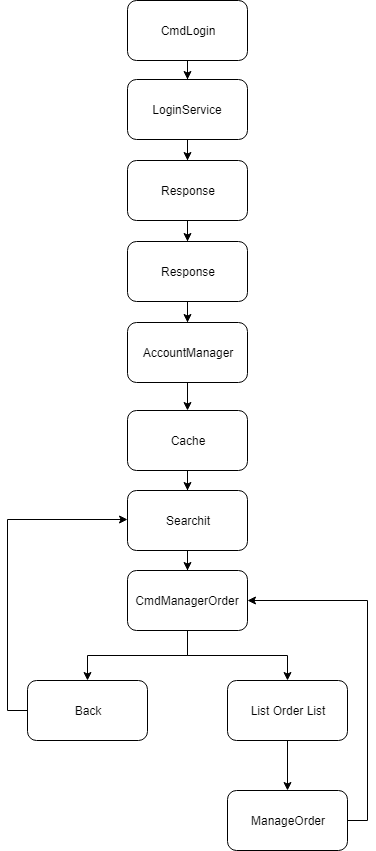
1. CmdShowPlanDetails -> addToShoppingCart
2. CmdShowShoppingCart -> deleteToShoppingCart
3. CmdShowShoppingCart ->displayShoppingCart

System testing :

Main -> CmdShowShoppingCart/CmdShowPlanDetails ->addToShoppingCart / deleteToShoppingCart /displayShoppingCart

Testing with making an order

ManagerOrderTest.java



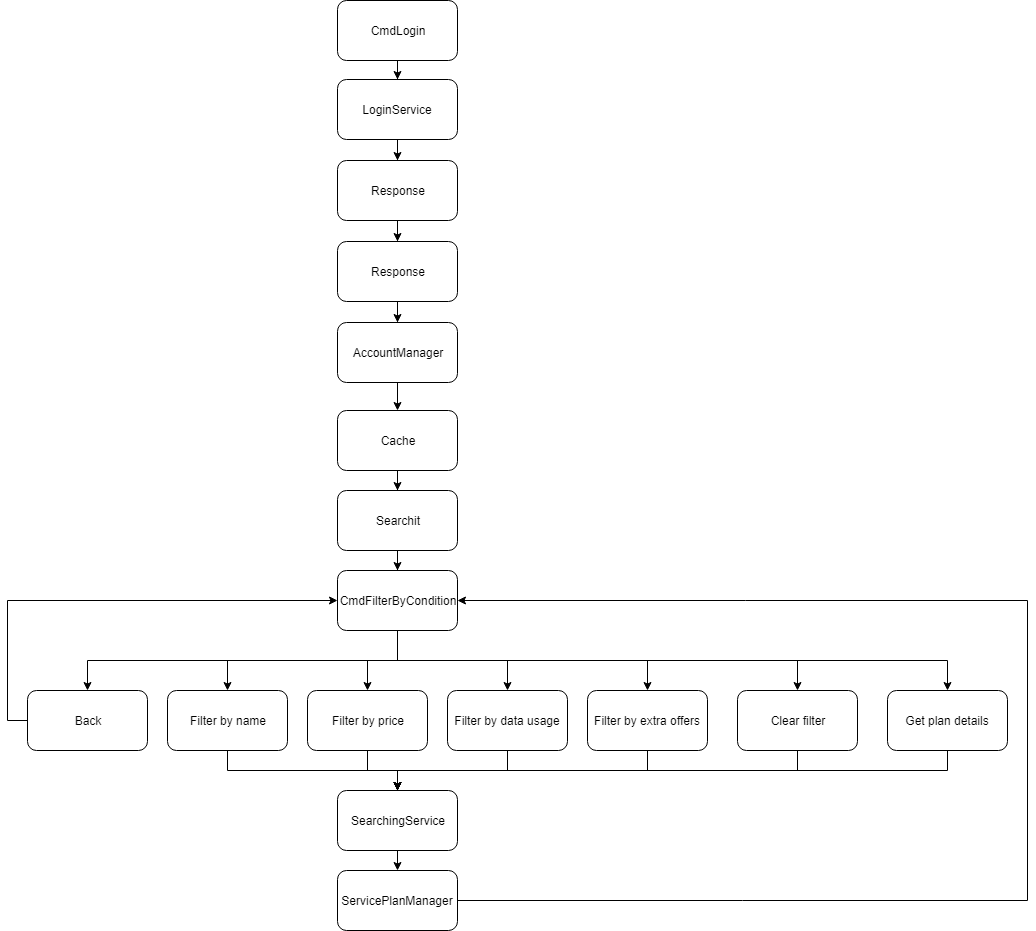


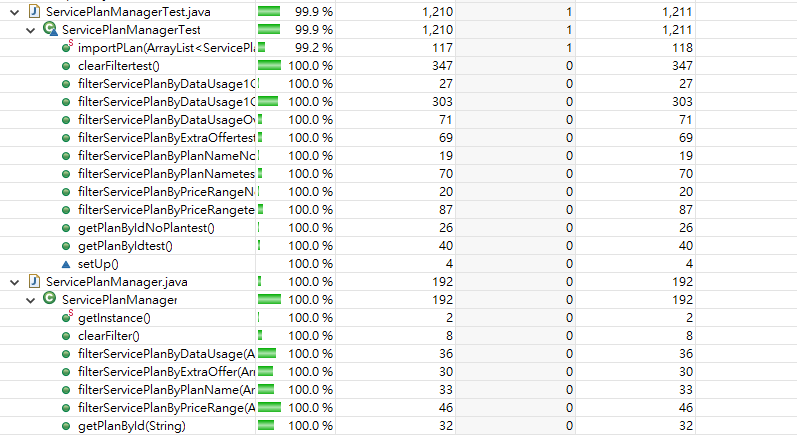
Unit test

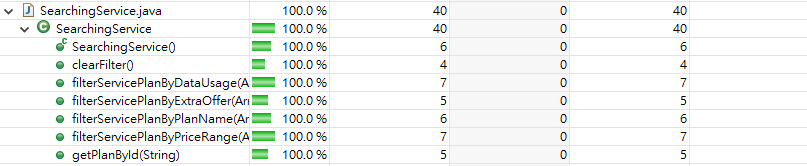
* Unit testing 1: **displayOrdertest():** use to test the order list is correct or not that adds a record in it
* Unit testing 2: **displayOrderNoPlantest():** use to test the order list is correct or not that add a record in it

Integration test

ServicePlanManagerTest.java







* Integration testing 1: **impoerPLan():** use to import plan data by json file
* Integration testing 2: **filterServicePlanByPlanNametest():** use to search the plan by valid name
* Integration testing 3: **filterServicePlanByPlanNameNoPlantest():** use to search the plan by invalid name
* Integration testing 4: **filterServicePlanByPriceRangetest():**use to search the plan between a range
* Integration testing 5: **filterServicePlanByPriceRangeNoPlantest():** use to search the plan between a range
* Integration testing 6: **filterServicePlanByDataUsage1GBtest():** use to search the plan that less than 1GB
* Integration testing 7: **filterServicePlanByDataUsage1GBTo5GBtest():** use to search the plan that between 1GB to 5GB
* Integration testing 8: **filterServicePlanByDataUsageOver5GBtest():**use to search the plan that over 5GB
* Integration testing 9: **filterServicePlanByExtraOffertest():** use to search the plan that having a extra offer

# Bug report

Bug-00001

**Title**: **Looping when filter by data usage**

**Status**: Closed

**Product**: Search It

**Type**: Crash

**Component**: Manager

**Version**: 1.0

**Platform**: PC Window

**Importance**: Critical

**Reported**: 2018-11-11 15:44:24 HKT by Terry Tam

**Modified**: 2018-11-13 20:20:55 HKT

**Assignee**: Leo Tsui



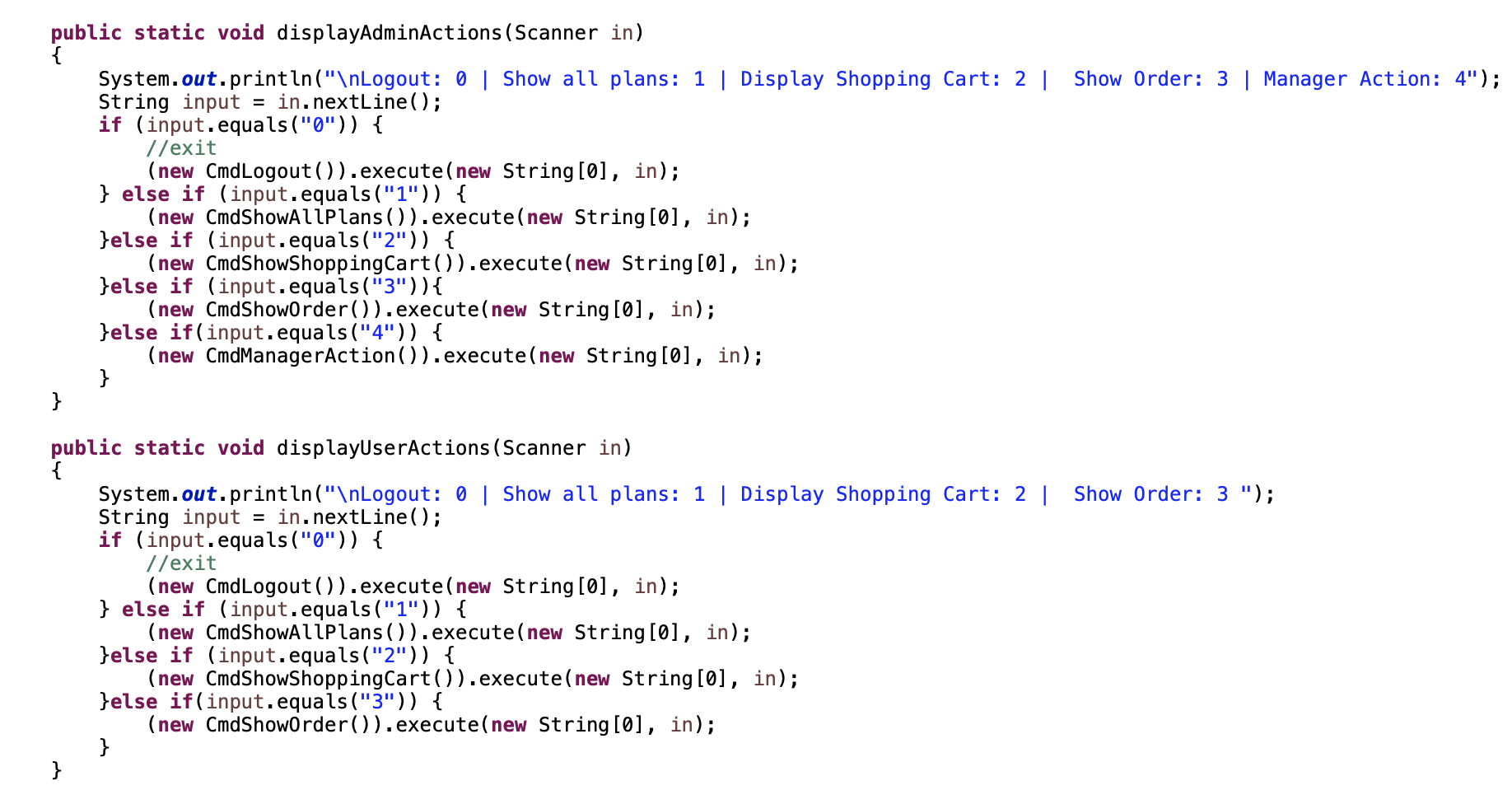




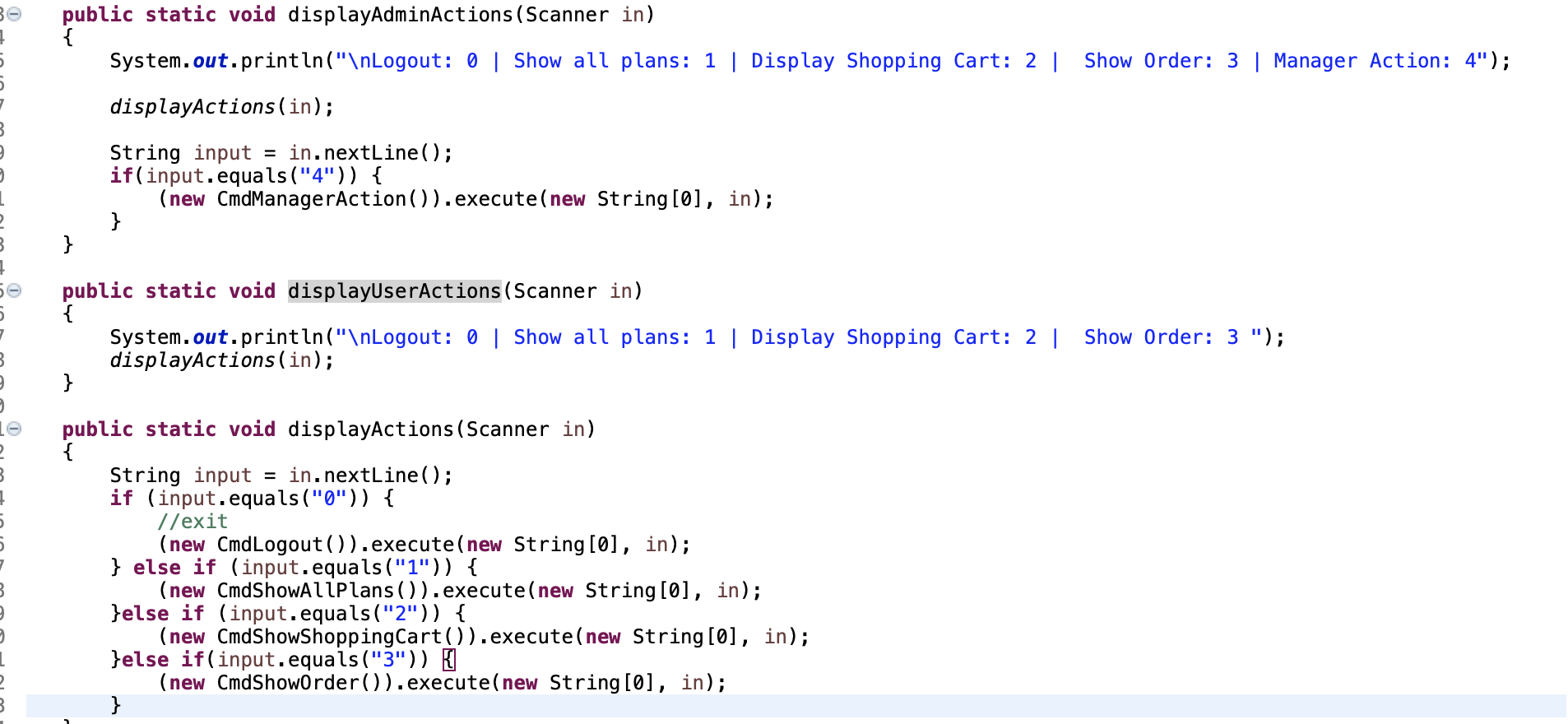
# Code refactoring

1. displayAdminActions function + displayUserActions function

We can see that admin actions and user actions have the same command with 0 to 3, so we can separate it and group it in a new function, displayActions.



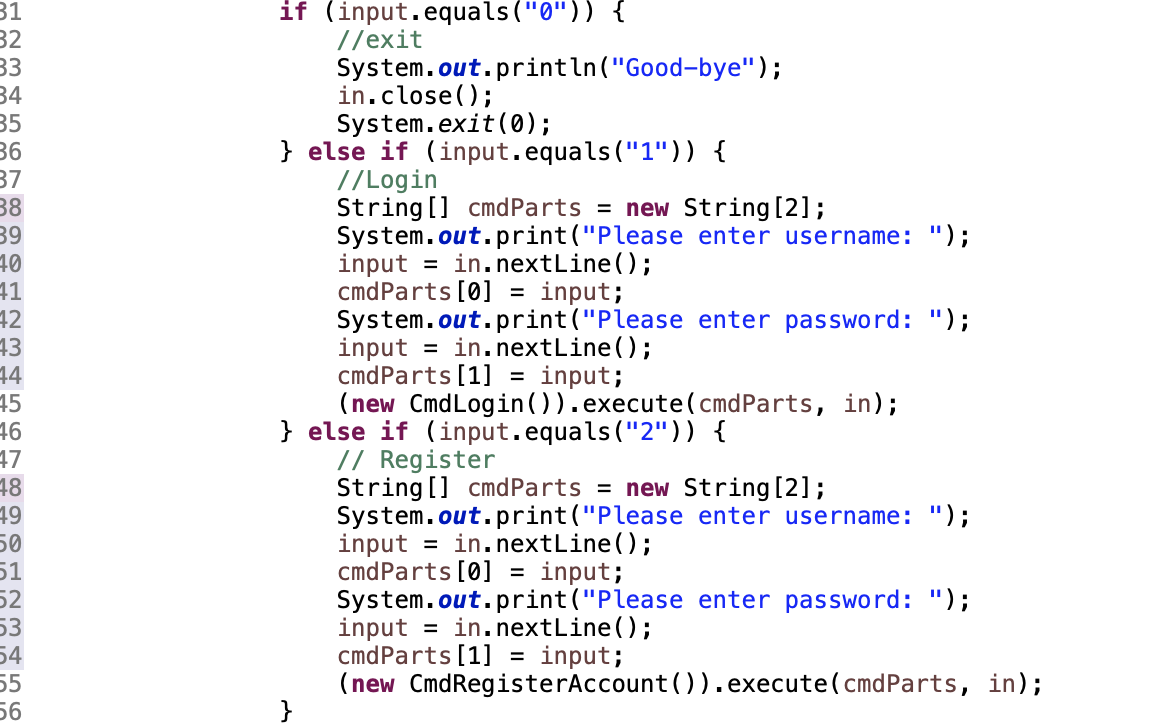
After



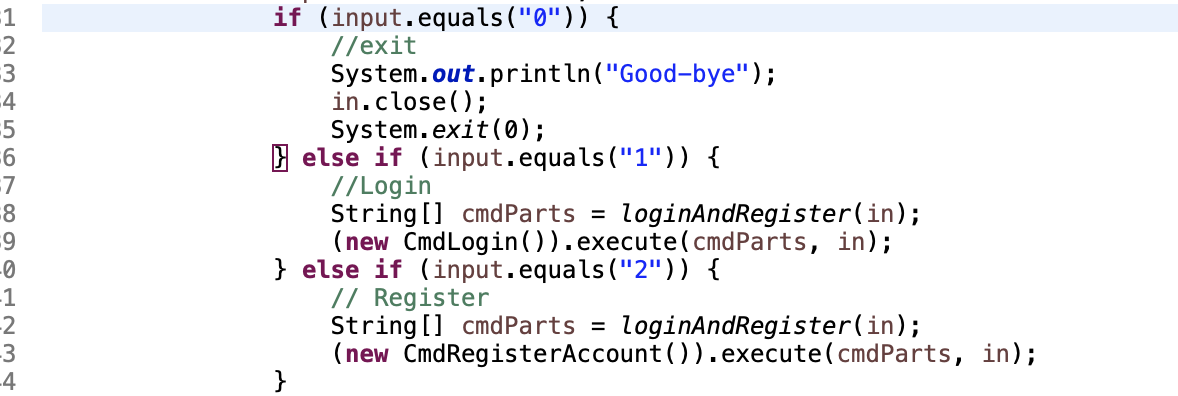
2. Login and register

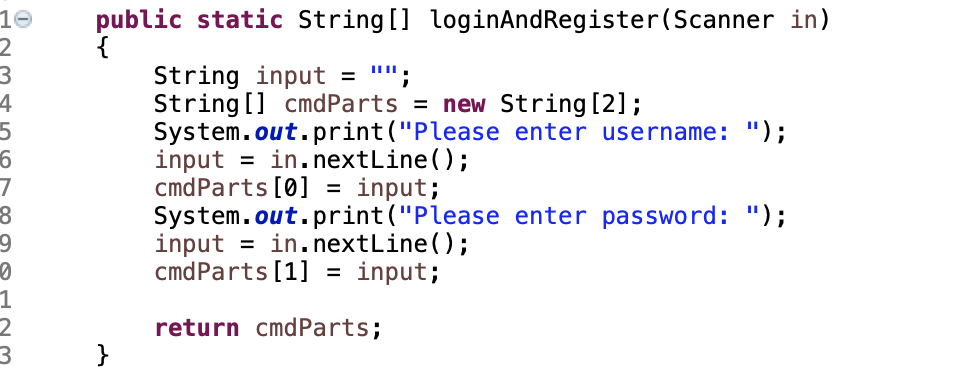
We can see that asking the username and password had been duplicated. Therefore, we can move them to a new function, loinAndRegister.

Before



After





3. Show order

In the displayOrder function, we can see that it includes adding order and display any service plan in the shopping cart. Therefore, we can separate the display shopping cart session into the new function, printOrder.

Before



After

