BIRMINGHAM CITY University

Course: Software Design

Report: D2- Enriching a Given Design

Code: CMP5354

Group Members:

1. **Sandip Magar (23189646)**
2. **Sujal Manandhar (23189654)**
3. **Nirdeshika Pandey (23189630)**

Date: 2024/10/09

Page Count: 23

Word Count: 1521

TABLE OF CONTENTS

Lists of Figures

Social Ads System Project

1. Introduction

The goal of this report is to provide a description of the overall functionality of the Social Ads system and its enhancements. The system consists of two sub-systems: the advertiser sub-system and the consumer sub-system. Upon subscribing, users receive a loyalty card, which comes with a monthly fee. This card provides additional privileges beyond basic system access. Advertisers can promote business services, while consumers can review and rate those services. Consumers can also enjoy discounts on products or services through the loyalty card. The enhancements involve implementing specific design patterns to meet additional requirements and extend the system's functionality. The system is implemented in Java, with diagram representations created using Draw.io.

a) Case Study

Online Social Advertisement is a marketing agency that aims to establish an online system—Social Ads—to host advertisements for businesses' services, including location, offerings, opening hours, promotional deals, and service updates. Social Ads is composed of two sub-systems: the advertiser sub-system and the consumer sub-system. Both advertisers and consumers can subscribe for a monthly fee, receiving a loyalty card that grants access to additional system services. For example, advertisers can promote their services, while consumers can provide reviews and ratings, as well as receive discounts.

The system should be designed to allow for future upgrades that can introduce additional services. This project involves enriching a partial class diagram and extending the existing Java implementation to incorporate these new requirements.

b) Problem Narrative

In the Social Ads system, consumers can subscribe for a monthly fee, creating either an individual or corporate profile. Subscribers receive a loyalty card that is used for verifying benefits and uploading reviews or ratings. Consumers can review and rank services, search in various modes, and enjoy perks such as product discounts, priority bookings, and gym membership discounts. The loyalty card ensures that consumers can claim these benefits and limits reviews or rankings to subscribed users.

Some of the key aspects of system's context:

- Social Ads is a UK-based website for advertisers to promote services and for consumers to rate, review, and receive discounts.
- The system maintains a database of advertisements and consumer reviews/ratings.
- Advertisers are for-profit organizations.
- Individual consumers must be over 18, verified via ID (driving license or passport).
- Corporate consumers are assumed to be over 18 and employees of the registered business.

- Subscription payments are made in British pounds.
- Bank details verification and financial transactions are outside the system's scope.
- An external mail server handles communication between advertisers and consumers.
- The system is notified upon completion of any transaction.

c) Aims and Objectives

The primary aim of this report is to provide a comprehensive overview of the process involved in enhancing the Social Ads advertiser sub-system. Specific objectives include:

- **Application of Design Patterns:** To detail the application of relevant design patterns to meet the new requirements, ensuring a flexible and maintainable system architecture.
- **Implementation and Extension:** To discuss the extension of the Java implementation in line with the enriched class diagram and new requirements.
- **Sequence Diagrams:** To provide an effective interaction diagram to showcase how the system works.

This report will provide insights into the development and enhancement process of the Social Ads system, emphasizing the importance of design patterns and systematic implementation to achieve a robust and flexible system.

2. Requirement Table

Req ID	Description
R1	Requirements table for the consumer sub-system of the Social Ads system.
R1.1	The consumer is requested to choose whether they are a corporate user or an individual consumer.
R1.2	The consumer enters their personal details (and company details if relevant) in the form.
R1.3	The system produces a 5-digit ID number for the newly created consumer
R1.4	The system creates a Loyalty Card for the consumer
R2	The system must allow the consumer to search for services in post code search mode
R2.1	The consumer inputs a sector of advertisements required.
R2.2	The consumer enters a postcode.
R2.3	The system returns a list of all the advertisements for businesses nearby the entered post code.
R3	The system must allow the consumer to search for advertisements in a certain money range
R3.1	The consumer inputs a sector of advertisements required
R3.2	The consumer enters a suitable money range.

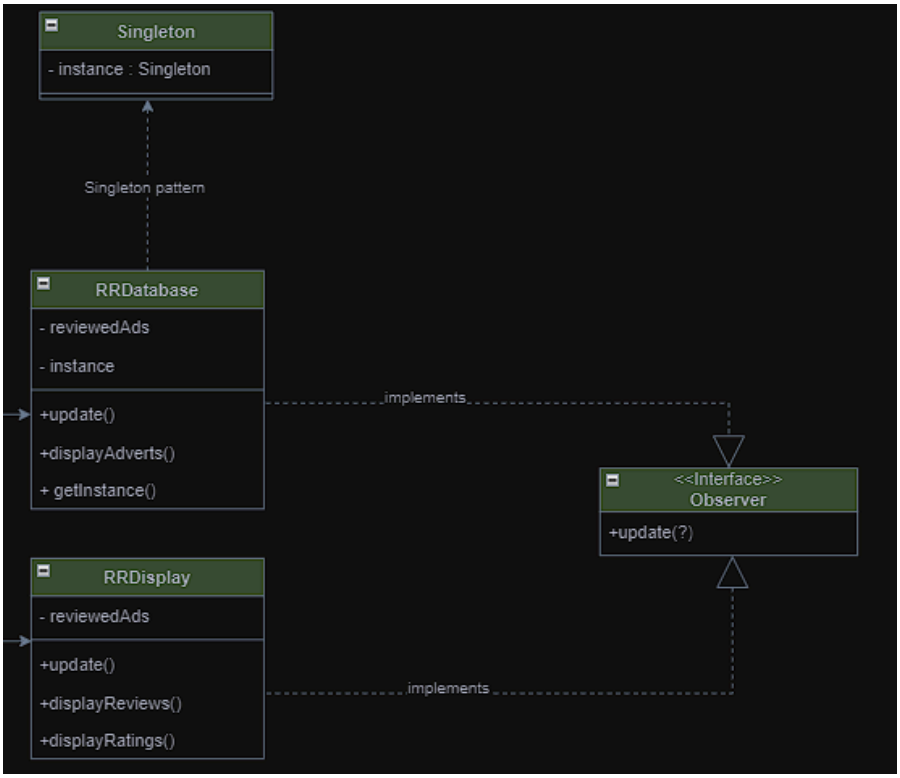
R3.3	The system returns a list of all the advertisements with price within the specified money range.
R4	The system must allow the consumer to search for advertisements with a given average rating.
R4.1	The consumer inputs the sector of advertisements required
R4.2	The consumer inputs an average star ranking or review rating.
R4.3	The system returns a list with all advertisements with the requested ranking or review.
R5	On being informed of a transaction with respect to an advertisement the reviewable/ratable advertisements are updated on the Loyalty Card.
R6	The system shall allow the consumer to view the reviewable/ratable advertisements on the Loyalty Card.
R7	The system shall allow the consumer to upload a ranking and/or review on an advertisement on the Loyalty Card
R7.1	The consumer selects the advertisement of concern from the Loyalty Card.
R7.2	The consumer enters the written review or rating.
R8	The system shall maintain a database of reviewed/rated advertisements
R9	The system shall maintain a database of advertisements
R10	The system shall allow the consumer to view all searched advertisements
R11	The system shall allow the consumer to view all reviewed/rated advertisements

Table 1: Requirements table for the consumer sub-system of the Social Ads system

3. Extend the Provided Class Diagram

a) **“RRDatabase”** and **“RRDisplay”** to implementing the interface **“Observer”**:

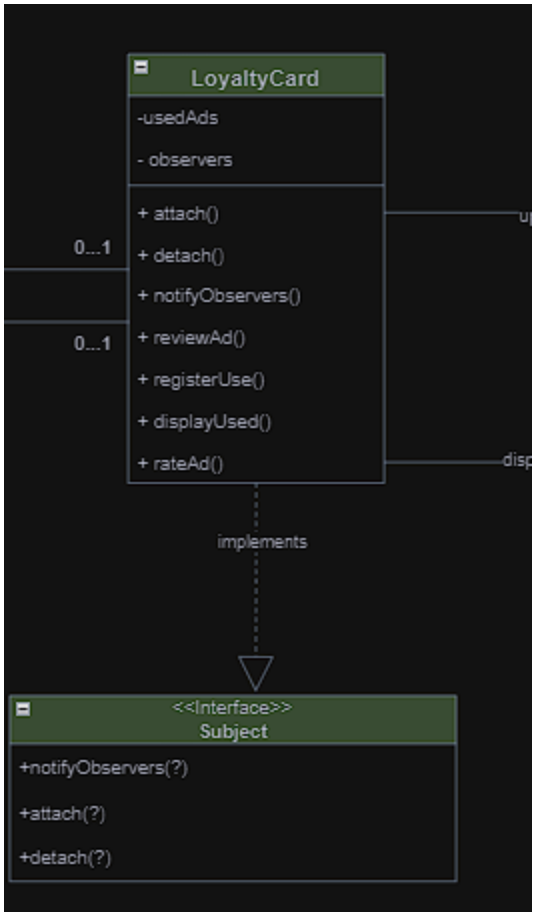
Since **“Observer”** is just an interface meaning that it is an abstract class so exact functionality is not done in this rather in **“RRDatabase”** and **“RRDisplay”** class that are created by implementing the Observer. So **“RRDatabase”** and **“RRDisplay”** are added to the diagram to show how the services are completed. A clear depiction of **“RRDatabase”** and **“RRDisplay”** implemented from Observer is also clearly shown using the diagram.



Fig(1): "RRDatabase" and "RRDisplay" to implementing the interface "Observer"

b) **"LoyaltyCard"** to implementing the interface **"Subject"**:

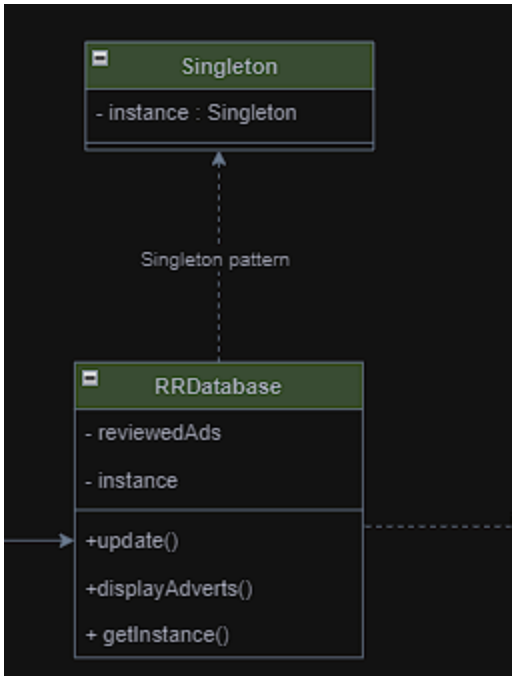
Since **"Subject"** is just an interface meaning that it is an abstract class so exact functionality is not done in this rather in the **"Loyalty"** class that is created by implementing the **"Subject"**. So **"LoyaltyCard"** are added to the diagram to show operations like adding new reviews and rating to the advertisement. A clear depiction of **"LoyaltyCard"** is implemented from **"Subject"** is also clearly shown using the diagram.



Fig(2): "LoyaltyCard" to implementing the interface "Subject"

c) **"ReviewDatabase"** as a Singleton:

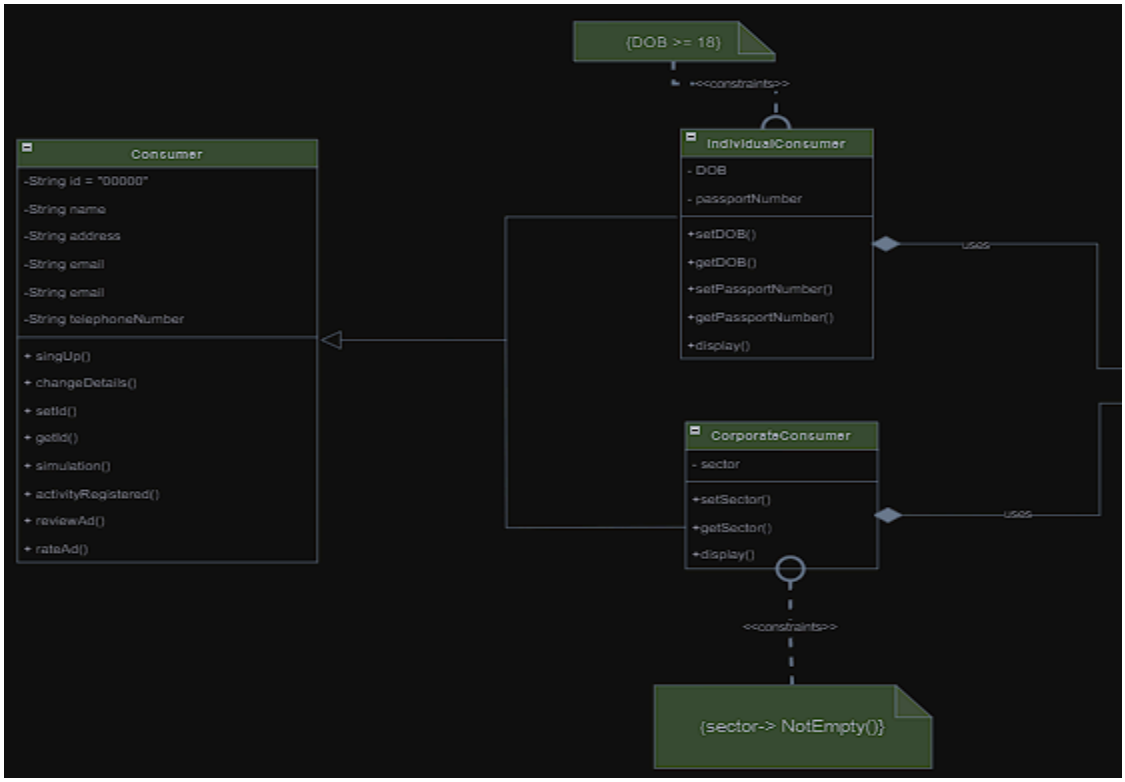
RRDatabase is declared as a Singleton meaning that one has power over all.



Fig(3): "ReviewDatabase" as a Singleton

d) **"IndividualConsumer"** and **"CorporateConsumer"** as subclasses of **"Consumer"** with suitable methods:

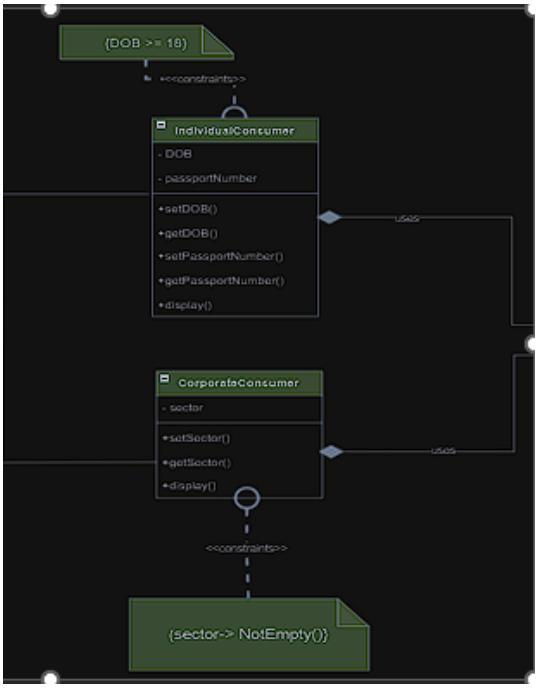
"Consumer" is superclass but users can differ depending on for what purpose they are using. **"IndividualConsumer"** can use the system for searching service, review and rate service whereas **"CorporateConsumer"** can use the system to promote their service. So two subclasses are created **"Individual"** and **"Corporate Consumer"** and they have their own additional attributes and methods aside from the superclass **"Consumer"**.



Fig(4): "IndividualConsumer" and "CorporateConsumer" as subclasses of "Consumer" with suitable methods

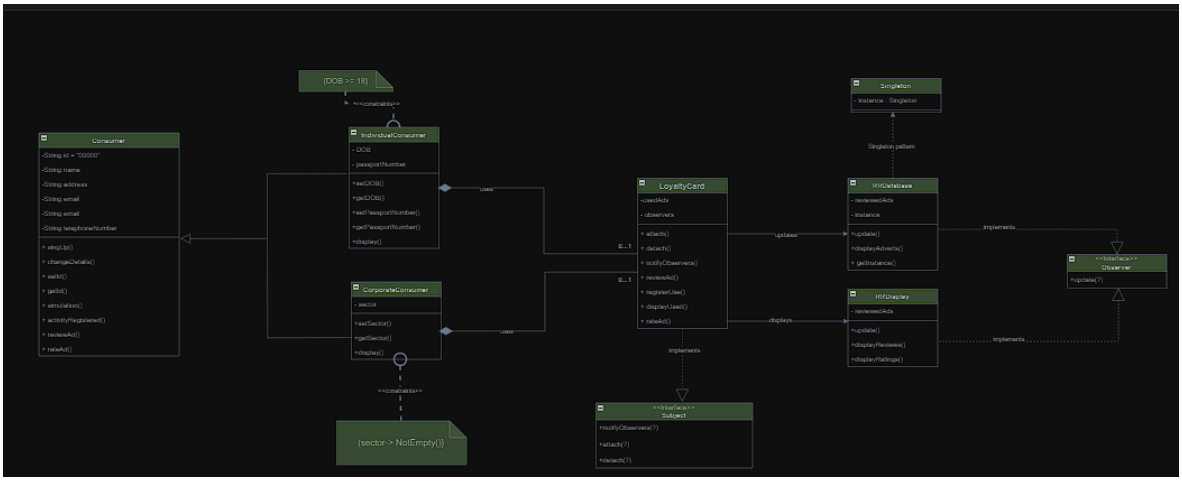
e) Additional Extension:

Different Constraints are added on different subclasses. Users must be or above 18 years of age are added in **"Individual"** whereas **"Corporate"** consumers must prove they belong to corporate by providing a sector.



Fig(5): Additional Extension

f) Overall rough Class Diagram:



Fig(6): Final Diagram for Extend the Provided Class Diagram

4. Extend the Implementation

a) Create **"ReviewDatabase"** as singleton:

```
public class RRDatabase implements Observer {
    private static RRDatabase instance;  // Singleton instance
    private ArrayList<Advertisement> reviewedAds;  // List of reviewed ads

    public RRDatabase() {
        reviewedAds = new ArrayList<Advertisement>();
    }

    // Singleton pattern: provide the instance
    public static RRDatabase getInstance() {
        if (instance == null) {
            instance = new RRDatabase();
        }
        return instance;
    }

    // Implement the update method from the Observer interface
    @Override
    public void update(Advertisement ad) {
        if (!reviewedAds.contains(ad)) {
            reviewedAds.add(ad);  // Add the reviewed ad
        }
    }
```

Fig(7): ReviewDatabase as Singleton

b) In **"SignUp()"**, the system should ask the consumer to identify as an individual or corporate consumer. The appropriate object type should then be created.

```java
public void signUp() {
    Scanner scanner = new Scanner(System.in);

    System.out.println("*****************************************");
    System.out.println("Sign up as: ");
    System.out.println("1. Individual Consumer");
    System.out.println("2. Corporate Consumer");
    System.out.println("*****************************************");

    int choice = scanner.nextInt();
    scanner.nextLine(); // Clear the newline left after nextInt()

    if (choice == 1) {
        // Create an IndividualConsumer
        IndividualConsumer individualConsumer = new IndividualConsumer(adBank);

        // Collect details specific to an IndividualConsumer
        System.out.println("Enter Date of Birth (DOB) in format YYYY-MM-DD: ");
        String dob = scanner.nextLine(); // Get Date of Birth
        LocalDate birthDate = LocalDate.parse(dob, DateTimeFormatter.ISO_LOCAL_DATE);
        LocalDate today = LocalDate.now();

        // Calculate the age
        Period age = Period.between(birthDate, today);

        if (age.getYears() >= 18) {
            individualConsumer.setDOB(dob);
            System.out.println("Enter Passport Number: ");
            String ppNum = scanner.nextLine();  // Get Passport Number
            individualConsumer.setPassportNumber(ppNum);
        } else {
            System.out.println("You must be over 18 to register.");
            signUp(); // Re-prompt the user to sign up
            return;
        }
```

```
    } else if (choice == 2) {
        // Create a CorporateConsumer
        CorporateConsumer corporateConsumer = new CorporateConsumer(adBank);

        // Collect details specific to a CorporateConsumer
        System.out.println("Enter Company Sector: ");
        String sector = scanner.nextLine();

        if (!sector.isEmpty()) {
            corporateConsumer.setSector(sector);
        } else {
            System.out.println("Company sector is required.");
            signUp(); // Re-prompt the user to sign up
            return;
        }

    } else {
        System.out.println("Invalid choice! Please choose 1 or 2.");
        signUp(); // Recursively prompt again for valid input
        return;
    }

    // Get common details for both types of consumers
    name = "John";
    telephoneNumber = "07384983844";
    email= "johnK@bcu.ac.uk";
    address = "B42 2SU";
    display();

}
```

Fig(8): Updated SignUp() Method

c) After the consumer reviews or rates an advertisement the loyaltyCard should update rrDatabase(display reviewed/rated advertisements) and rrDisplay(store all reviewed/rated advertisements) in accordance with the Observer pattern.

```
public void reviewAd(String title) {
    System.out.println("*****************************************");
    System.out.println("The consumer reviews an advertisement");
    System.out.println("*****************************************");
    // Review an advertisement
    loyaltyCard.reviewAd(title);
}

public void rateAd(String title) {
    System.out.println("*****************************************");
    System.out.println("The consumer rates an advertisement");
    System.out.println("*****************************************");
    // Rate an advertisement
    loyaltyCard.rateAd(title);
}
```

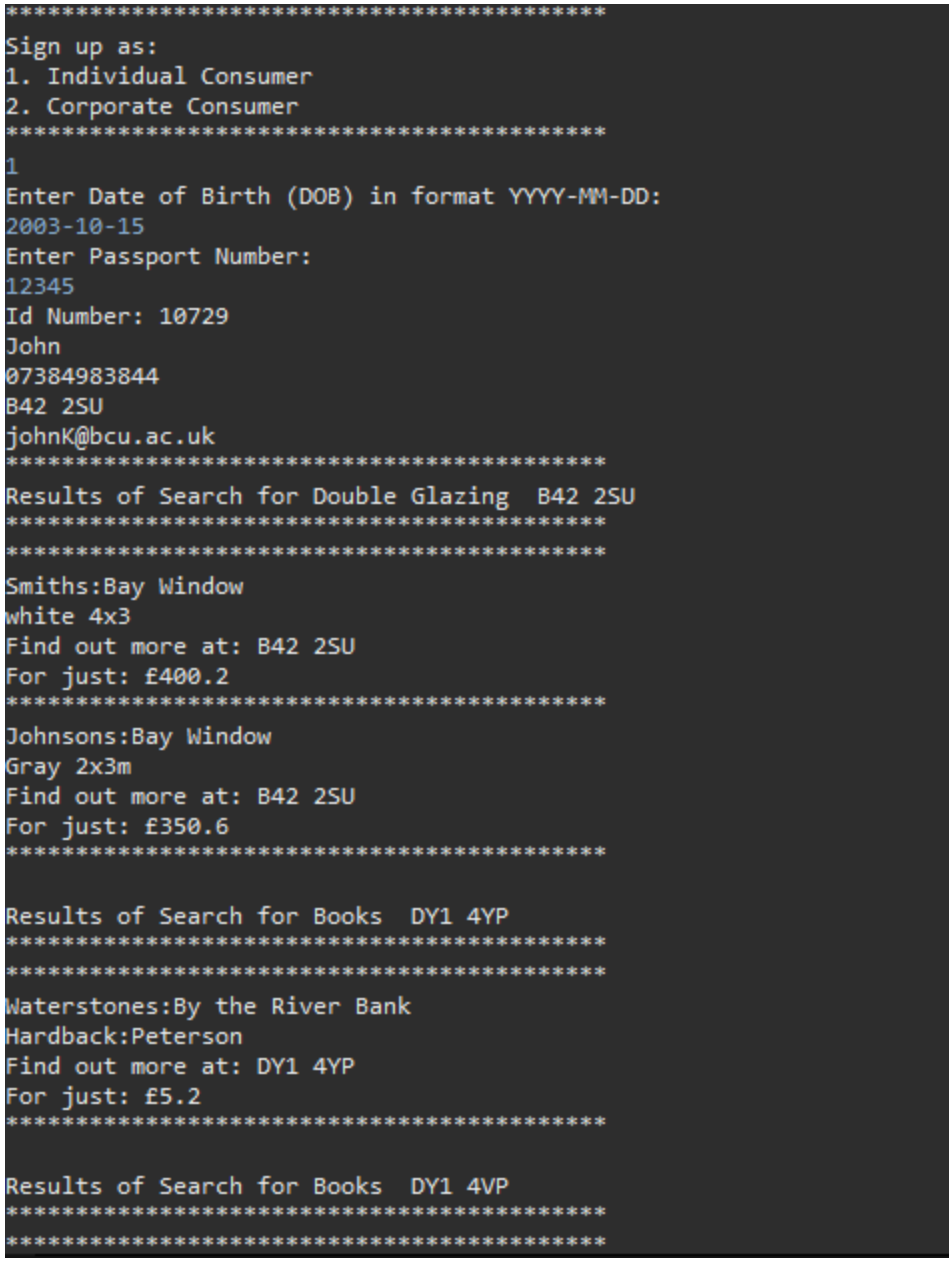
Fig(9): Review or Rates accordance with the Observer pattern

5. Produce Output Trace

a) The customer is an individual consumer:
b) The details of the consumer are printed:
c) The consumer searches:
   - (”Double Glazing”,”B42 2SU”)
   - (”Books”,”DY1 4YP”)
   - (”Books”,”DY1 4VP”)
d) The consumer displays all searches:
e) Activity is registered on:
   - “Smiths:Bay Window”
   - “Johnsons:Bay Window”
f) The consumer displays all reviewable/ratable advertisements:
g) The consumer selects the following to review:
   - Smiths:Bay Window”
   - “B&Q:Hammer”
h) The consumer displays all reviews from rrDisplay:
i) The consumer selects the following to rate:
   - “Smiths:Bay Window”
   - “Screwfix:Screw Driver”
j) The consumer displays Reviews from rrDisplay:
k) The reviewed/rated advertisements are printed from rrDattabase:

```
public void simulation() {
    // search
    searchSectorByLocation("Double Glazing", "B42 2SU");
    searchSectorByLocation("Books", "DY1 4YP");
    searchSectorByLocation("Books","DY1 4VP");


    // display all results
    displaySearch.displayAllActivity(activity);

    // some advertisements are used
    activityRegistered();

    // The consumer displays reviewable/ratable advertisements
    loyaltyCard.displayUsed();

    // Review an advertisement and display
    reviewAd("Smiths:Bay Window");
    reviewAd("B&Q:Hammer");

    // Display Reviews from Database
    rrDisplay.displayReviews();

    // Rate an advertisement and display
    rateAd("Smiths:Bay Window");
    rateAd("Screwfix:Screw Driver");

    // Display Reviews from Database
    rrDisplay.displayRatings();

    // Dump database
    rrDatabase.displayAdverts();
}
```

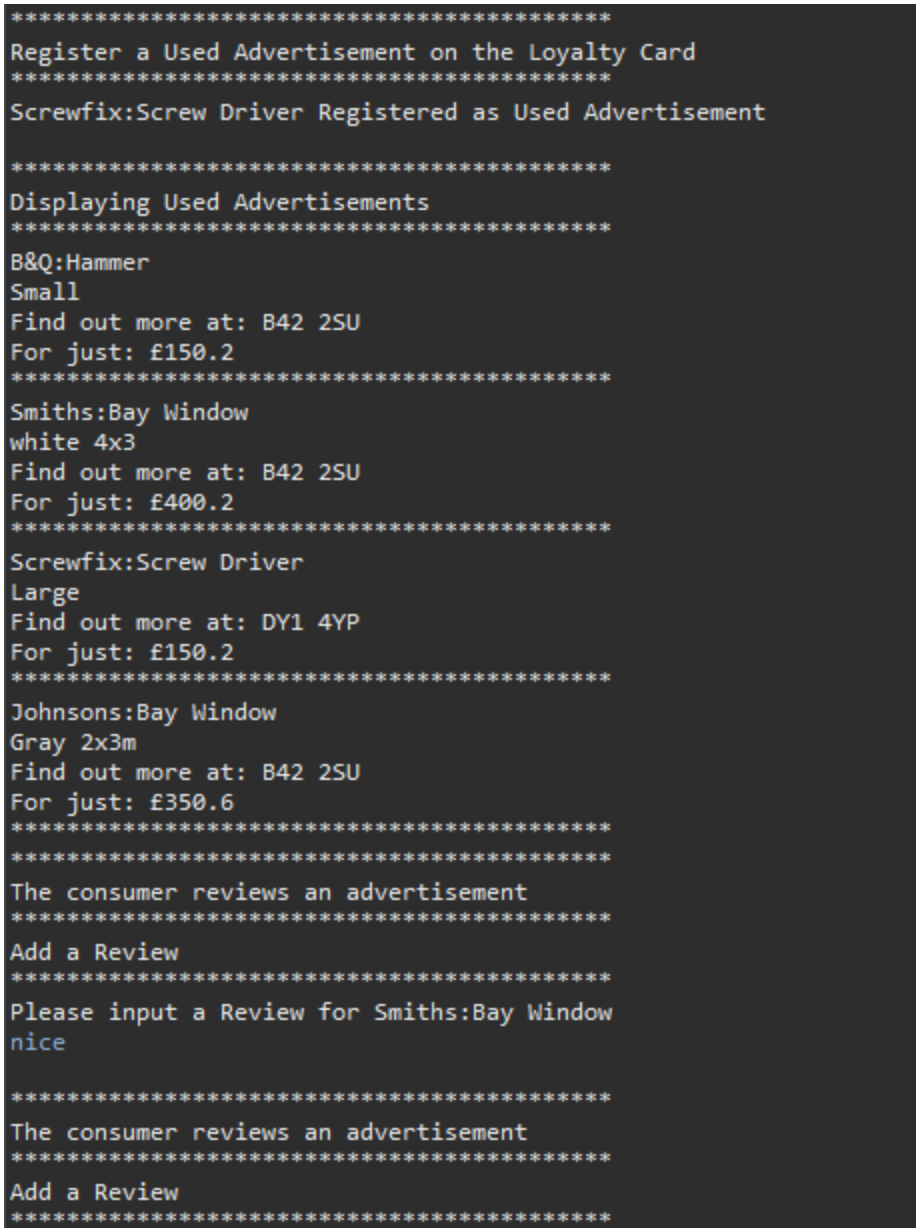
Fig(10): Java Implementation for Simulation Method



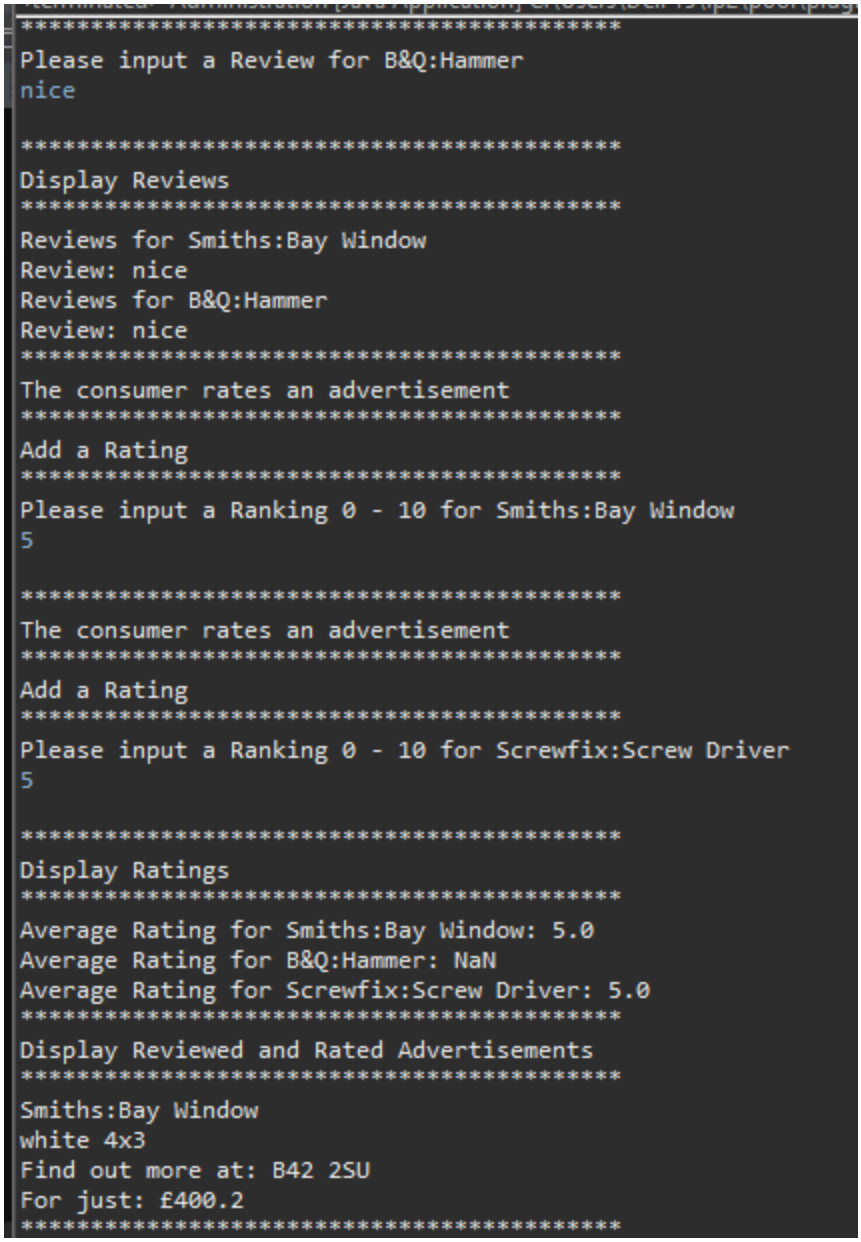
Fig(11): First Output for simulation()

```
Waterstones:By the River Bank
Hardback:Peterson
Find out more at: DY1 4YP
For just: £5.2
****************************************

All Search Results
****************************************
****************************************
Smiths:Bay Window
white 4x3
Find out more at: B42 2SU
For just: £400.2
****************************************
Johnsons:Bay Window
Gray 2x3m
Find out more at: B42 2SU
For just: £350.6
****************************************
Waterstones:By the River Bank
Hardback:Peterson
Find out more at: DY1 4YP
For just: £5.2
****************************************

****************************************
Register a Used Advertisement on the Loyalty Card
****************************************
Smiths:Bay Window Registered as Used Advertisement

****************************************
Register a Used Advertisement on the Loyalty Card
****************************************
Johnsons:Bay Window Registered as Used Advertisement

****************************************
Register a Used Advertisement on the Loyalty Card
****************************************
B&Q:Hammer Registered as Used Advertisement

****************************************
```

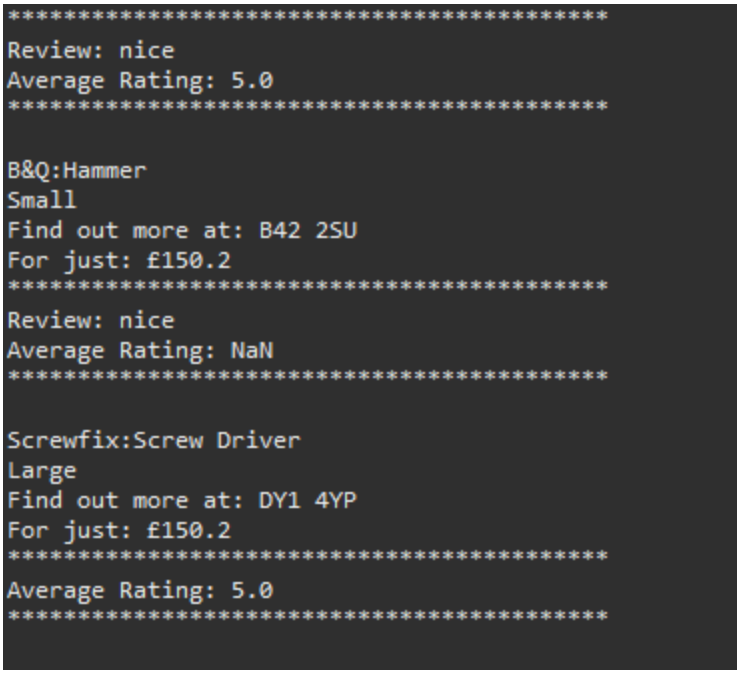
Fig(12): Second Output for simulation()



Fig(13): Third Output for simulation()



Fig(14): Fourth Output for simulation()

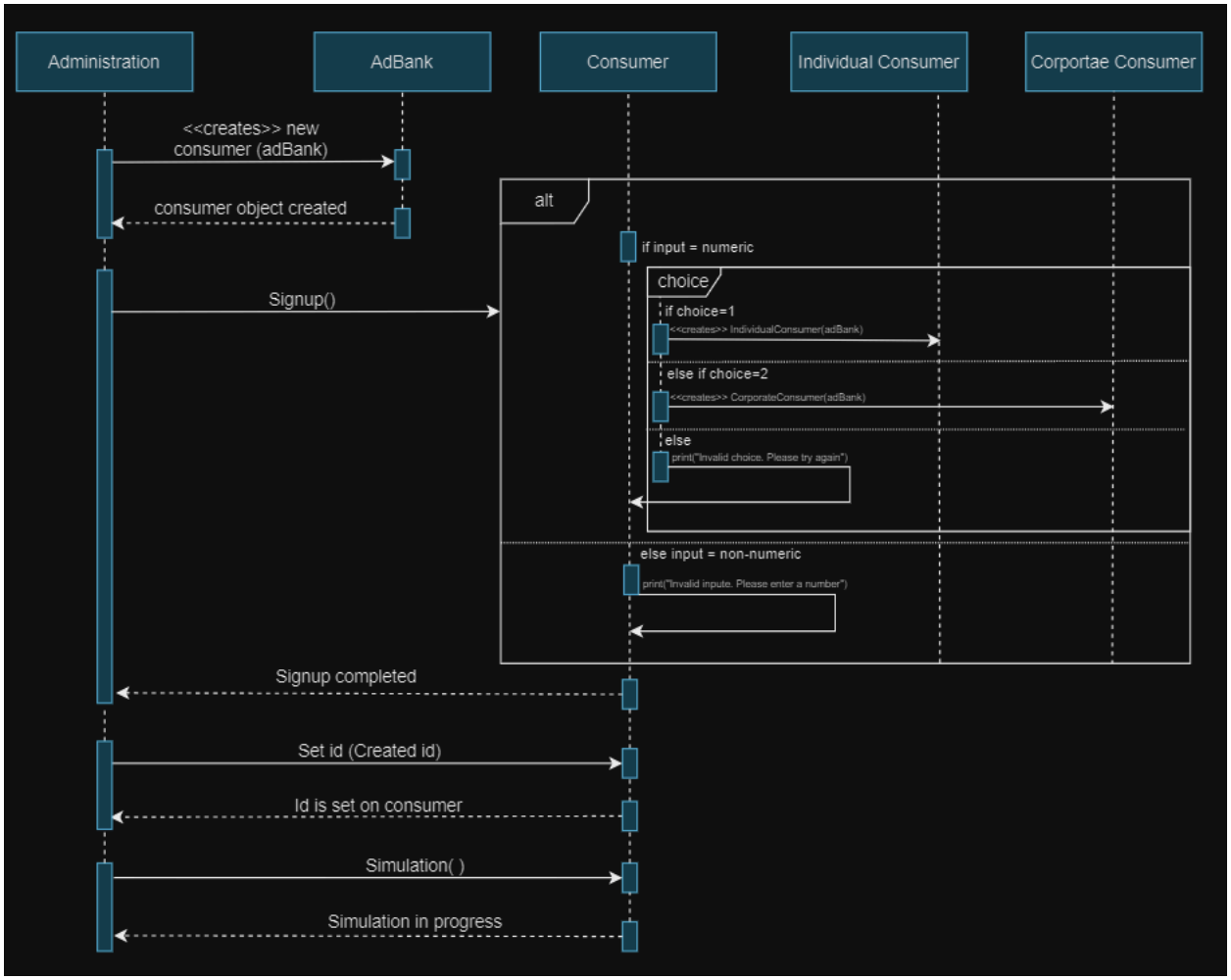


Fig(15): Fifth Output for simulation()

6. Produce Sequence Diagrams

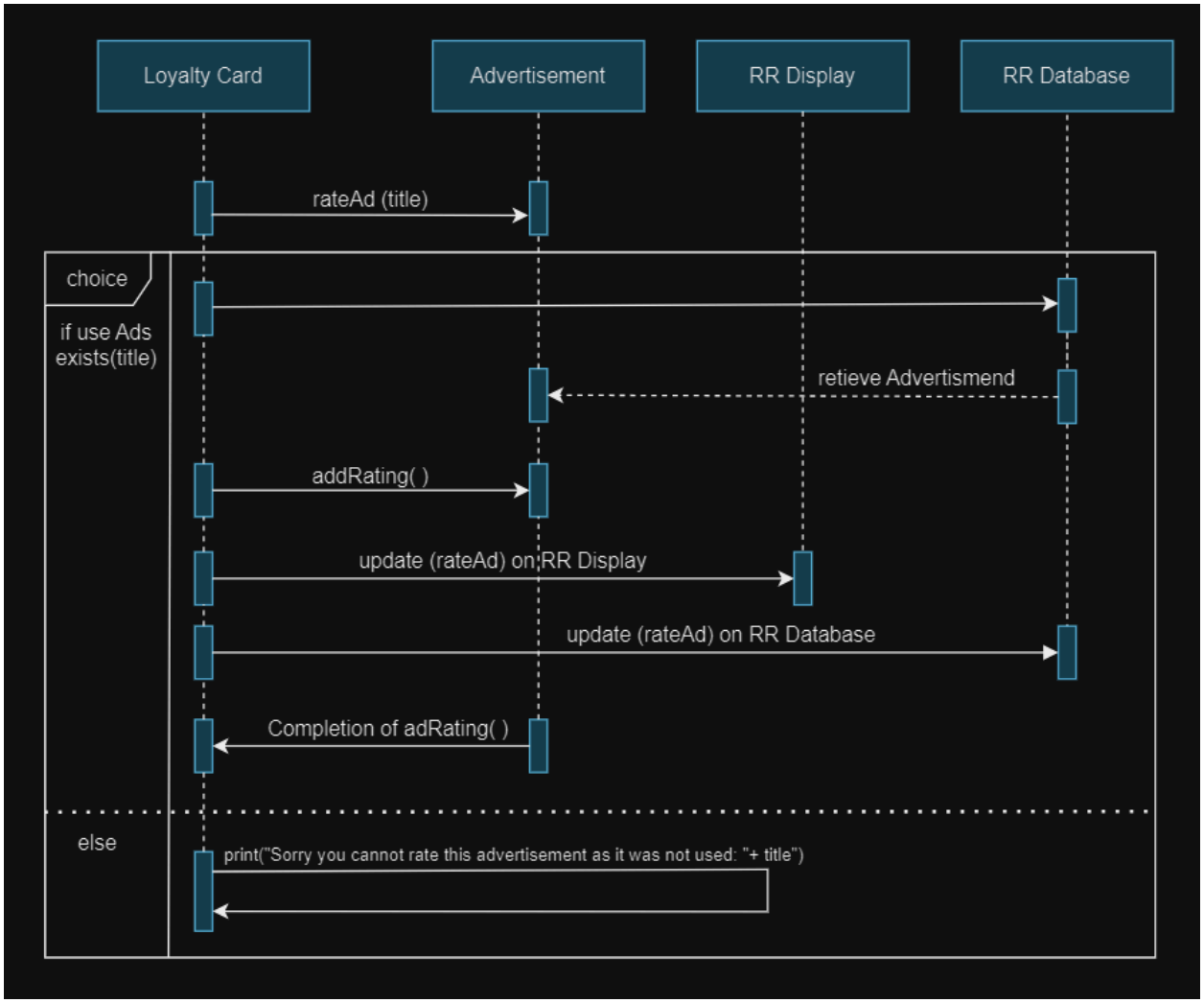
Draw the sequence diagrams for the following methods:

a) Class:Administration Method: SignUp(String title) - Include object interactions up to the call ”consumer.simulation()”. You do not need to show the sequence of calls inside the simulation() method.



Fig(16): Sequence diagram of "Administration"

b) Class:LoyaltyCard, Method; rateAd(String title):



Fig(17): Sequence diagram of "Loyaltycard"