

**FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY****AACS2204 OBJECT-ORIENTED PROGRAMMING TECHNIQUES****ASSIGNMENT****SEMESTER 202501**

| | | |
|--------------------------|---|------------------|
| Programme (Year & Group) | : | DCS1S3 |
| Tutor Name | : | CHONG KENT LOONG |
| Tutorial Group | : | GROUP 10 |
| Date Submitted | : | 02 May 2025 |

| No | Name | Student ID | Marks | | | Total Marks |
|----|----------------------|------------|-----------|-----------|-----------|-------------|
| | | | Section A | Section B | Section C | |
| 1 | CHONG WEN QI | 2410573 | | | | |
| 2 | WYNНИE WONG KAI XUAN | 2410779 | | | | |
| 3 | NG KAH KAN | 2410012 | | | | |
| 4 | | | | | | |
| 5 | | | | | | |

**FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY****AACS2204 OBJECT-ORIENTED PROGRAMMING TECHNIQUES**

Diploma in Computer Science (DCS)

Year 1 Semester 3

Tutorial Group 10

PLAGIARISM STATEMENT

Read, complete, and sign this statement to be submitted with the written report.

We confirm that we have read and shall comply with all the terms and conditions of TAR University of Management and Technology's plagiarism policy.

We declare that this assignment is free from all forms of plagiarism and for all intents and purposes is my own properly derived work.

Declaration Statement Acknowledged by

| No | Name | Student ID | Signature | Date |
|----|----------------------|------------|-----------|-------------|
| 1 | CHONG WEN QI | 2410573 | wenqi | 02 May 2025 |
| 2 | WYNНИE WONG KAI XUAN | 2410779 | Wynnie | 02 May 2025 |
| 3 | NG KAH KAN | 2410012 | nkk | 02 May 2025 |
| 4 | | | | |



FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

AACS2204 OBJECT-ORIENTED PROGRAMMING TECHNIQUES

Diploma in Computer Science (DCS)

Year 1 Semester 3

Tutorial Group 10

PLAGIARISM STATEMENT FORM

I, Name Chong Wen Qi Student ID 2410573
Programme Diploma In Computer Science Tutorial Group Group 10
confirm that the submitted work are all my own work and is in my own words.

I Chong Wen Qi acknowledge the use of AI generative technology.

Signatur : **wenqi**
e _____
Date : 02 May 2025



FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

AACS2204 OBJECT-ORIENTED PROGRAMMING TECHNIQUES

Diploma in Computer Science (DCS)

Year 1 Semester 3

Tutorial Group 10

PLAGIARISM STATEMENT FORM

I, Name Ng Kah Kan Student ID 2410012
Programme Diploma In Computer Science Tutorial Group Group 10
confirm that the submitted work are all my own work and is in my own words.

I Ng Kah Kan acknowledge the use of AI generative technology.

Signatur : nkk
e
Date : 02 May 2025



FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

AACS2204 OBJECT-ORIENTED PROGRAMMING TECHNIQUES

Diploma in Computer Science (DCS)

Year 1 Semester 3

Tutorial Group 10

PLAGIARISM STATEMENT FORM

I, Name Wynnie Wong Kai Xuan Student ID 2410779
Programme Diploma In Computer Science Tutorial Group Group 10
confirm that the submitted work are all my own work and is in my own words.

I Wynnie Wong Kai Xuan acknowledge the use of AI generative technology.

Signatur : Wynnie

e

Date : 02 May 2025

Assignment Assessment Rubrics

| No | <u>Student Name</u> | <u>Student ID</u> | <u>Final Marks</u> |
|----|----------------------|-------------------|--------------------|
| 1 | CHONG WEN QI | 2410573 | |
| 2 | WYNНИE WONG KAI XUAN | 2410779 | |
| 3 | NG KAH KAN | 2410012 | |
| 4 | | | |

| | |
|---------------------------|--------------|
| Programme of Study | D (CS) |
| Tutorial Group | Group (10) |

| | |
|--------------|--|
| CLO 2 | Demonstrate an object-oriented program using appropriate programming fundamentals with regards to arrays, methods and exception handling. (P4, PLO3) |
|--------------|--|

Section A: Program Output & Correctness - Individual Marks (25 Marks)

| <u>Task</u> | <u>Weight</u> | <u>Max. Marks</u> | <u>Excellent</u> | <u>Good</u> | <u>Average</u> | <u>Poor</u> | <u>Very Poor</u> | <u>Scores</u> (Point x Weight) | <u>Remarks</u> | | | | | | | | | | |
|-----------------------------|---------------|-------------------|--|---|--|--|---|---|----------------|-------|---|--|---|--|---|--|---|--|--|
| | | | 5 | 4 | 3 | 2 | 1 | | | | | | | | | | | | |
| Completeness of the program | 2.0 | 10 | Completed 100% of the functional requirements. All operations are implemented correctly. | Completed all of the functional requirements but implemented with minor errors. | Completed most of the functional requirements but implemented with errors. | Completed < 50% of the functional requirement and < 50% of the operations implemented correctly. | Did not complete any of the functional requirement at all | <table border="1"> <tr> <th>No</th> <th>Marks</th> </tr> <tr> <td>1</td> <td></td> </tr> <tr> <td>2</td> <td></td> </tr> <tr> <td>3</td> <td></td> </tr> <tr> <td>4</td> <td></td> </tr> </table> | No | Marks | 1 | | 2 | | 3 | | 4 | | |
| No | Marks | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | |
| Exception Handling | 1.0 | 5 | Output is correct with comprehensive | Output is correct with appropriate | Output is correct with little | Output is somewhat correct with little proper | Output is totally wrong with no | <table border="1"> <tr> <th>No</th> <th>Marks</th> </tr> </table> | No | Marks | | | | | | | | | |
| No | Marks | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | |
|--|------------|----------|--|---|--|---|---|----|-------|--|
| | | | exception handling. | exception handling. | exception handling. | exception handling. | proper exception handling. | 1 | | |
| | | | | | | | | 2 | | |
| | | | | | | | | 3 | | |
| | | | | | | | | 4 | | |
| Design of the Output | 1.0 | 5 | Excellent formatted output. Program displays more than expected | Good formatted output and program displays as expected. | Appropriate output design and display as expected. | Poorly designed output. | Minimum to no formatted output with messy display. | No | Marks | |
| | | | | | | | | 1 | | |
| | | | | | | | | 2 | | |
| | | | | | | | | 3 | | |
| | | | | | | | | 4 | | |
| Program Code Quality & Standard | 1.0 | 5 | All identifiers strictly conform to the standard Java naming convention and all are meaningful | Most of the identifiers conform to the standard Java naming convention and are meaningful | Some of the identifiers conform to the standard Java naming convention and some are meaningful | Limited numbers of the identifiers conform to the standard Java naming convention or are barely meaningful. | All of the identifiers conform to the standard Java naming convention or not meaningful | No | Marks | |
| | | | | | | | | 1 | | |
| | | | | | | | | 2 | | |
| | | | | | | | | 3 | | |
| Total Marks (25 marks): | | | | | | | | 1 | | |
| | | | | | | | | 2 | | |
| | | | | | | | | 3 | | |

| | | |
|--|---|--|
| | 4 | |
|--|---|--|

| CLO 3 | Analyse the concepts of encapsulation, inheritance and polymorphism based on programming problems. (C4, PLO2) | | | | | | | | |
|---|---|-------------------|---|---|--|--|--|------------------------------------|----------------|
| <u>Section B: Object-Oriented Concept - Group Marks (45 Marks)</u> | | | | | | | | | |
| Task | Weight | Max. Marks | <u>Excellent</u> | <u>Good</u> | <u>Average</u> | <u>Poor</u> | <u>Very Poor</u> | <u>Scores (Point x Weight)</u> | <u>Remarks</u> |
| | | | 5 | 4 | 3 | 2 | 1 | | |
| Class Implementation | 1.0 | 5m | Identified the proper classes, methods and attributes to solve particular problems. | Identified appropriate classes, methods and attributes to solve particular problems with minor errors. | Able to identify only some classes, methods and attributes to solve particular problems. | Inappropriate classes, methods and attributes are used. | Unable to show understanding on the usage of classes, methods and attributes. | | |
| Object Implementation | 1.0 | 5m | Able to create a structure of objects collaborating among themselves to carry out tasks properly. | Able to create a structure of object collaborating among themselves to carried out task properly with little minor mistakes | Structure of collaborating object created with some mistakes | Structure of collaborating objects created is barely correct. | Wrong structure of object collaboration like one object doing everything itself. | | |
| Abstraction | 1.0 | 5m | Define the class at the proper level of abstraction (abstract classes and methods). | Define the class at the proper level of abstraction with minor mistake | Classes defined at the proper level of abstractions with some mistakes | < 50% of correct abstraction is used | Lack of abstraction (abstract classes and methods). | | |
| Encapsulation | 2.0 | 10m | Completely correct implementation of encapsulation (private modifier, | Correct implementation of encapsulation with only some minor mistakes | Acceptable amount of correct implementation of encapsulation | < 50% of correct implementation for encapsulation applied in program | Completely incorrect implementation of encapsulation (private modifier, | | |

| | | | | | | | | | |
|--------------------------------|------------|------------|--|--|---|---|--|--|--|
| | | | setter and getter methods). | | is applied in program | | setter and getter methods) | | |
| Inheritance | 2.0 | 10m | Completely correct use of inheritance (correct use of extends keyword, super & sub class) | Correct implementation of inheritance with only some minor mistakes | Acceptable amount of correct implementation of inheritance is applied in program | < 50% of correct implementation for inheritance applied in program | Completely incorrect use and implementation of inheritance | | |
| Polymorphism | 2.0 | 10m | Completely correct use and implementation of polymorphism. Methods <code>toString()</code> & <code>equals()</code> correctly overridden. | Correct implementation of polymorphism with only some minor mistakes | Acceptable amount of correct implementation of polymorphism is applied in program | < 50% of correct implementation for polymorphism applied in program | Completely incorrect use and implementation of polymorphism. | | |
| Total Marks (45 marks): | | | | | | | | | |

| CLO 3 | Analyse the concepts of encapsulation, inheritance and polymorphism based on programming problems. (C4, PLO2) | | | | | | | | |
|---|---|-------------------|--|--|--|---|--|------------------------------------|----------------|
| <u>Section C: Object-Oriented Concept - Group Marks (30 Marks)</u> | | | | | | | | | |
| Task | Weight | Max. Marks | <u>Excellent</u> | <u>Good</u> | <u>Average</u> | <u>Poor</u> | <u>Very Poor</u> | <u>Scores (Point x Weight)</u> | <u>Remarks</u> |
| | | | 5 | 4 | 3 | 2 | 1 | | |
| Cohesion | 1.0 | 5m | The code demonstrates exceptional cohesion with all modules, classes, or functions having a clear and single responsibility. | The code demonstrates good cohesion overall where most modules, classes, or functions have a clear and single responsibility | Some modules, classes, or functions have a clear and single responsibility, but others may be slightly ambiguous | Few modules, classes, or functions have a clear and single responsibility. | Modules, classes represent more than one entity and lack of clear responsibility. Poor class cohesion. | | |
| Coupling | 1.0 | 5m | Minimal or no direct dependency between modules, classes, or functions. Coupling is effectively managed, resulting in highly maintainable and modular code | Most modules, classes, or functions have minimal direct dependencies. | Some modules, classes, or functions have moderate direct dependencies. | Many modules, classes, or functions have significant direct dependencies. | Modules, classes, or functions have excessive and tightly coupled dependencies which make module be harder to reuse or test. | | |
| Association, Aggregation and Composition | 2.0 | 10m | Associations between classes are accurately identified and properly established. | Associations between classes are generally identified and properly established. | Some associations between classes may be missing or improperly established. | Several associations between classes are missing or improperly established. | Completely incorrect use and implementation of association, aggregation and | | |

| | | | | | | | | | |
|--------------------------------|------------|------------|---|---|---|---|---|--|--|
| | | | | | | | composition relationship. | | |
| UML Class Diagram | 2.0 | 10m | The UML class diagram demonstrates excellent understanding and implementation. All classes, attributes, and methods are accurately represented with appropriate visibility, data types, and associations. | Most classes, attributes, and methods are accurately represented with appropriate visibility, data types, and associations. | Some classes, attributes, and methods may be missing or inaccurately represented, with visibility, data types, or associations not fully defined. | Many classes, attributes, and methods are missing or inaccurately represented, with visibility, data types, or associations not adequately defined. | Classes, attributes, and methods are missing or inaccurately represented, with visibility, data types, or associations lacking clarity or definition. | | |
| Total Marks (30 marks): | | | | | | | | | |

Table of Contents

| | |
|----------------------------|-----------|
| 1. Introduction | 14 |
| 2. Task Allocation | 15 |
| 3. UML Diagram | 17 |
| 4. System Workflows | 18 |
| 4.1 Main Page | 18 |
| 4.2 Room Management | 19 |
| 4.3 Reservation management | 29 |
| 4.4 Person Management | 36 |
| 5. References | 48 |

1. Introduction

At TT Hotel, operational efficiency and service quality are top priorities. To achieve these goals, we have developed a hotel management system designed specifically for employees. The system is designed to centralize and streamline key hotel operations while ensuring data security and easy access for authorized employees. It provides comprehensive CRUD (create, read, update, delete) capabilities covering three important management areas: personnel management, reservation management, and room management. In terms of personnel management, the system enables employees to efficiently handle employee and customer records. Room attendants play an important role in hotel operations, and they are also fully integrated into the employee management module to ensure smooth coordination between departments.

In addition to personnel management, the system also provides a comprehensive reservation management module. Employees can easily create, update, view, and delete customer reservations, as well as assign rooms and track check-in and check-out activities. This feature is essential for maintaining accurate reservation records and providing guests with a smooth experience. In addition, the room management module also gives employees full control over room details, including room type, availability, status updates (such as available), and prices. Real-time room status tracking will help hotels maximize occupancy rates and optimize overall room maintenance plans, thereby improving daily operational efficiency.

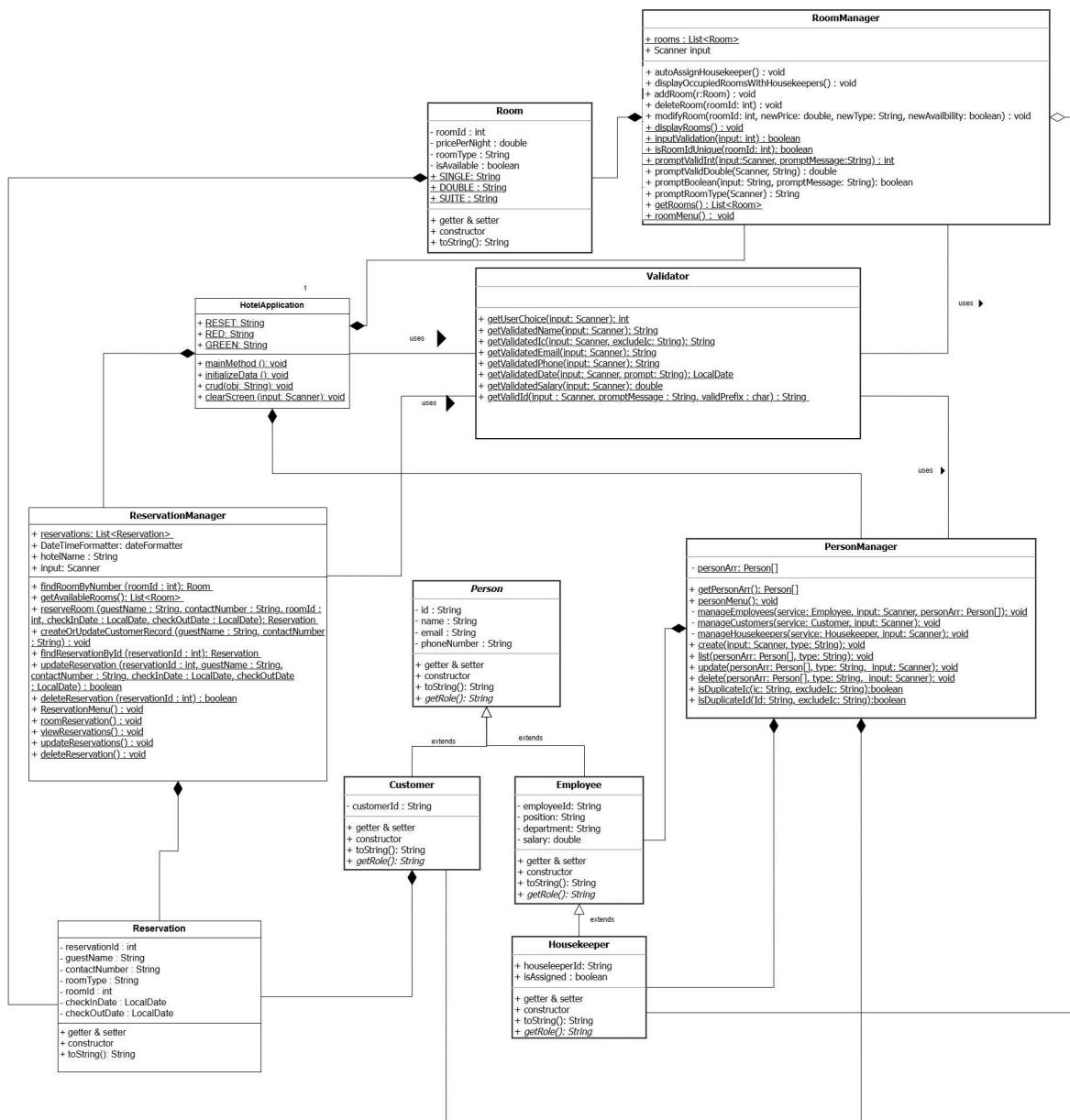
Overall, the hotel management system is designed to improve workflow, increase data accuracy, and protect sensitive information while elevating the service standards of TT Hotels. By adopting this system, we are committed to creating a more organized, efficient, and professional work environment. We believe that this advancement will directly contribute to the continued growth and success of the hotel, providing a better experience not only for our employees, but also for the guests we serve every day.

2. Task Allocation

| Student Name | Modules Handled | Description |
|----------------------|------------------------|---|
| Chong Wen Qi | Room Management | <p>The Room Module enables TT Hotel's staff to efficiently manage all room-related operations. The module provides comprehensive functionality for creating, viewing, modifying, and deleting rooms in the system. Staff can easily track room information including room IDs, prices, types (Single, Double, Suite), and availability status. The system supports features such as housekeeping assignments, status tracking, and validation to prevent errors like duplicate room numbers. The Room Module provides a complete solution for managing hotel accommodations from creation through daily operational needs.</p> |
| Ng Kah Kan | Reservation Management | <p>The Reservation Management module enables TT Hotel's staff to efficiently handle all aspects of guest reservations. The module provides comprehensive functionality for creating, viewing, modifying, and canceling reservations. Staff can easily manage booking dates, room assignments, special requests, and payment information. The system supports features such as checking room availability, processing group bookings, handling extensions or early checkouts, and generating confirmation details. The module ensures accurate tracking of reservations, which helps optimize room allocation, prevent double bookings, and enhance the overall guest experience from pre-arrival through departure.</p> |
| Wynnie Wong Kai Xuan | Person Management | <p>The Person Management module helps TT Hotel's staff efficiently manage all staff and guest records. The module provides comprehensive CRUD (Create, Read, Update, Delete) functionality covering roles such as receptionist, manager, and housekeeping. Staff can easily access and update personal information, positions, schedules, and guest information. The</p> |

| | | |
|--|--|--|
| | | module ensures fast and secure access to important information, which helps centralize data, improve internal coordination, and enhance service quality. |
|--|--|--|

3. UML Diagram



4. System Workflows

4.1 Main Page

```
===== Hotel Management System =====
1. Room Management
2. Reservation Management
3. Person Management
0. Exit
Enter your choice: 
```

The menu option include:

1. Room Management (Option 1) - Allow the staff access to the room management system.
2. Reservation Management (Option 2) - Allow staff to do room reservation for customers.
3. Person Management (Option 3) - Allow staff to handle all user types(employees, customers, housekeepers).
4. Exit (Option 0) - User can end the program.

4.2 Room Management

4.2.1 Option Page

```
===== Hotel Management System =====
1. Room Management
2. Reservation Management
3. Person Management
0. Exit
Enter your choice: 1

Press Enter to continue...
|
```

- User starts the application and selects "Room Management" (option 1) from the main menu

```
=====Room Management =====
1. Add Room
2. List All Room
3. Update Room
4. Delete Room
0. Back to Previous Page
5. View Assigned Housekeeper
Choose an option: |
```

This takes them to the Room Management menu with the following options:

1. Add Room
2. List All Rooms
3. Update Room
4. Delete Room
5. View Assigned Housekeeper
6. Back to Main Menu

4.2.2 Create a new room

```
=====Room Management =====
1. Add Room
2. List All Room
3. Update Room
4. Delete Room
0. Back to Previous Page
5. View Assigned Housekeeper
Choose an option: 1

Adding new Room....
Enter Room ID: -1200
Please enter positive number.
Enter Room ID: dvds2
Not a valid integer.
Enter Room ID:
Not a valid integer.
Enter Room ID: 1001
Room ID 1001 already exists. Please use a unique room ID.
Enter Room ID: 1006+
Not a valid integer.
Enter Room ID: 1006
Enter Price per Night: -123
Please enter positive number.
Value failed validation
Enter Price per Night: as2
Not a valid decimal number.
Enter Price per Night: 123
Enter Room Type (1 SINGLE / 2 DOUBLE / 3 SUITE): 2
Room added successfully.

Press Enter to continue...
```

- User selects option 1 (Add Room)

System prompts for:

- Room ID (must be unique and positive and number)
- Price per night (must be positive and decimal number)
- Room type (SINGLE, DOUBLE, or SUITE)
- New room is created with default availability set to true

- Room is added to the collection and success message is displayed

4.2.3 Listing Room Information

```
=====Room Management =====
```

1. Add Room
2. List All Room
3. Update Room
4. Delete Room
0. Back to Previous Page
5. View Assigned Housekeeper

Choose an option: 2

| Room ID | Price per Night | Type | Availability |
|---------|-----------------|--------|--------------|
| 1001 | 100.00 | SINGLE | Available |
| 1002 | 100.00 | SINGLE | Available |
| 1003 | 100.00 | SINGLE | Available |
| 2001 | 150.00 | DOUBLE | Available |
| 2002 | 150.00 | DOUBLE | Available |
| 3001 | 250.00 | SUITE | Available |
| 1006 | 123.00 | DOUBLE | Available |

```
Press Enter to continue...
```

- User selects option 2 (List All Rooms)

System displays all rooms in a formatted table with columns:

- Room ID
- Price per Night
- Room Type
- Availability status

4.2.4 Updating Room

```
=====Room Management =====
1. Add Room
2. List All Room
3. Update Room
4. Delete Room
0. Back to Previous Page
5. View Assigned Housekeeper
Choose an option: 3

Modifying a Room....
Enter Room ID to modify: 1001
Enter new Price: 256
Enter Room Type (1 SINGLE / 2 DOUBLE / 3 SUITE): 4
  Invalid room type number. Please enter 1, 2, or 3.
Enter Room Type (1 SINGLE / 2 DOUBLE / 3 SUITE): 2
Is the room available? (true/false): fa
  Please enter 'true' or 'false'.
Is the room available? (true/false): false
Room modified successfully.
```

Press Enter to continue...

- User selects option 3 (Update Room)

System prompts for:

- Room ID to modify
- New price per night
- New room type
- New availability status

If room ID exists, the room is updated with new values

Success or failure message is displayed

```
Modifing a Room....  
Enter Room ID to modify: 1111  
Enter new Price: 565  
Enter Room Type (1 SINGLE / 2 DOUBLE / 3 SUITE): 1  
Is the room available? (true/false): f;ase  
Please enter 'true' or 'false'.  
Is the room available? (true/false): false  
Room ID not found.
```

```
Press Enter to continue...
```

If the Room ID did not exist in the data it will not update successfully.

4.2.5 Deleting a Room

```
=====Room Management =====
1. Add Room
2. List All Room
3. Update Room
4. Delete Room
0. Back to Previous Page
5. View Assigned Housekeeper
Choose an option: 3
```

Modifing a Room....

```
=====Room Management =====
1. Add Room
2. List All Room
3. Update Room
4. Delete Room
0. Back to Previous Page
5. View Assigned Housekeeper
Choose an option: 4
```

| Room ID | Price per Night | Type | Availability |
|---------|-----------------|--------|---------------|
| 1001 | 256.00 | DOUBLE | Not Available |
| 1002 | 100.00 | SINGLE | Available |
| 1003 | 100.00 | SINGLE | Available |
| 2001 | 150.00 | DOUBLE | Available |
| 2002 | 150.00 | DOUBLE | Available |
| 3001 | 250.00 | SUITE | Available |
| 1006 | 123.00 | DOUBLE | Available |

Deleting a Room....

Enter Room ID to delete: 1001

Room deleted successfully.

Press Enter to continue...

```
=====Room Management =====
1. Add Room
2. List All Room
3. Update Room
4. Delete Room
0. Back to Previous Page
5. View Assigned Housekeeper
Choose an option: 4
```

| Room ID | Price per Night | Type | Availability |
|---------|-----------------|--------|--------------|
| 1002 | 100.00 | SINGLE | Available |
| 1003 | 100.00 | SINGLE | Available |
| 2001 | 150.00 | DOUBLE | Available |
| 2002 | 150.00 | DOUBLE | Available |
| 3001 | 250.00 | SUITE | Available |
| 1006 | 123.00 | DOUBLE | Available |

Deleting a Room....

Enter Room ID to delete: 1111

Room ID not found.

Press Enter to continue...

- User selects option 4 (Delete Room)
- System displays all rooms
- System prompts for Room ID to let user choose to delete
- If room ID exists, room is removed from collection
- Success or failure message is displayed

4.2.6 Housekeeper Assignment

```
=====Room Management =====
1. Add Room
2. List All Room
3. Update Room
4. Delete Room
0. Back to Previous Page
5. View Assigned Housekeeper
Choose an option: 5

===== Occupied Rooms and Housekeepers =====
No occupied rooms at the moment.

Press Enter to continue...
|
```

```
=====Room Management =====
1. Add Room
2. List All Room
3. Update Room
4. Delete Room
0. Back to Previous Page
5. View Assigned Housekeeper
Choose an option: 5
```

```
Auto assigned housekeeper Ben to Room 1001
```

```
===== Occupied Rooms and Housekeepers =====
Room 1001 | Status: Occupied | Housekeeper: Ben
```

```
Press Enter to continue...
```

```
=====Room Management =====
```

1. Add Room
2. List All Room
3. Update Room
4. Delete Room
0. Back to Previous Page
5. View Assigned Housekeeper

```
Choose an option: 5
```

```
Auto assigned housekeeper Ben to Room 1001
```

```
Auto assigned housekeeper Bryan to Room 1002
```

```
No available housekeeper for Room 1003
```

```
No available housekeeper for Room 2001
```

```
No available housekeeper for Room 2002
```

```
No available housekeeper for Room 3001
```

```
===== Occupied Rooms and Housekeepers =====
```

| | | |
|-----------|------------------|-------------------------------|
| Room 1001 | Status: Occupied | Housekeeper: Ben |
| Room 1002 | Status: Occupied | Housekeeper: Bryan |
| Room 1003 | Status: Occupied | Housekeeper: Not assigned yet |
| Room 2001 | Status: Occupied | Housekeeper: Not assigned yet |
| Room 2002 | Status: Occupied | Housekeeper: Not assigned yet |
| Room 3001 | Status: Occupied | Housekeeper: Not assigned yet |

```
Press Enter to continue...
```



- User selects option 5 (View Assigned Housekeeper)
- System automatically assigns available housekeepers to occupied rooms
- System displays all occupied rooms with their assigned housekeepers
- If there is not available housekeeper but have occupied room it will show not assigned yet

4.3 Reservation management

4.3.1 Room reservation menu

```
=====
|           Hotel Reservation System          |
=====

=====Reservation Management =====
1. Add Reservation
2. List All Reservation
3. Update Reservation
4. Delete Reservation
0. Back to Main Menu
Enter your choice: 
```

The main reservation management menu offers four options:

1. Add Reservation (Option 1)
2. List All Reservation (Option 2)
3. Update Reservation (Option 3)
4. Delete Reservation (Option 4)
5. Back to Main Menu (Option 5)

4.3.2 Add Reservation

When option 1 is selected:

- The system displays available rooms with their types, numbers, and prices
- User enters room number (1001 in the example)
- User enters guest information:
 - Name: NG KAH KAN
 - IC: 061228-14-1395
 - Customer ID: C004
 - Phone Number: 011-3008206
 - Check-in date: 2025-05-02
 - Check-out date: 2025-05-03

```
Enter your choice: 1
=====
|          Room Reservation          |
=====

Available Rooms
-----
Room Type      Room Number      Room Price
-----
SINGLE         1001             100.00
SINGLE         1002             100.00
SINGLE         1003             100.00
DOUBLE          2001            150.00
DOUBLE          2002            150.00
SUITE           3001            250.00
-----
Enter room number: 1001
Enter name: NG KAH KAN
Enter IC (include "-"): 061228-14-1395
Enter Customer ID: C004
Enter Phone Number: 011-3008206
Enter check-in date (yyyy-MM-dd): 2025-05-02
Enter check-out date (yyyy-MM-dd): 2025-05-03
New customer record created.
```

After successful reservation:

- The system confirms the reservation was successful
- Displays reservation details including:
 - Reservation ID: 1001
 - Guest information

- Room information
- Check-in/out dates
- Stay duration (1 day)

```
Reservation successful!
Reservation ID: 1001
Guest Name: NG KAH KAN
Contact Number: 011-3008206
Room Type: SINGLE
Room Number: 1001
Check-in Date: 2025-05-02
Check-out Date: 2025-05-03
Stay Duration: 1 days
```

```
Press Enter to continue...
```

4.3.3 List All Reservation / View Reservation

When option 2 is selected:

- The system displays all current reservations with complete details

```
===== All Reservations =====

Reservation ID: 1001
Guest Name: NG KAH KAN
Contact Number: 011-3008206
Room Type: SINGLE
Room Number: 1001
Check-in Date: 2025-05-02
Check-out Date: 2025-05-03

Press Enter to continue...
█
```

4.3.4 Update Reservation

When option 3 is selected:

- User enters the reservation ID to update (1001)
- System displays current reservation details
- User can update:
 - Guest name (updated to NG KAH KAH)
 - IC (updated to 061228-14-1396)
 - Customer ID (updated to C005)
 - Contact number (updated to 0113008-207)
 - Check-in/out dates
- System confirms successful update and displays updated details

```
Enter your choice: 3

===== Update Reservation =====
Enter reservation ID: 1001

Current Reservation Details:
Reservation ID: 1001
Guest Name: NG KAH KAN
Contact Number: 011-3008206
Room Type: SINGLE
Room Number: 1001
Check-in Date: 2025-05-02
Check-out Date: 2025-05-03
Enter guest name [NG KAH KAN]: NG KAH KAH
Enter IC [061228-14-1395]: 061228-14-1396
Enter Customer ID [C004]: C005
Enter contact number [011-3008206]: 0113008-207
Enter check-in date (yyyy-MM-dd): 2025-05-02
Enter check-out date (yyyy-MM-dd): 2025-05-03

Reservation updated successfully!
Reservation ID: 1001
Guest Name: NG KAH KAH
Contact Number: 0113008-207
Room Type: SINGLE
Room Number: 1001
Check-in Date: 2025-05-02
Check-out Date: 2025-05-03
Stay Duration: 1 days

Press Enter to continue...
```

4.3.5 Delete Reservation

When option 4 is selected:

- User enters the reservation ID to delete (1001)
- System displays reservation details
- User confirms deletion (y)
- System confirms deletion was successful

```
Enter your choice: 4

===== Delete Reservation =====
Enter reservation ID: 1001

Reservation Details:
Reservation ID: 1001
Guest Name: NG KAH KAH
Contact Number: 0113008-207
Room Type: SINGLE
Room Number: 1001
Check-in Date: 2025-05-02
Check-out Date: 2025-05-03

Are you sure you want to delete this reservation? (y/n): y
Reservation deleted successfully!
```

4.3.6 Back to Main Menu

When option 0 is selected:

- System returns to the main menu

```
=====
|           Hotel Reservation System          |
=====

=====Reservation Management =====
1. Add Reservation
2. List All Reservation
3. Update Reservation
4. Delete Reservation
0. Back to Previous Page
Enter your choice: 0
Returning to Main Menu...

Press Enter to continue...
```

4.4 Person Management

4.4.1 Option Page

```
===== Person Management System =====
1. Employee Management
2. Customer Management
3. Housekeeper Management
0. Exit
Choose a management module: █
```

The option include:

1. Employee Management (Option 1) - This option lets you manage regular employees of the hotel.
Note: Regular employees handle general hotel operations but are not specialized for housekeeping tasks.
2. Customer Management (Option 2) - This option lets you manage hotel guests and customers.
3. Housekeeper Management (Option 3) - This option lets you manage housekeeping staff specifically: Housekeepers are a specialized type of employee.
4. Exit (Option 0) - This option returns you to the main menu.

4.4.2 Management Option

```
=====Employee Management =====
```

- 1. Add Employee
- 2. List All Employee
- 3. Update Employee
- 4. Delete Employee
- 0. Back to Main Menu

Choose an option: █

```
=====Customer Management =====
```

- 1. Add Customer
- 2. List All Customer
- 3. Update Customer
- 4. Delete Customer
- 0. Back to Main Menu

Choose an option: █

```
=====Housekeeper Management =====
```

- 1. Add Housekeeper
- 2. List All Housekeeper
- 3. Update Housekeeper
- 4. Delete Housekeeper
- 0. Back to Main Menu

Choose an option: █

Each management option (Employee, Customer, Housekeeper) follows the same CRUD pattern:

- 1. Add (Option 1)
- 2. List All (Option 2)
- 3. Update (Option 3)
- 4. Delete (Option 4)
- 5. Back to Main Menu (Option 0)

4.4.3 Add Person

- System collects basic information (IC, name, email, phone)
- Additional information is collected based on person type:
 - **Customer:** Customer ID

```
=====Customer Management =====
1. Add Customer
2. List All Customer
3. Update Customer
4. Delete Customer
0. Back to Previous Page
Choose an option: 1
Enter IC (include "-"): 123456-12-1234
Enter name: wynnies
Enter Email:
wynnies@gmail.com
Enter Phone Number: 012-1234567
Enter Customer ID:C234
Customer created successfully!
```

- **Employee:** Employee ID, position, department, salary

```
=====Employee Management =====
1. Add Employee
2. List All Employee
3. Update Employee
4. Delete Employee
0. Back to Previous Page
Choose an option: 1
Enter IC (include "-"): 123456-12-1234
Enter name: wynnies
Enter Email:
wynnies@gmail.com
Enter Phone Number: 123-1234567
Enter Employee ID:
E234
Enter Position:
staff
Enter Department:
HR
Enter Salary:
2000
Employee created successfully!
```

- **Housekeeper:** Employee details plus Housekeeper ID

```
=====Housekeeper Management =====
1. Add Housekeeper
2. List All Housekeeper
3. Update Housekeeper
4. Delete Housekeeper
0. Back to Previous Page
Choose an option: 1
Enter IC (include "-"): 123456-12-1234
Enter name: wer
Enter Email:
wer@wer.com
Enter Phone Number: 132-1234567
Enter Employee ID: E567
Enter Position:
staff
Enter Department:
housekeeper
Enter Salary:
2000
Enter Housekeeper ID: H345
Housekeeper created successfully!
```

- System validates all inputs (proper format for IC, email, phone, etc.)

```
=====Employee Management =====
1. Add Employee
2. List All Employee
3. Update Employee
4. Delete Employee
0. Back to Previous Page
Choose an option: 1
Enter IC (include "-"): 123456-12-
Invalid IC! IC must be exactly 14 characters long.
Enter IC (include "-"): 123456-12-1234
Enter name: wynnie
Enter Email:
wynnie
Invalid email format! Please enter again: Enter Email:
wynnie@gmail.com
Enter Phone Number: 123-12
Invalid phone format! Please use format XXX-XXXXXXX. Enter again: Enter Phone Number: 123-1234567
Enter Employee ID:
xxxx
Enter Position:
staff
Enter Department:
hr
Enter Salary:
2000
Employee created successfully!
```

- System checks for duplicate ICs and IDs

```
=====Customer Management =====
1. Add Customer
2. List All Customer
3. Update Customer
4. Delete Customer
0. Back to Previous Page
Choose an option: 1
Enter IC (include "-"): 123456-12-1234
This IC already exists. Please enter a different one.
Enter IC (include "-"): 234567-12-1234
Enter name: SDF
Enter Email:
SDF@SDF.COM
Enter Phone Number: 012-1234567
Enter Customer ID:df
Error: ID must be at least 4 characters long (e.g., C001). Please try again.
Enter Customer ID:C001
Error: Customer ID already exists. Please enter a different one.
Enter Customer ID:C234
Error: Customer ID already exists. Please enter a different one.
Enter Customer ID:C345
Customer created successfully!
```

- Person is added to the system

4.4.4 List Persons

System displays all persons of the selected type with their details

- Employee

```
=====Employee Management =====
1. Add Employee
2. List All Employee
3. Update Employee
4. Delete Employee
0. Back to Previous Page
Choose an option: 2

--- Employee List ---
IC: 001010-10-1011,Name: Ali, Email: ali123@gmail.com, Phone Number: 012-3456789
Employee Id: E001 Position: Manager Department: HR Salary:RM 5000.00

IC: 001111-11-1111,Name: Abu, Email: abu456@gmail.com, Phone Number: 012-4567891
Employee Id: E002 Position: Staff Department: Customer Service Salary:RM 3500.00

IC: 123456-12-1234,Name: wynnies, Email: wynnies@gmail.com, Phone Number: 123-1234567
Employee Id: E234 Position: staff Department: HR Salary:RM 2000.00
```

- Customer

```
=====Customer Management =====
1. Add Customer
2. List All Customer
3. Update Customer
4. Delete Customer
0. Back to Previous Page
Choose an option: 2

--- Customer List ---
IC: 901111-10-3456,Name: Alice, Email: alice666@gmail.com, Phone Number: 018-9876543
Customer Id: C001

IC: 911222-10-2468,Name: Ammy, Email: ammy789@gmail.com, Phone Number: 018-5678901
Customer Id: C002

IC: 123456-12-1234,Name: wynnies, Email: wynnies@gmail.com, Phone Number: 012-1234567
Customer Id: C234

IC: 234567-12-1234,Name: SDF, Email: SDF@SDF.COM, Phone Number: 012-1234567
Customer Id: C345
```

- Housekeeper

```
=====Housekeeper Management =====
```

- 1. Add Housekeeper
- 2. List All Housekeeper
- 3. Update Housekeeper
- 4. Delete Housekeeper
- 0. Back to Previous Page

```
Choose an option: 2
```

```
--- Housekeeper List ---
```

```
IC: 920123-10-3579,Name: Ben, Email: ben@gmail.com, Phone Number: 012-5678912  
Employee Id: E003 Position: Housekeeper Department: Housekeeping Salary:RM 2500.00  
Housekeeper Id: H001
```

```
IC: 920112-10-1001,Name: Bryan, Email: bryan@gmail.com, Phone Number: 012-6789123  
Employee Id: E004 Position: Housekeeper Department: Housekeeping Salary:RM 2500.00  
Housekeeper Id: H002
```

```
IC: 123456-12-1234,Name: wer, Email: wer@wer.com, Phone Number: 132-1234567  
Employee Id: E567 Position: staff Department: housekeeper Salary:RM 2000.00  
Housekeeper Id: H345
```

4.4.5 Update Person

- User enters IC to identify the person
- User can update any field (name, email, phone, plus role-specific fields)
- System validates all inputs

```
=====Customer Management =====
1. Add Customer
2. List All Customer
3. Update Customer
4. Delete Customer
0. Back to Previous Page
Choose an option: 3
Enter IC to Update (include "-"): 234567-12-1234
Enter name: 345
Invalid name! Only alphabets and spaces are allowed.
Enter name: sdf
Enter Email:
234
Invalid email format! Please enter again: Enter Email:
wer@wer.com
Enter Phone Number: 012-3456789
Enter new Customer ID: C567
Customer updated successfully!
```

- Confirmation is displayed

- Employee

```
=====Employee Management =====
1. Add Employee
2. List All Employee
3. Update Employee
4. Delete Employee
0. Back to Previous Page
Choose an option: 3
Enter IC to Update (include "-"): 123456-12-1234
Enter name: wynnie
Enter Email:
wynnie@gmail.com
Enter Phone Number: 123-1234567
Enter new position: manager
Enter new department: HR
Enter new salary: 5000
Employee updated successfully!
```

- Customer

```
=====Customer Management =====
1. Add Customer
2. List All Customer
3. Update Customer
4. Delete Customer
0. Back to Previous Page
Choose an option: 3
Enter IC to Update (include "-"): 234567-12-1234
Enter name: sdf
Enter Email:
sdf@sdf.com
Enter Phone Number: 123-1234567
Enter new Customer ID: C567
Customer updated successfully!
```

- Housekeeper

```
=====Housekeeper Management =====
```

1. Add Housekeeper
2. List All Housekeeper
3. Update Housekeeper
4. Delete Housekeeper

0. Back to Previous Page

Choose an option: 3

Enter IC to Update (include "-"): 123456-12-1234

Enter name: wer

Enter Email:

wer@gmail.com

Enter Phone Number: 123-1234567

Enter new position: manager

Enter new department: housekeeper

Enter new salary: 4000

Housekeeper updated successfully!

4.4.6 Delete Person

- User enters IC of person to delete
- System removes the person and shifts the array to fill the gap
- Error message will be displayed if the IC not in the array(Example: Customer)

```
=====Customer Management =====
1. Add Customer
2. List All Customer
3. Update Customer
4. Delete Customer
0. Back to Previous Page
Choose an option: 4
Enter Customer IC to delete: 234567-12-1234
Customer with ID 234567-12-1234 not found.
```

- Confirmation is displayed
- Employee

```
=====Employee Management =====
1. Add Employee
2. List All Employee
3. Update Employee
4. Delete Employee
0. Back to Previous Page
Choose an option: 4
Enter Employee IC to delete: 123456-12-1234
Employee with IC 123456-12-1234 deleted.
```

- Customer

```
=====Customer Management =====
1. Add Customer
2. List All Customer
3. Update Customer
4. Delete Customer
0. Back to Previous Page
Choose an option: 4
Enter Customer IC to delete: 234567-12-1234
Customer with IC 234567-12-1234 deleted.
```

- Housekeeper

```
=====Housekeeper Management =====
```

1. Add Housekeeper
2. List All Housekeeper
3. Update Housekeeper
4. Delete Housekeeper

0. Back to Previous Page

Choose an option: 4

Enter Housekeeper IC to delete: 123456-12-1234

Housekeeper with IC 123456-12-1234 deleted.

5. References

1. *Polymorphism in Java - javatpoint.* (n.d.). [www.javatpoint.com.
https://www.javatpoint.com/runtime-polymorphism-in-java](http://www.javatpoint.com/runtime-polymorphism-in-java)
2. *Inheritance (The JavaTM Tutorials > Learning the Java Language > Interfaces and Inheritance).* (n.d.).
[https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html](http://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html)
3. *UML Class Diagram Tutorial.* (n.d.).
[https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/](http://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/)