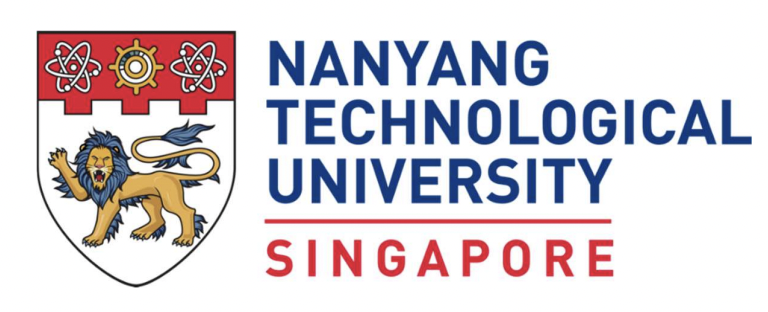
**NANYANG TECHNOLOGICAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

****

**CE2002 OBJECT-ORIENTED DESIGN AND PROGRAMMING**

***AY21/22 Semester 1 Group Assignment***

***Restaurant Reservation and Point of Sales System (RRPSS)***

**Lab Group: SE4**

**Group Number: 5**

| **Name of Group Members** | **Matriculation Number** |
| --- | --- |
| Loh Zhi Heng | U2022581B |
| Lee Cheow Teng | U2020098A |
| Oh Zhi Hua | U2021091A |
| Hoang Minh Nhat | U2020852F |

**Date of Submission: 14th November 2021**

**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/CZ2002)** | **Lab Group** | **Signature/Date** |
| --- | --- | --- | --- |
| Loh Zhi Heng | CE2002 | SE4 | 12/11/2021 |
| Lee Cheow Teng | CE2002 | SE4 | 12/11/2021 |
| Oh Zhi Hua | CE2002 | SE4 | 12/11/2021 |
| Hoang Minh Nhat | CE2002 | SE4 | 12/11/2021 |

# Table of Contents

[**Table of Contents**](#_tonyuhg7b4eo) **3**

[**Demonstration Link**](#_1fob9te) **3**

[**Design Considerations**](#_iqfz1iehl69j) **4**

[Approach Taken](#_3znysh7) 4

[Principles Used](#_ow8phc9vix81) 4

[Inheritance](#_ptgluqkfvnph) 4

[Interface/Abstract Class](#_quhqd0hmy0nq) 4

[Method Overriding](#_ftq2clz2ixog) 4

[Assumptions Made](#_hsc3j3wd8w6c) 5

[**UML Class Diagram**](#_dl4oxm35qfka) **7**

[**UML Sequence Diagram**](#_vwlzzmtz3w3x) **8**

[Deleting Expired Reservations](#_8980o6uxt9v9) 8

[Remove/Cancel An Existing Reservation](#_yvywj3g96ci3) 8

[Check Reservation](#_qr1fiv5ucsn8) 9

[Check-in with Reservation](#_f8548uo9c423) 9

[**Extra Test Cases**](#_lgu6p8aayovq) **10**

[General](#_6fjfaeol1gqc) 10

[Menu / Promotion Queries](#_yh3y4ivpbvep) 10

[Order Queries](#_wwg83827d1uf) 11

[Reservation Queries](#_ljrfnljvej8h) 12

[Print Sales Report Queries](#_6fj2u6ep9wrz) 13

# Demonstration Link

<https://youtu.be/_Kk1W9Kfxuc>

# Design Considerations

## Approach Taken

For the project, we opted to use the **Model-View-Controller (MVC) Framework**, which is often used for web-based applications. We can segregate UI logic, business logic, and entity level logic using the framework. Additionally, the ability to develop in parallel, combined with the framework's support for **Test Driven Development**, allows us to test and build separate components simultaneously.

## Principles Used

#### Inheritance

A promotion package is an **item** on the restaurant’s menu, hence the Promotion class extends the Item class.

Since both **Staff** and **Customers** are derived from the **abstraction of People**, they both extend the People class.

#### Interface/Abstract Class

**Item** class implements **IItem**, which is an **interface** class that provides the prototype methods that Item and Promotion implements.

**Staff** and **Customer** classes extend **People** class, which is an **abstract** class that provides base methods for both classes to implement and refine.

**PromotionController**, **CategoryController**, and **TableController**, all implement **ISearch**, which is an **interface** class that provides the prototype method that all three controllers implement.

#### **Method Overriding**

Since Promotion extends Item, it inherits methods of Item, which it can either **refine** or **reuse**.

The three controllers listed above implement **ISearch**, therefore they provide implementations for the method in ISearch.

## Assumptions Made

* There are 3 types of categories: Main, Sides, and Drinks.
  + New categories may be added, however not on the fly.
* There are 4 staff members in the restaurant with ID 0, 1, 2, and 3.
  + New staff members may be added via the data files.
* All of the tables are initially unoccupied when the restaurant (the app) is opened.
  + This is representative of the start of daily restaurant operations.
* Customer’s ID required when checking into the restaurant, else must register a new customer ID. When a new customer is registered, his/her membership is initially false.
* Our policy of allocating tables based on the number of pax:
  + For 1 or 2 pax: Only allocates tables of 2 or 4 seats.
  + For 3 or 4 pax: Only allocates tables of 4 or 6 seats.
  + For 5 or 6 pax: Only allocates tables of 6 or 8 seats.
  + For 7 or 8 pax: Only allocates tables of 8 or 10 seats.
  + For 9 or 10 pax: Only allocates tables of 10 seats.
* When a customer checks in, a table cannot be allocated if it is currently occupied or is reserved in the next 15 minutes.
* A reservation can only be made at least 2 hours in advance; a table can only be allocated if it is not reserved within 2 hours before and after the time of the reservation.
  + Please see below for the reason why we set the buffer to 2 hours.
* A customer will take at most 2 hours to finish the meal.
* Current orders placed prior to the update will not be updated if an item or promotion package is modified in the midst of business hours. This is to avoid customers getting an unpleasant awakening about the item they ordered due to a price difference between time of ordering and the time of payment.
  + A copy of the item or promotion object will be made and added to the table’s order attribute.
* A promotion package can have 1…\* items. This is to facilitate groups of different sizes.
  + E.g: Set for 2, has 2 mains. Set for 4, has 4 mains.
* Order is being added to reports based on **date/time of creation** rather than **date/time of checkout**
* Report generated is based on the current **day / month** request for generation being sent.
  + E.g. Request at 13th Nov
    - **Daily** will generate report for **13th Nov**
    - **Monthly** will generate report for **month of Nov**

# 

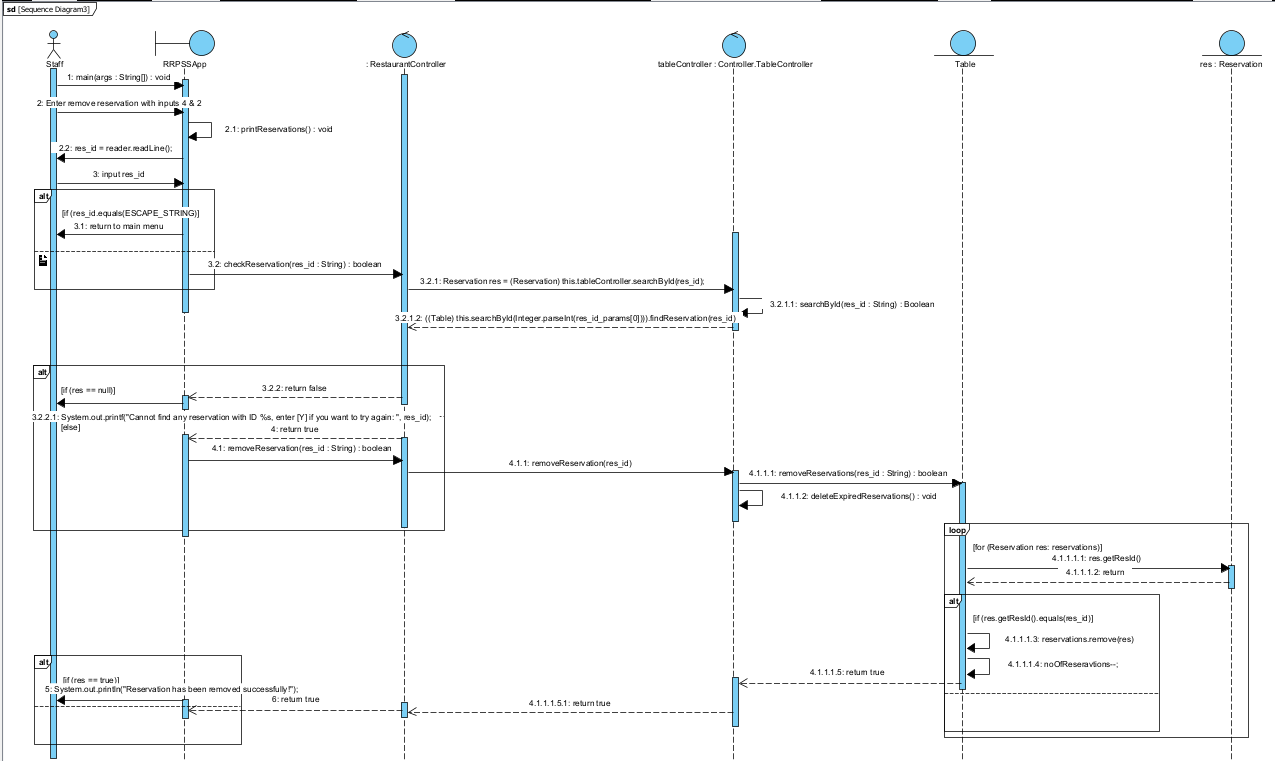
# UML Class Diagram

# 

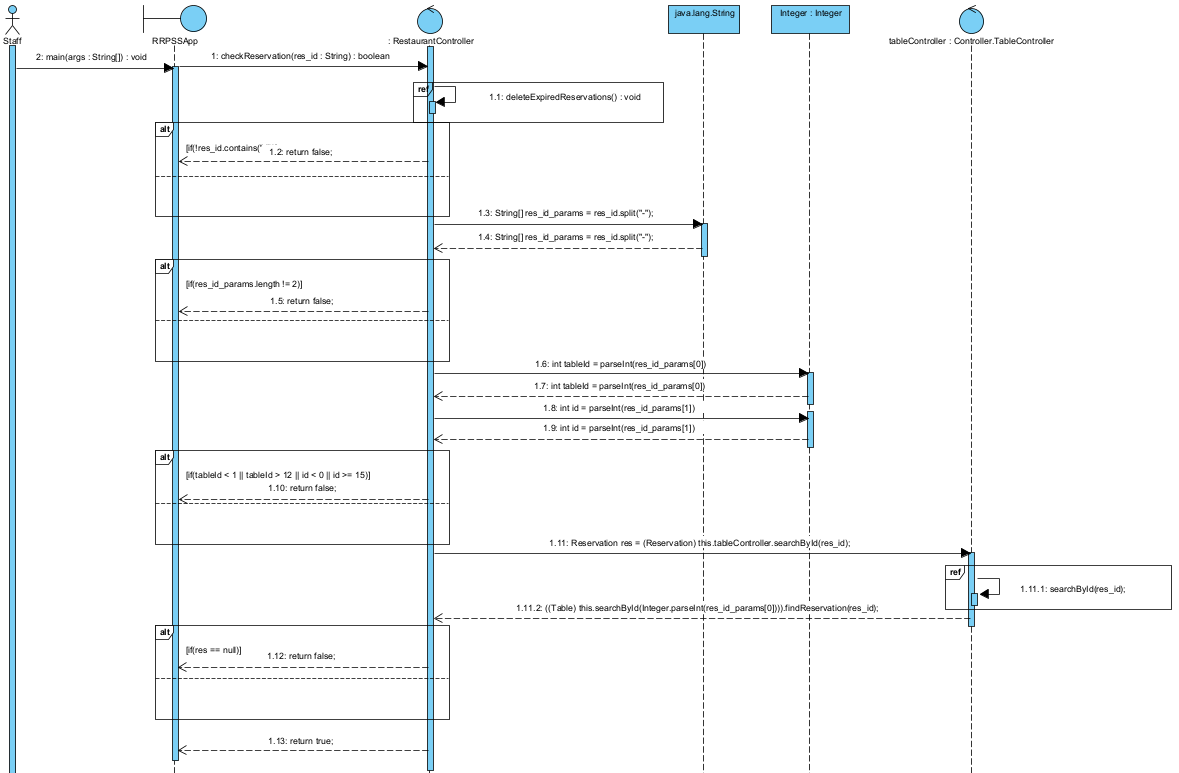
# UML Sequence Diagram

## Deleting Expired Reservations

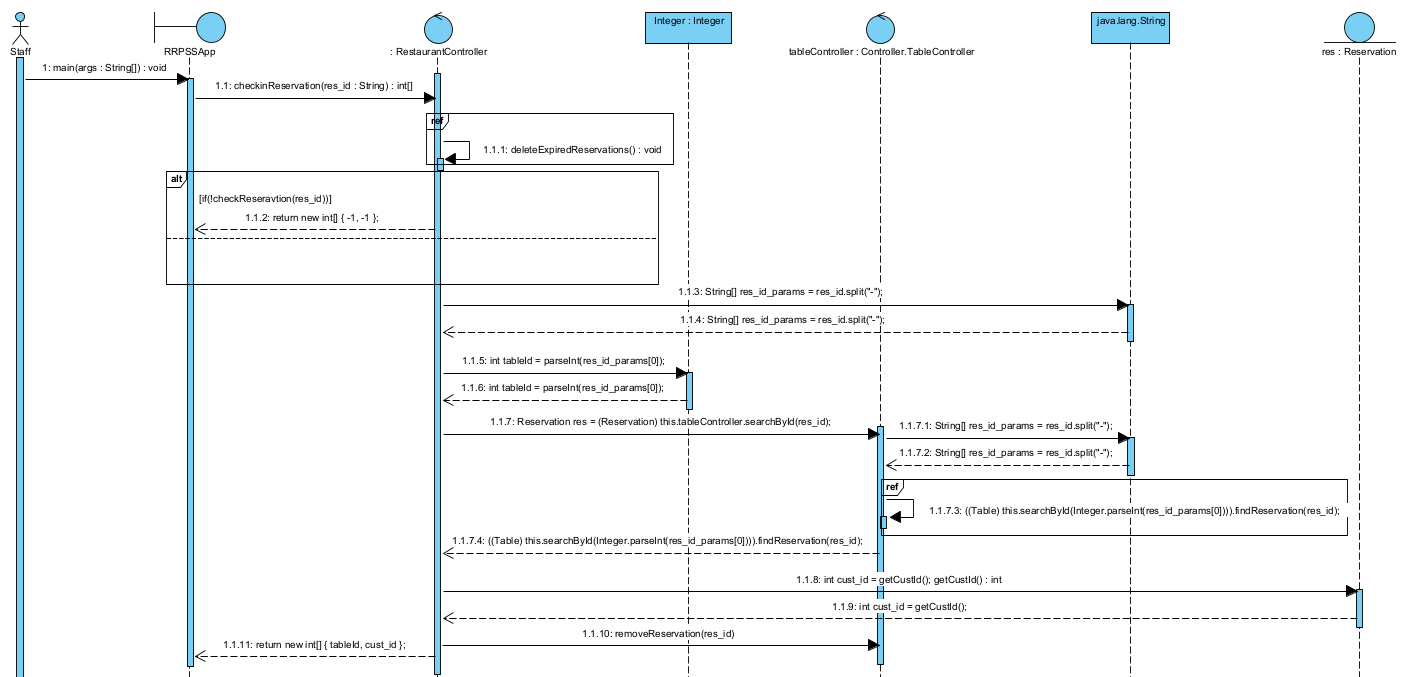
## Remove/Cancel An Existing Reservation



## Check Reservation



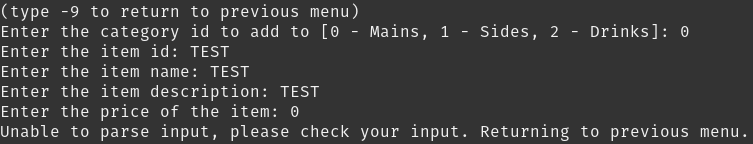
## Check-in with Reservation



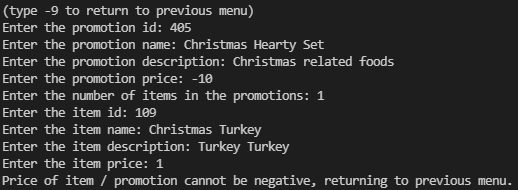
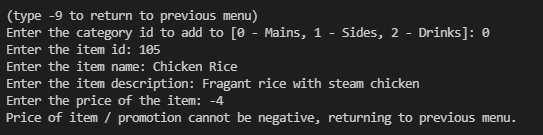
# Extra Test Cases

## General

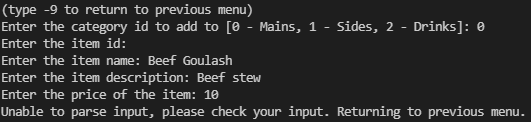
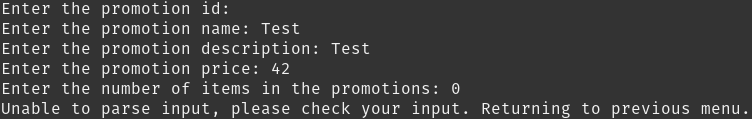
**Test Scenario 4.1.1: Input format mismatch**

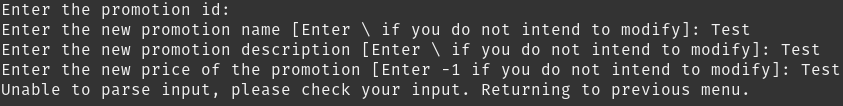
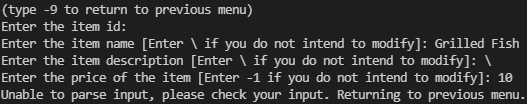
****

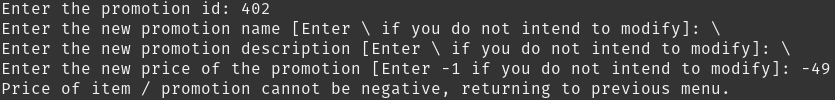
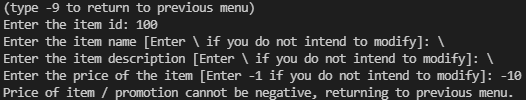
## Menu / Promotion Queries

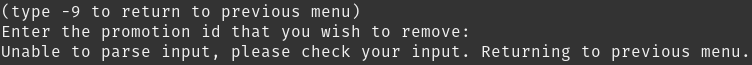
**Test Scenario 4.2.1: Add Item to menu / Promotion with negative price**

**Test Scenario 4.2.2: Add Item to menu / Promotion without ID**

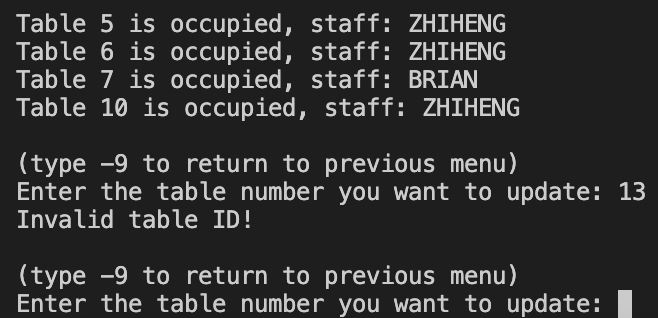
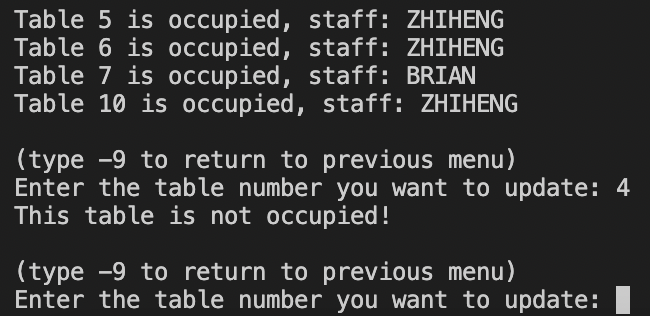
****

**Test Scenario 4.2.3: Update Item in menu / Promotion without ID**

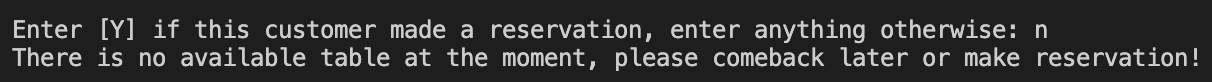
**Test Scenario 4.2.4: Update Item in menu / Promotion in menu with negative price**

**Test Scenario 4.2.5: Remove Item from menu / Promotion without ID******

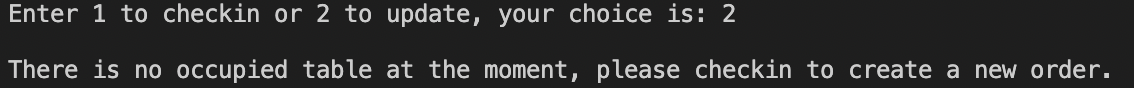
## Order Queries

**Test Scenario 4.3.1: Update a current order of an unoccupied/invalid table.**

**Test Scenario 4.3.2: Walk-in dining but all of the tables are either reserved or occupied.**

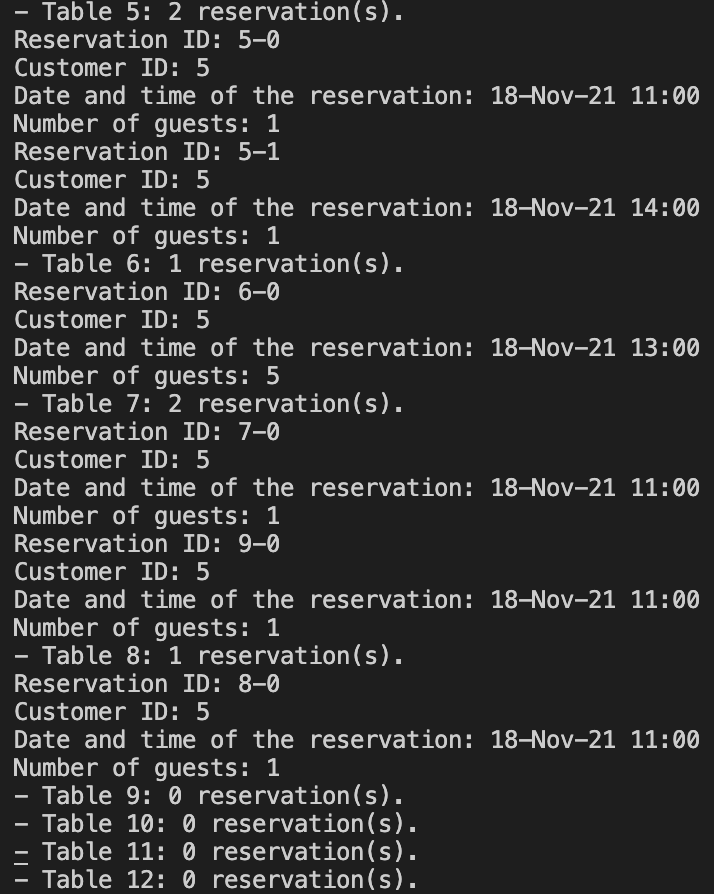
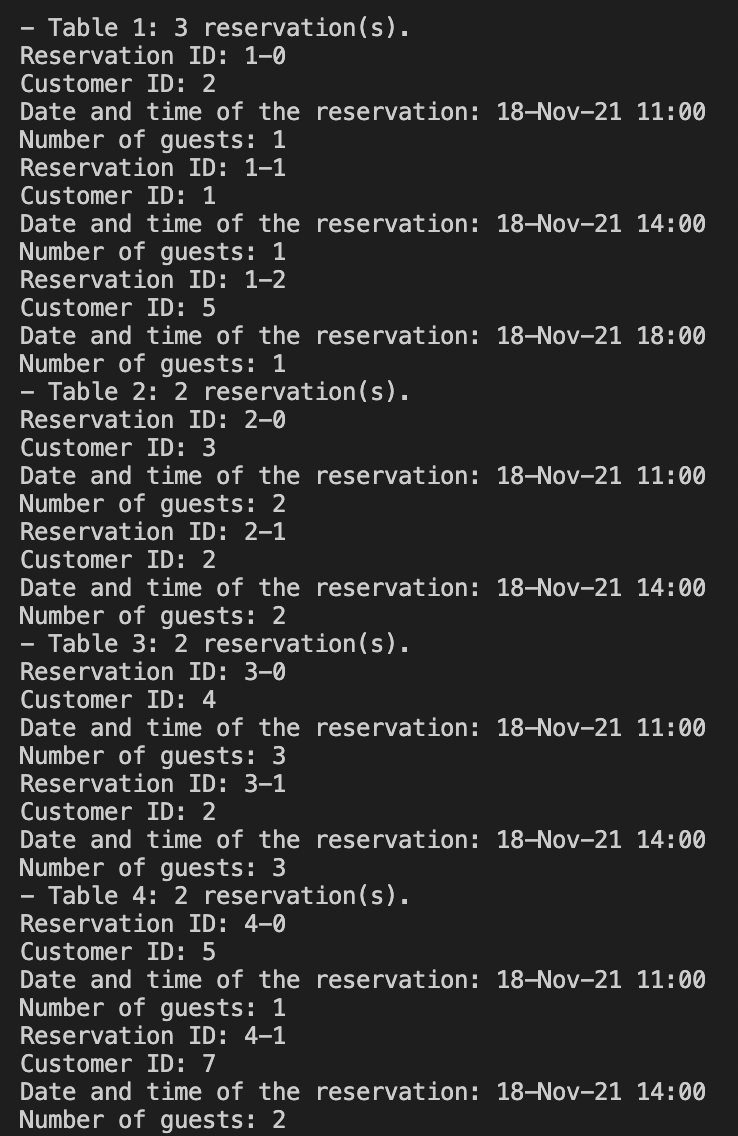
****

**Test Scenario 4.3.2: Try to update a current order but all of the tables are unoccupied.**

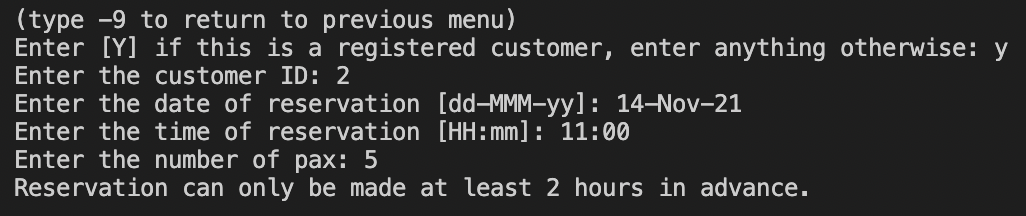
****

## Reservation Queries

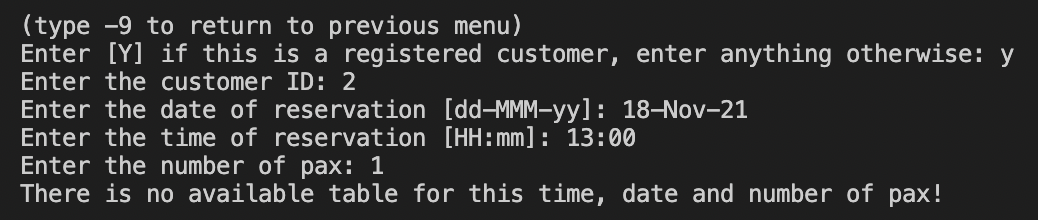
List of reservations for all of the tests below:



**Test Scenario 4.4.1: Input a time that is not at least 2 hours upon the current time, regardless of the number of pax.**

****

**Test Scenario 4.4.2: Input a time that is at least 2 hours, regardless of the number of pax, but there is no available table within 2 hours before and after the desired time.**

****

**Test Scenario 4.4.3: Update a reservation with a time that is not at least 2 hours upon the current time.**

Input: reservation ID (8-0), new time date (14-Nov-21 11:00).

Result: the same as the Test Scenario 4.5.1 and the reservation will remain unchanged.

**Test Scenario 4.4.4: Update a reservation time that is at least 2 hours upon the current time, but there is no available table within 2 hours before or after the desired time.**

Input: reservation ID (8-0), new time date (18-Nov-21 13:00).

Result: the same as the Test Scenario 4.5.2 and the reservation will remain unchanged.

**Test Scenario 4.4.5: Update a reservation with a new number of pax, which may allocate another table and may result in a clash.**

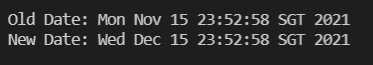
Input: reservation ID: 8-0, new number of pax: 2

Result: the same as the Test Scenario 4.5.2 and the reservation will remain unchanged.

## Print Sales Report Queries

**Test Scenario: Report generation for orders with differing creation and checkout date.**

1. Create order (Any generic order works).
2. Shift time by 1 month & checkout order.



1. Print daily or monthly report.

