ASSIGNMENT

CZ2002: Object-Oriented Design & Programming

Building an OO Application

Lab Group: FSP4, GROUP 5

Date of Submission: 15th April 2016

Version: 3.0

Student Name	Matriculation Number
Huang Jian Wei	U1521567A
Yong Guo Jun	U1440217C
Chua Boon Tat Prince	U1521245E
Leong Xinni	U1522224D
Sim Long Siang	U1522053H

SCHOOL OF COMPUTER ENGINEERING

NANYANG TECHNOLOGICAL UNIVERSITY

Declaration of Original Work for CE/CZ2002 Assignment

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 honored 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 (CE2002 or CZ2002)	Lab Group	Signature /Date
Huang Jian Wei	CZ2002	FSP4 (Group 5)	
Yong Guo Jun	CZ2002	FSP4 (Group 5)	
Chua Boon Tat Prince	CZ2002	FSP4 (Group 5)	
Leong Xinni	CZ2002	FSP4 (Group 5)	
Sim Long Siang	CZ2002	FSP4 (Group 5)	

Important notes:

1. Name must **EXACTLY MATCH** the one printed on your Matriculation Card.

Table of Contents

1. Introduction

The Hotel Reservation and Payment System (HRPS) is an application solely used by the hotel staff. It computerizes the processes of making hotel reservations, recordings of orders and displaying of records.

The report will cover the different design considerations and principles that were adopted while implementing HRPS. The class diagram and sequence diagram will be included to show the relationships and interactions between the classes and objects during runtime. It also aids the readers to better understand how HRPS flows as a whole. Test cases will also be provided to show what the validations covered, what the different inputs are from the user, and what are the results generated by the HRPS.

2. Design Considerations

2.1. Approach

HRPS is implemented using JAVA Object-Oriented programming language. Before the implementation phrase, the design phrase was carefully considered and one of the few objectives we came out with was to ensure that our application adopts the loose coupling (means that all classes should be independent as far as possible, in the case where changes have to be made to a class, the other classes would remain unchanged), and high cohesive (means that the class is focused on what it should be doing, relating to the intention of the class) principle. With that, anyone who takes on this application will be able to have easy testing, reusability, utmost extendibility and maintainability. Therefore, Model-View-Controller architecture was adopted and further illustrated as follows:

- Model – It encapsulates the core functionality and data, represented by the entity classes being tracked by the system.
- View – It implements the User Interface (UI), interaction between the user and the system, represented by the entity classes.
- Controller – It reacts to user input, encompassing the logic classes to meet the functional requirements of the system.

2.2. Design Principles

2.2.1. Single Responsibility Principle (SRP)

There should never be more than one reason for a class to change. For instance, in HRPS, the *CheckInController* class is only responsible for retrieving the room number and changing the availability status of the room accordingly. Meaning to say if *CheckIn* class were to be changed (extended), it would not implicate the other classes thus having high cohesiveness.

2.2.2. Liskov Substitution Principle (LSP)

This principle is fulfilled in HRPS when a derived class must be substitutable for their base class, whilst retaining the original behaviour of the class, provided if its pre-conditions are not stronger than the base class method (expect no more) and its post-conditions are no weaker than the base class method (provide no less). In this case, *Reservation* class is the derived class and the base class will be *WalkIn*. They shared most of the same variables (*roomNo*, *roomType*, *bedType*, *smoking*, *guestNationality*, *roomStartDate*, *roomEndDate*, *roomAvailability*, *guestCountry*, *wifi*, *numberOfGuest*, *roomView*, *guestName*, *guestAddress*, *guestContacts*, *guestGender*, *guestIdentity*, *guestCreditCardNo*, *guestCreditCardCSV*, *guestCreditCardExpDAte*, *checkInStatus*) and methods (*get* and *set*), and in order not to create the same variables again, inheritance is forged between these two classes.

2.2.3. Don’t Repeat Yourself Principle (DRY)

This principle states that every class must have a single, unambiguous authoritative representation within a system. The HRPS application avoids unnecessary duplication of codes and functionality, thus enabling the codes to be simpler and more maintainable. One example that uses the DRY principle is in the *ReservationMenu* Class whereby validateResID() method is used in various methods like searchForResNo(), searchForResIDPrint(), and updateResList(). With this implementation, it removes the need for the duplication of codes as validateResID() method can be reused.

2.3. Object-Oriented Concepts

Adopting proper object-orientated programming concepts is important in order to simplify development and maintenance of an application. Two object-oriented concepts are used in the coding of the HRPS application, namely Encapsulation/Information Hiding and Inheritance.

2.3.1. Encapsulation/Information Hiding

Encapsulation is the concept of building a barrier to protect an object’s private data. It allows the object’s classes (get and set methods) to be accessed through the public methods. Information hiding hides the details or implementation of the class from the users. One example of implementing encapsulation and information hiding concept in HRPS is the *Room* Class where it contains several private attributes like *roomType*, *roomNo*, *bedType*, *smoking*, *startDate*, *endDate*, *wifi*, *roomView*, *availability* and *checkInStatus*. These private fields mentioned are accessed and modified through the public *get* and *set* methods of the same class thus having full control of what is in stored in its private fields.

2.3.2. Inheritance

Inheritance is the process of defining a new class, also known as the subclass or derived class, which inherits the properties and behaviors (methods) of a parent class, known as a base class or superclass. It can create new classes without extensive duplication of code and can also reuse the parent's code which works exactly like the (DRY) principle mentioned under one of the design principles (2.2.3.). One such example that inheritance is utilized in HRPS implementation is the *Reservation* that is extended to *WalkIn* class.

2.4. Assumptions

There are some assumptions made during the development of HRPS application. The assumptions are made in order for users to know how far are they able to utilize the application if they were to adhere to them. The assumptions are stated below:

1. The hotel staff will need to key in the details of the 48 rooms with respect to the preferences of the guests as HRPS starts off with a clean state of 48 empty rooms.
2. The guest will strictly check out on the day of the end date stated by them.
3. The guest can extend their stay by updating the room end date.
4. The room availability status "Under Maintenance" can only happen in the process of updating of room details or before when the room is being checked in.
5. Each room number can only allow one reservation made regardless of the difference in time frame. It is taken into consideration that the duration of stay might be extended and priority goes to the guest who is already occupying it at that point in time. Therefore if we were to allow second reservation made to the same room, there will be complications that the guest has to swop to a new room (if there is any) and this would not please the guest.
6. The start date and end date of stay for updating of room details are validated to the exact calendar date (taken into consideration of leap years, exact number of days per month, and making sure end date of stay is later than start date of stay).
7. The contact number of the guest is assumed to be in 8 digits given the fact that if the guest is not of a local nationality, he/she is supposed to have purchase a local contact number.

8. The driving license or passport identity of the guest is assumed to be of 9 characters with the first and last characters being alphabets and the rest are all numbers given the fact that the guest might not be of a local nationality therefore not starting with "S".
9. The name, nationality, and country of the guest only allow alphabets to be entered.
10. The credit card details stated by the guest are assumed to be correct if they are of 16 digits of credit card numbers (no alphabets), 3 digits of CSV numbers (no alphabets) and a valid date (MM/YY).
11. The Room Service Menu item only accepts numerical values for updating of prices.
12. All hotel staff is entitled to the same level of authority in accessing HRPS application.

2.5. Further Enhancements

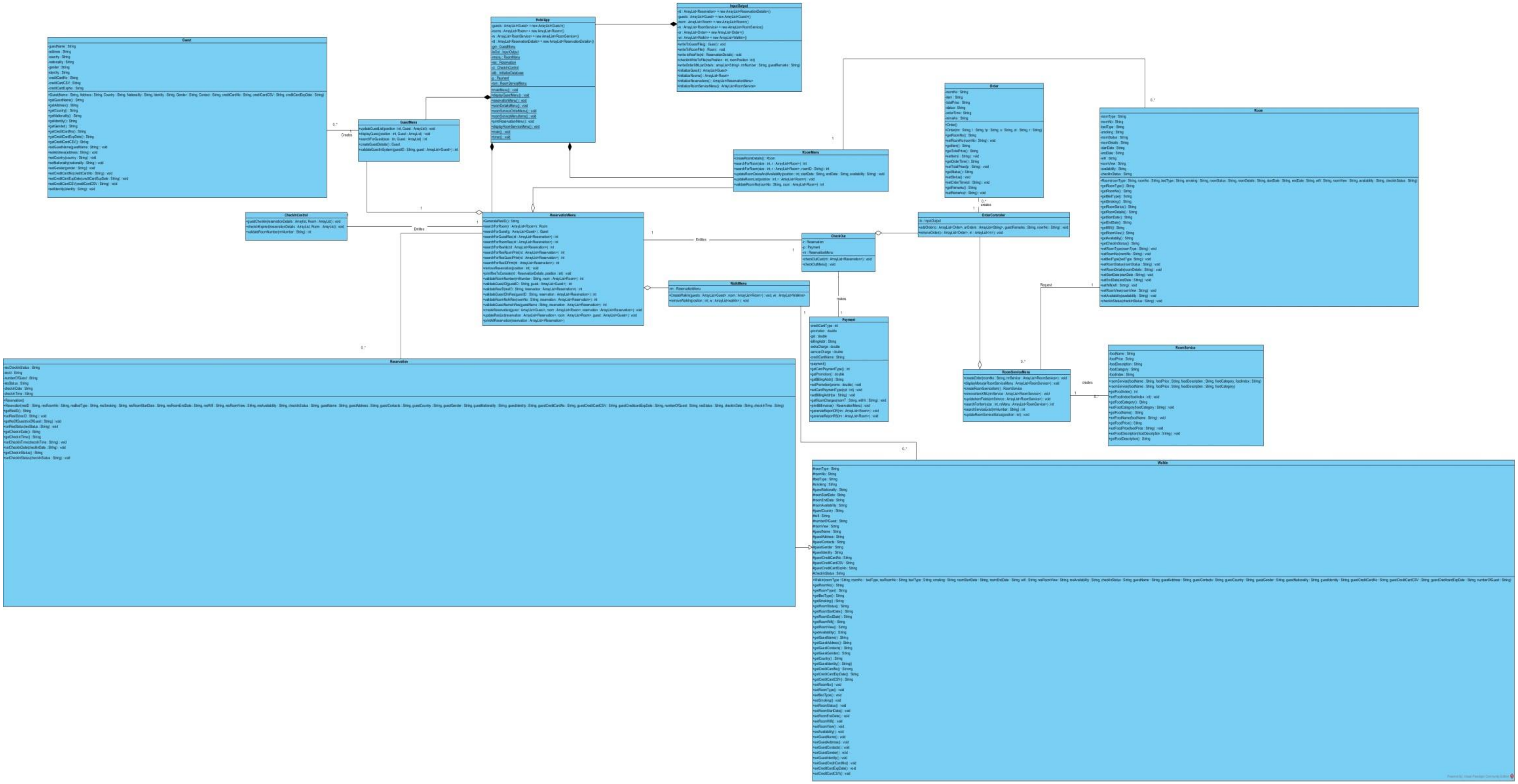
The considerable 2 new features for further enhancements are listed below:

1. HRPS will allow more than one reservation to be made to one room number so long as it is within a different time frame.
2. HRPS will store the payment made after checking out into a database to allow the hotel to make adequate analysis to the revenues generated.

There will be explanations listed below to illustrate how the current design of HRPS application can cater to the above 2 new features using the design principles adopted.

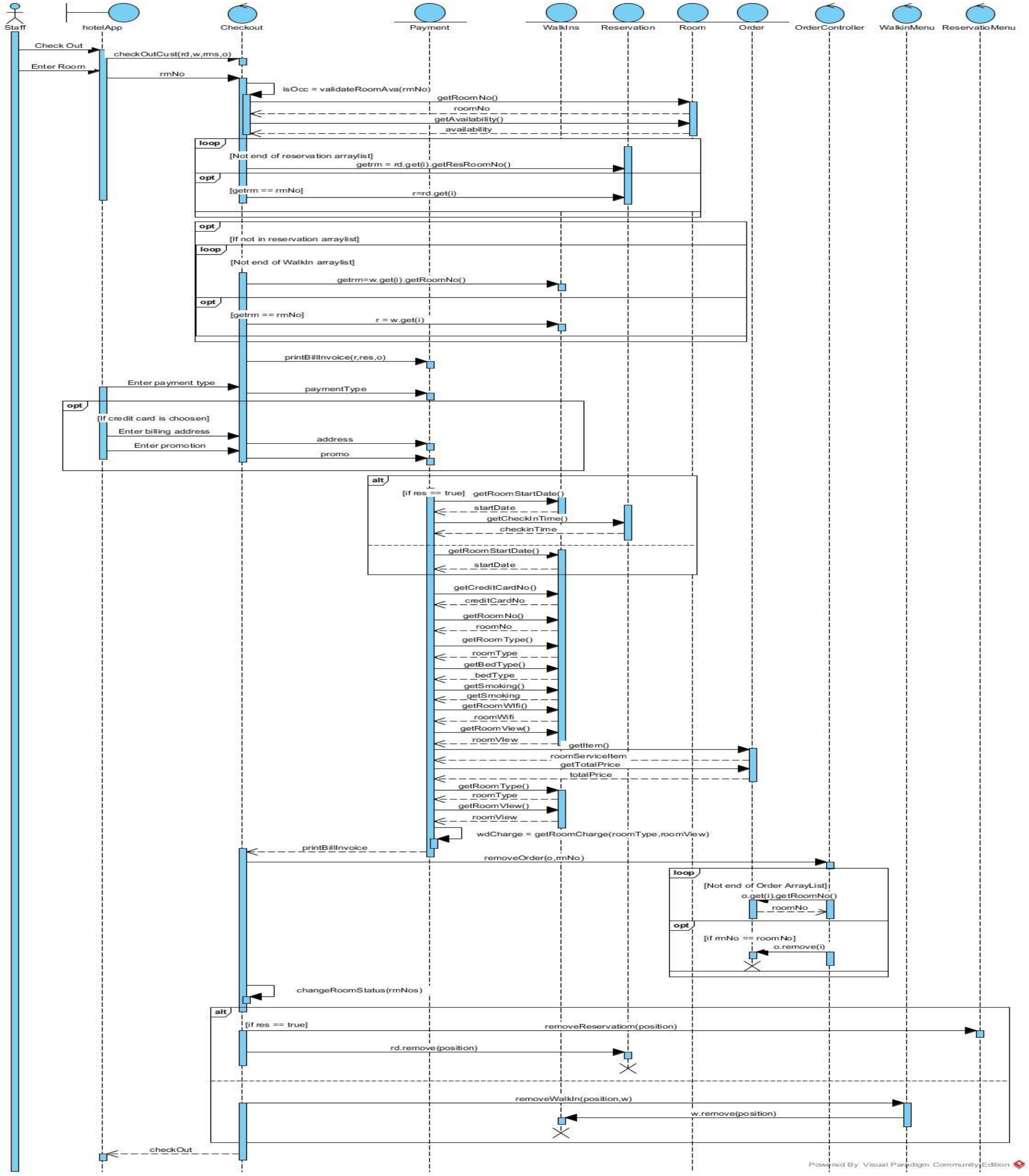
1. With reference to the new feature number (1), it shows that there is room for reusability and extendibility from the current function (Under Assumption number 4.) that exists in HRPS.
2. With reference to the new feature number (2), a "Receipt.xml" can be added so that the Hotel Staff can analyze from the past guests' receipts.

3. Class Diagram (Refer to PDF for clearer image)

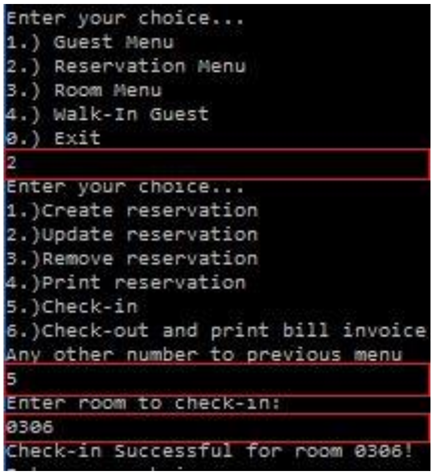


4. Sequence Diagram

(Refer to PDF for clearer image)

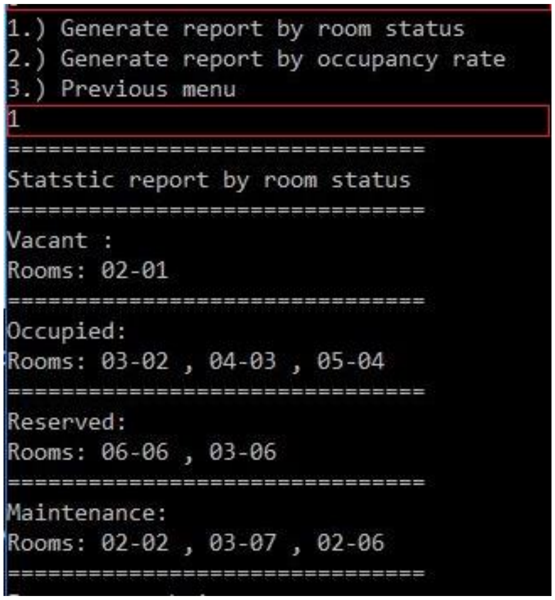


5. Test Cases



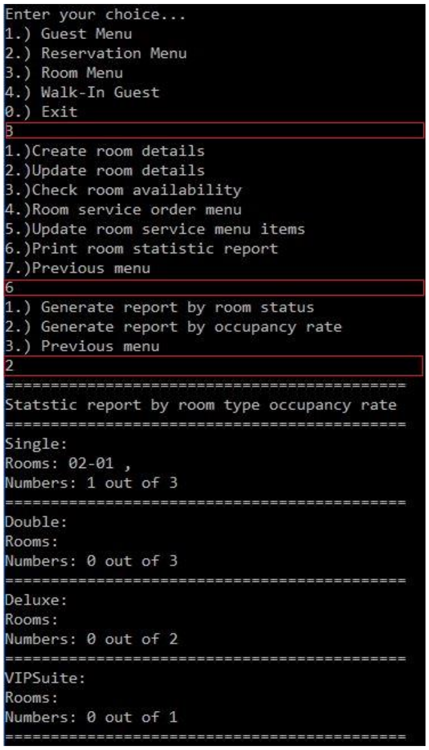
```
Enter your choice...
1.) Guest Menu
2.) Reservation Menu
3.) Room Menu
4.) Walk-In Guest
0.) Exit
2
Enter your choice...
1.)Create reservation
2.)Update reservation
3.)Remove reservation
4.)Print reservation
5.)Check-in
6.)Check-out and print bill invoice
Any other number to previous menu
5
Enter room to check-in:
0306
Check-in Successful for room 0306!
```

Figure1.1 – Checking in



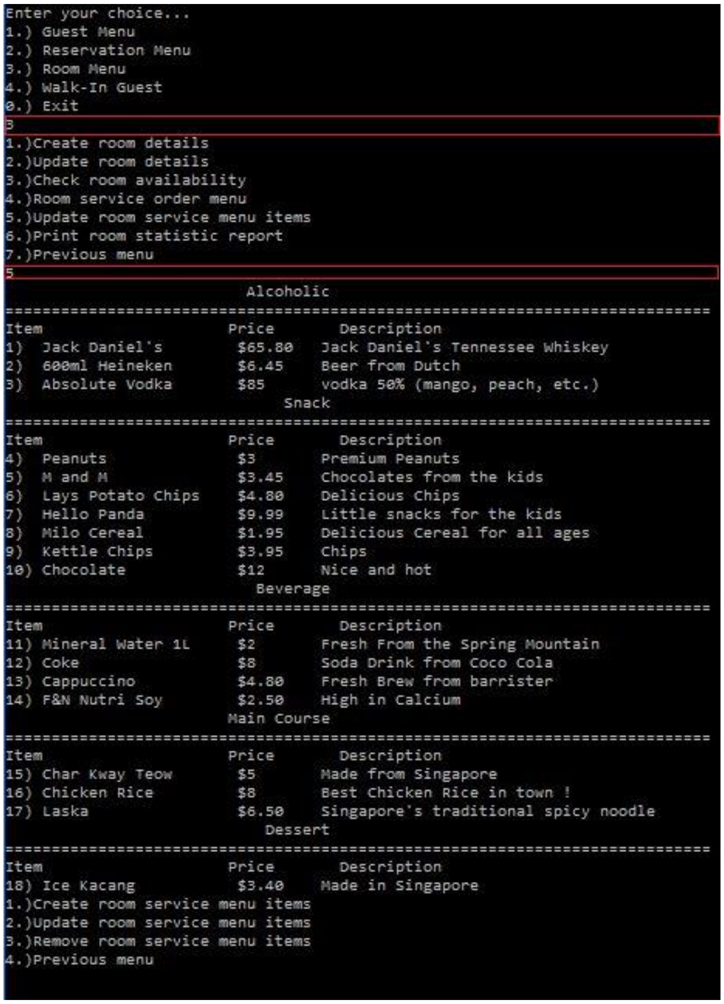
```
1.) Generate report by room status
2.) Generate report by occupancy rate
3.) Previous menu
1
==============================
Statstic report by room status
==============================
Vacant :
Rooms: 02-01
==============================
Occupied:
Rooms: 03-02 , 04-03 , 05-04
==============================
Reserved:
Rooms: 06-06 , 03-06
==============================
Maintenance:
Rooms: 02-02 , 03-07 , 02-06
==============================
```

Figure1.2 – Checking room availability by room statistical report (room status)



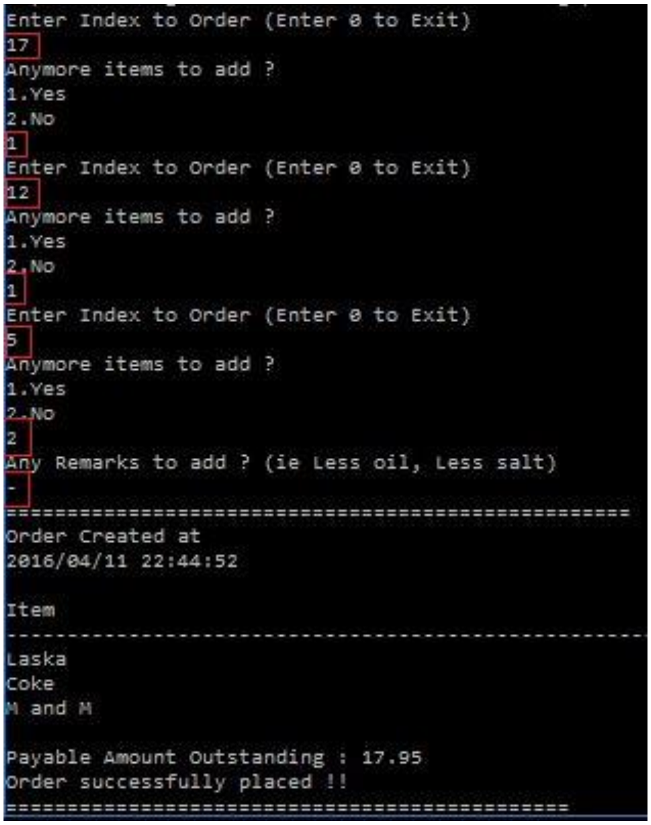
```
Enter your choice...
1.) Guest Menu
2.) Reservation Menu
3.) Room Menu
4.) Walk-In Guest
0.) Exit
3
1.)Create room details
2.)Update room details
3.)Check room availability
4.)Room service order menu
5.)Update room service menu items
6.)Print room statistic report
7.)Previous menu
6
1.) Generate report by room status
2.) Generate report by occupancy rate
3.) Previous menu
2
=========================================
Statstic report by room type occupancy rate
=========================================
Single:
Rooms: 02-01 ,
Numbers: 1 out of 3
=========================================
Double:
Rooms:
Numbers: 0 out of 3
=========================================
Deluxe:
Rooms:
Numbers: 0 out of 2
=========================================
VIPSuite:
Rooms:
Numbers: 0 out of 1
=========================================
```

Figure1.3 – Checking room details by room statistical report (room type occupancy status)



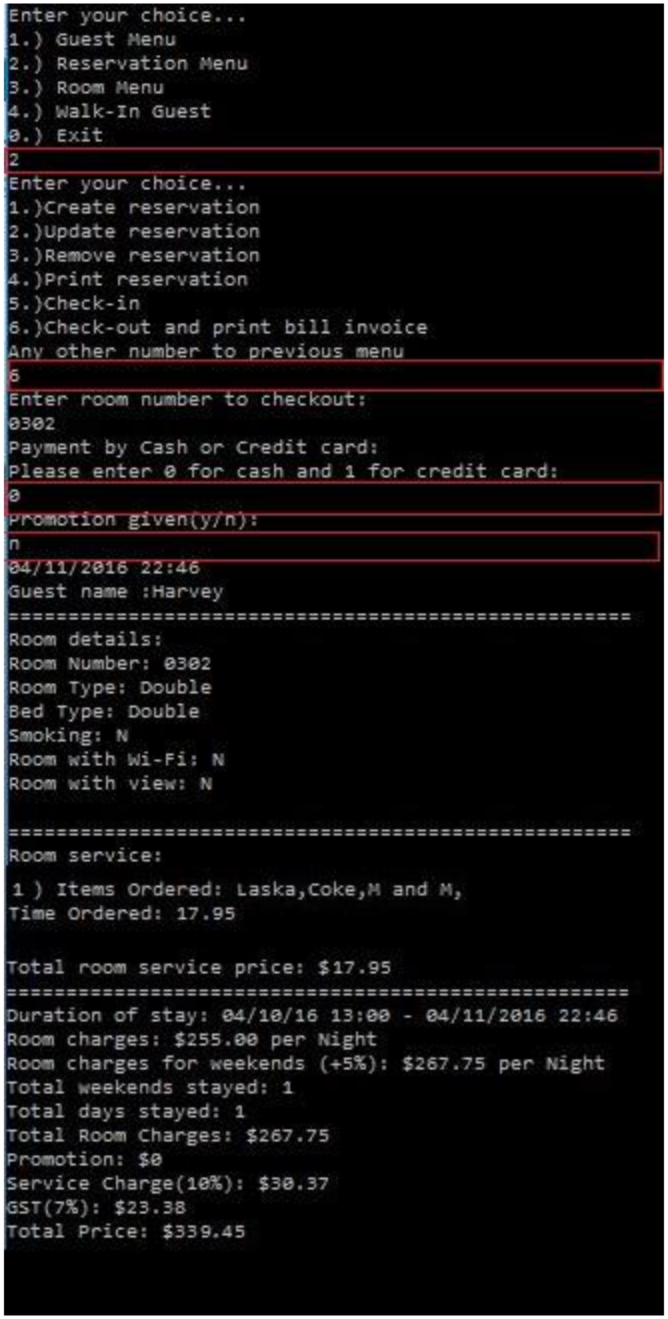
```
Enter your choice...
1.) Guest Menu
2.) Reservation Menu
3.) Room Menu
4.) Walk-In Guest
0.) Exit
3
1.)Create room details
2.)Update room details
3.)Check room availability
4.)Room service order menu
5.)Update room service menu items
6.)Print room statistic report
7.)Previous menu
5
                          Alcoholic
============================================================================
Item                    Price       Description
1)  Jack Daniel's        $65.80   Jack Daniel's Tennessee Whiskey
2)  600ml Heineken       $6.45    Beer from Dutch
3)  Absolute Vodka       $85      vodka 50% (mango, peach, etc.)
                              Snack
============================================================================
Item                    Price       Description
4)  Peanuts              $3       Premium Peanuts
5)  M and M              $3.45    Chocolates from the kids
6)  Lays Potato Chips    $4.80    Delicious Chips
7)  Hello Panda          $9.99    Little snacks for the kids
8)  Milo Cereal          $1.95    Delicious Cereal for all ages
9)  Kettle Chips         $3.95    Chips
10) Chocolate            $12      Nice and hot
                           Beverage
============================================================================
Item                    Price       Description
11) Mineral Water 1L     $2       Fresh From the Spring Mountain
12) Coke                 $8       Soda Drink from Coco Cola
13) Cappuccino           $4.80    Fresh Brew from barrister
14) F&N Nutri Soy        $2.50    High in Calcium
                        Main Course
============================================================================
Item                    Price       Description
15) Char Kway Teow       $5       Made from Singapore
16) Chicken Rice         $8       Best Chicken Rice in town !
17) Laska                $6.50    Singapore's traditional spicy noodle
                            Dessert
============================================================================
Item                    Price       Description
18) Ice Kacang           $3.40    Made in Singapore
1.)Create room service menu items
2.)Update room service menu items
3.)Remove room service menu items
4.)Previous menu
```

Figure1.4 – Displaying of room service menu



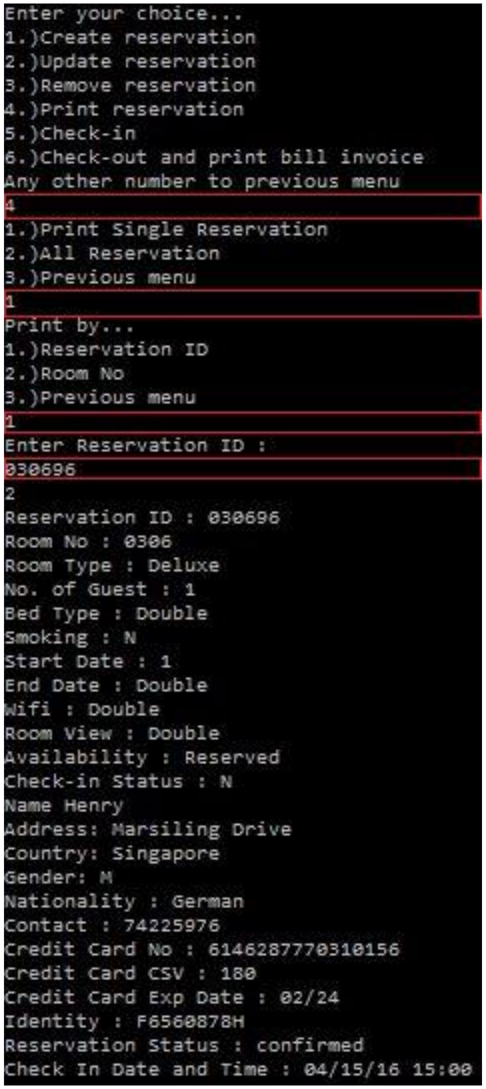
```
Enter Index to Order (Enter 0 to Exit)
17
Anymore items to add ?
1.Yes
2.No
1
Enter Index to Order (Enter 0 to Exit)
12
Anymore items to add ?
1.Yes
2.No
1
Enter Index to Order (Enter 0 to Exit)
5
Anymore items to add ?
1.Yes
2.No
2
Any Remarks to add ? (ie Less oil, Less salt)
-
==================================================
Order Created at
2016/04/11 22:44:52

Item
--------------------------------------------------
Laska
Coke
M and M

Payable Amount Outstanding : 17.95
Order successfully placed !!
================================================
```

Figure1.5 – Ordering of room service menu items



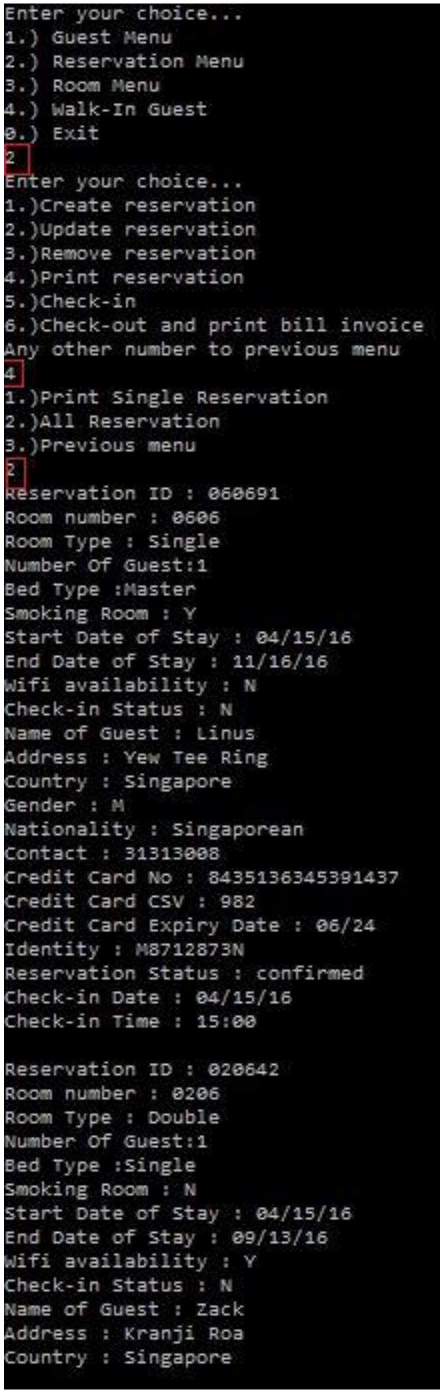
```
Enter your choice...
1.) Guest Menu
2.) Reservation Menu
3.) Room Menu
4.) Walk-In Guest
0.) Exit
2
Enter your choice...
1.)Create reservation
2.)Update reservation
3.)Remove reservation
4.)Print reservation
5.)Check-in
6.)Check-out and print bill invoice
Any other number to previous menu
6
Enter room number to checkout:
0302
Payment by Cash or Credit card:
Please enter 0 for cash and 1 for credit card:
0
Promotion given(y/n):
n
04/11/2016 22:46
Guest name :Harvey
====================================================
Room details:
Room Number: 0302
Room Type: Double
Bed Type: Double
Smoking: N
Room with Wi-Fi: N
Room with view: N

====================================================
Room service:
 1 ) Items Ordered: Laska,Coke,M and M,
Time Ordered: 17.95

Total room service price: $17.95
====================================================
Duration of stay: 04/10/16 13:00 - 04/11/2016 22:46
Room charges: $255.00 per Night
Room charges for weekends (+5%): $267.75 per Night
Total weekends stayed: 1
Total days stayed: 1
Total Room Charges: $267.75
Promotion: $0
Service Charge(10%): $30.37
GST(7%): $23.38
Total Price: $339.45
```

Figure1.6 – Checking out and printing bill invoice



```
Enter your choice...
1.)Create reservation
2.)Update reservation
3.)Remove reservation
4.)Print reservation
5.)Check-in
6.)Check-out and print bill invoice
Any other number to previous menu
4
1.)Print Single Reservation
2.)All Reservation
3.)Previous menu
1
Print by...
1.)Reservation ID
2.)Room No
3.)Previous menu
1
Enter Reservation ID :
030696
2
Reservation ID : 030696
Room No : 0306
Room Type : Deluxe
No. of Guest : 1
Bed Type : Double
Smoking : N
Start Date : 1
End Date : Double
Wifi : Double
Room View : Double
Availability : Reserved
Check-in Status : N
Name Henry
Address: Marsiling Drive
Country: Singapore
Gender: M
Nationality : German
Contact : 74225976
Credit Card No : 6146287770310156
Credit Card CSV : 180
Credit Card Exp Date : 02/24
Identity : F6560878H
Reservation Status : confirmed
Check In Date and Time : 04/15/16 15:00
```

Figure1.7 – Printing reservation(s) by ReservationID



```
Enter your choice...
1.) Guest Menu
2.) Reservation Menu
3.) Room Menu
4.) Walk-In Guest
0.) Exit
2
Enter your choice...
1.)Create reservation
2.)Update reservation
3.)Remove reservation
4.)Print reservation
5.)Check-in
6.)Check-out and print bill invoice
Any other number to previous menu
4
1.)Print Single Reservation
2.)All Reservation
3.)Previous menu
2
Reservation ID : 060691
Room number : 0606
Room Type : Single
Number Of Guest:1
Bed Type :Master
Smoking Room : Y
Start Date of Stay : 04/15/16
End Date of Stay : 11/16/16
Wifi availability : N
Check-in Status : N
Name of Guest : Linus
Address : Yew Tee Ring
Country : Singapore
Gender : M
Nationality : Singaporean
Contact : 31313008
Credit Card No : 8435136345391437
Credit Card CSV : 982
Credit Card Expiry Date : 06/24
Identity : M8712873N
Reservation Status : confirmed
Check-in Date : 04/15/16
Check-in Time : 15:00

Reservation ID : 020642
Room number : 0206
Room Type : Double
Number Of Guest:1
Bed Type :Single
Smoking Room : N
Start Date of Stay : 04/15/16
End Date of Stay : 09/13/16
Wifi availability : Y
Check-in Status : N
Name of Guest : Zack
Address : Kranji Roa
Country : Singapore
```

Figure1.8 – Printing all reservations