IS2103 – Enterprise Systems Server-side Design and Development

AY 2024/25 Semester 1

Pair Project

Opening Narrative

Figure 1 – Artist's impression of the new Merlion Hotel external facade and its Premier Room.

Merlion Hotel is a new hotel located in the central business district of Singapore that is currently under construction. It is scheduled to commence business operation in the first quarter of 2025. As part of its preparation to commence business operations, Merlion Hotel has engaged **Kent Ridge Technology (KRT)**, a global technology consulting and solutions development company headquartered in Singapore, to develop a new **Hotel Reservation System (HoRS)** to support its core business processes. More specifically, the new HoRS will be used by Merlion Hotel to manage its room inventory, price rates, room reservations and guests.

In addition to its own internal use, HoRS will also be exposed to external partners such as price comparison websites, online travel websites and travel agencies. This will allow Merlion Hotel to sell its room inventory to a wider audience. At this juncture, Merlion Hotel has already inked an agreement with **Holiday.com**. This would allow users of Holiday.com's **Holiday Reservation System** to search for and reserve rooms in Merlion Hotel as part of their holiday packages.

You and your partner have recently been recruited by KRT to design and develop this exciting enterprise-scale software system. You are all roaring to get started!

Business Domain and System Domain Introduction

The Merlion Hotel has more than 500 rooms in five different configurations – Deluxe Room, Premier Room, Family Room, Junior Suite and Grand Suite. From time to time, the hotel may change its room configurations to cater to changing needs of guests. The hotel has various amenities to cater to the needs of guests, but these would be managed with a different information system.

The new system mainly manages the various room types as well as the room inventory. Each room is numbered using a combination of a two-digit floor number and a two-digit sequence number for that floor, e.g., Room 2015 would refer to the fifteenth room on floor twenty.

A room is available for allocation to a particular guest reservation if the room status is available and has not been allocated to another guest reservation for the entire duration of the current reservation. This business rule is important as the guests cannot be asked to change room in the middle of their stay. Thus, the system needs to ensure that there is sufficient room inventory to fulfil a new reservation before accepting it. In the event of overbooking for a particular room type, the guests can be upgraded to the next higher available room type at no additional cost.

In other words, when a new reservation request is received, the system does not need to allocate a hotel room before confirming the reservation. The system only needs to ensure that the hotel has sufficient room inventory to fulfil the new reservation. Allocation of hotel rooms to guest reservations is done as a daily batch job at 2 am in the morning. If the system encounters any errors during the room allocation process, an exception report would be generated. The duty manager will need to check the exception report in the morning to resolve the errors.

The standard check-out time is 12 noon on the day of departure and the standard check-in time is 2 pm on the day of arrival. The housekeeping department will typically perform cleaning between 12 noon to 2 pm. Early check-in is possible only if the room that is allocated to the guests is ready before 2 pm. Late check-out is possible only if the room has not been allocated to guests on the day of departure.

Merlion Hotel adopts a four-tier pricing strategy for its room rates:

- Published Rate – This is the base rate per night for a room type that is offered to all walk-in reservations regardless of the check-in date.

- Normal Rate – This is the normal rate per night for a room type that is offered to online reservation via HoRS, including external partners. This is the default rate per night for any room type offered to all online reservations (own and partner).

- Peak Rate – This is the peak rate per night for a room type that is offered to all online reservations (own and partner) via HoRS, including external partners, during predefined peak seasons.

- Promotion Rate – This is the discounted rate per night for a room type that is offered to all online reservations (own and partner) via HoRS, including to external partners, during predefined promotional periods.

The total reservation fee payable by the guest for a reservation is calculated by summing the prevailing rate per night of each night of stay for the entire duration of stay. For example, if a guest books a Deluxe Room for 3 nights, the reservation fee will be the sum of first day’s rate per night, second day’s rate per night and third day’s rate per night. If either peak rate or promotion rate is defined for a particular room type on a particular date, it will take precedence over the normal rate. If both peak rate and promotion rate are defined for a particular room type on a particular date, the promotion rate will take precedence.

High-level System Architecture

All software elements constituting the HoRS are to be developed in Java using Jakarta Enterprise Edition (Jakarta EE). In particular, Jakarta Enterprise Beans (EJB) and Jakarta Persistence API (JPA) technologies are to be used in conjunction with a suitable Relational Database Management System (RDBMS) such as MySQL. Only Command-line Interface (CLI) client applications are required.

The high-level architecture of HoRS is depicted in a block diagrams as shown in Figure 2 below. HoRS Management Client is an application used by Merlion Hotel's employees and HoRS Reservation Client is used by guests. External partners will typically utilise their own client applications to connect to HoRS's web services.

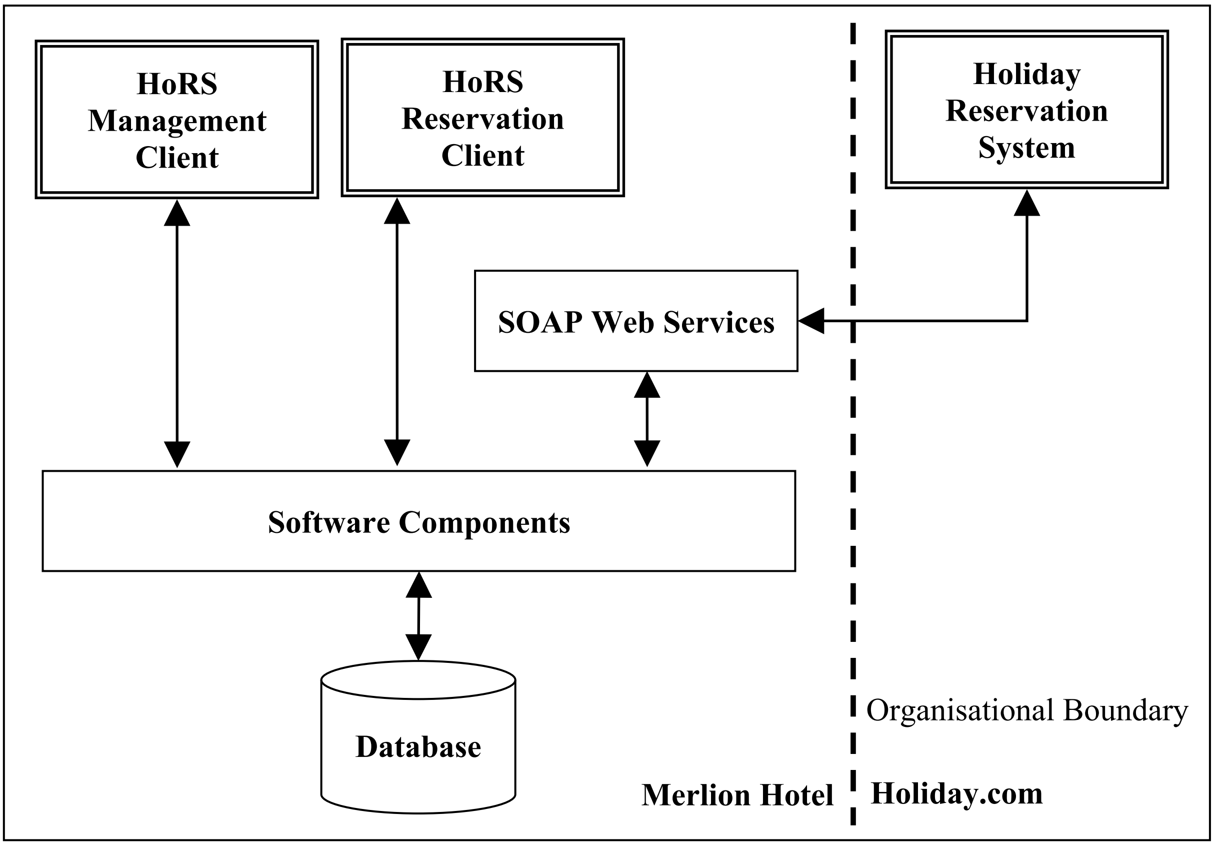

Figure 2 – High-level architecture of the Hotel Reservation System (HoRS).

Project Tasks – Architecture and Design

Design a suitable **logical data model** consisting of a set of entity classes, their attributes, and the relationships among the entity classes to support the entire HoRS software system. You are **NOT** required to draw a UML class diagram. You are only required to create the entity classes using JPA and then apply the **forward engineering** technique to generate the physical data model, i.e., the underlying relational database tables.

Java annotations from JPA must be used appropriately to decorate the entity classes to enforce the strictest possible integrity constraints on the logical data model. More explicitly, you are to ensure the correctness and integrity of the data that are eventually stored in the underlying relational database according to some well-defined business rules or assumptions. For examples:

```
@Column(length = 32, nullable = false)
private String firstName;
@Column(length = 9, nullable = false, unique = true)
private String identificationNumber;
```

are preferred over:

```
private String firstName;
private String identificationNumber;
```

And:

```
@ManyToOne(optional = false)
@JoinColumn(nullable = false)
private Customer customer;
```

are preferred over:

```
@ManyToOne
private Customer customer;
```

Using Jakarta EE, design a suitable **physical architecture** for the HoRS software system taking reference from the high-level architecture shown in Figure 2. The Holiday Reservation System should be created as a Java Platform, Standard Edition (Java SE) client application that utilises the HoRS's web services.

Document the business rules, rationales and assumption for your logical data model in a Microsoft Word document. You are **NOT** required to draw any formal software engineering diagram such as UML diagram. If you wish to provide any illustrations, you may use any informal block diagram notation.

Project Tasks – Business Use Cases for HoRS Management Client

The UML use case diagram for the HoRS Management Client is shown in Figure 3 and 4. A brief description of each business use case together with the associated business rules are given in Table 1.

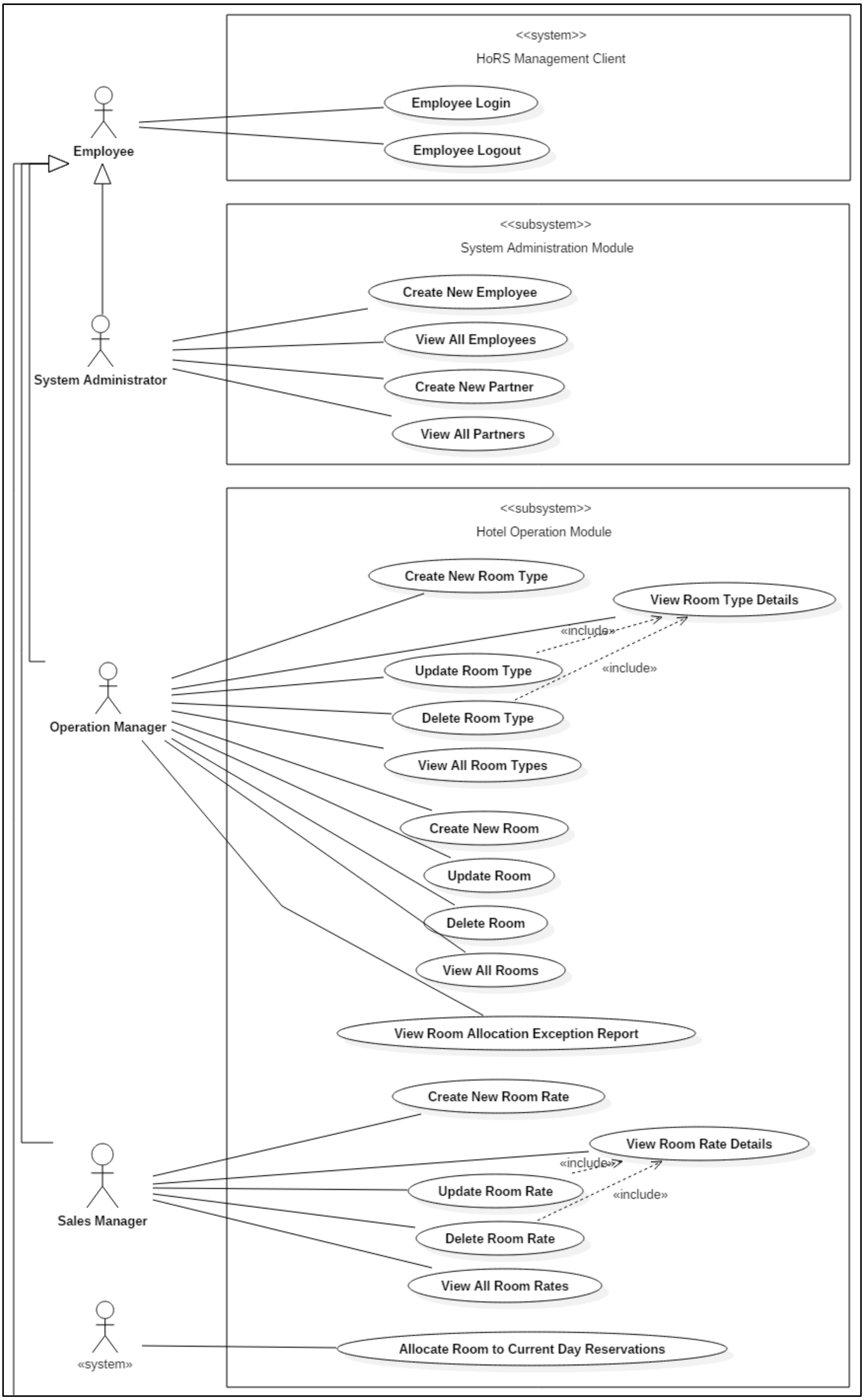

Figure 3 – UML use case diagram for the HoRS Management Client (Part 1 of 2).

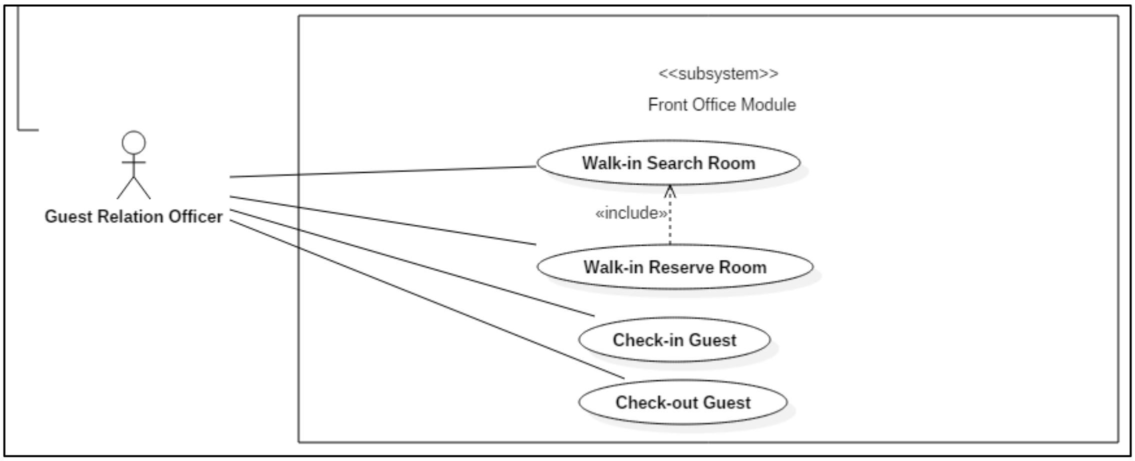

Figure 4 – UML use case diagram for the HoRS Management Client (Part 2 of 2).

S/N	Use Case	Description/Business Rules
1	Employee Login	• Allows an employee to login to the system and assume the preconfigured user role. • May only be performed if employee is not currently login to the system. • Employee must be currently login to the system to perform all other use cases. • A default system administrator account should be created as part of data initialisation.
2	Employee Logout	• Logout the employee. • May only be performed if employee is currently login to the system.
3	Create New Employee	• Creates a new employee record with the required credentials (username and password) and user role (corresponds to use case actor). • It is not necessary to support update and delete of employee records.
4	View All Employees	• Display a list of all employee records in the system.
5	Create New Partner	• Creates a new partner record with the required credentials. • All employees of the same partner organisation are assumed to login to HoRS remotely using a single account. • It is not necessary to support update and delete of partner records.
6	View All Partners	• Display a list of all partner records in the system.
7	Create New Room Type	• Creates a new room type record. • Basic attributes should include name, description, size, bed, capacity and amenities.
8	View Room Type Details	• View the details of a particular room type record.
9	Update Room Type	• Update the details of a particular room type record.
10	Delete Room Type	• Delete a particular room type record. • A room type record can only be deleted if it is not used. • Otherwise, it should be marked as disabled, and no new room should be created for disabled room type.
11	View All Room Types	• Display a list of all room type records in the system.
12	Create New Room	• Creates a new room record. • Basic attributes should include room type, room number and room status (available and not available).

S/N	Use Case	Description/Business Rules
13	Update Room	• Update the details of a particular room record. • Room status is also updated via this use case.
14	Delete Room	• Delete a particular room record. • A room record can only be deleted if it is not used. • Otherwise, it should be marked as disabled, excluded from the hotel room inventory for that room type and should not be allocated to a new reservation.
15	View All Rooms	• Display a list of all room records in the system.
16	View Room Allocation Exception Report	• Generate a report showing two types of room allocation exception. • First type is no available room for reserved room type but upgrade to next higher room type is available – A room is automatically allocated by the system. • Second type is no available room for reserved room type and no upgrade to next higher room type is available – No room is automatically allocated by the system.
17	Create New Room Rate	• Creates a new room rate record. • Basic attributes should include name, room type, rate type (published, normal, peak and promotion), rate per night, validity period (start/end dates for peak and promotion rate types only).
18	View Room Rate Details	• View the details of a particular room rate record.
19	Update Room Rate	• Update the details of a particular room rate record.
20	Delete Room Rate	• Delete a particular room rate record. • A room rate record can only be deleted if it is not used. • Otherwise, it should be marked as disabled and new reservation should not be made with the disabled room rate.
21	View All Room Rates	• Display a list of all room rate records in the system.
22	Allocate Room to Current Day Reservations	• Retrieve a list of all reservations for check-in on the current date and allocate the required room(s) for the reserved room type. • If the required room(s) for the reserved room type is not available, raise an exception in the exception report (see use case 16). • This use case should normally be triggered by a server-side timer daily at 2 am. But for development and evaluation purposes, the system should also allow the use case to be triggered manually for an arbitrary date to be input by the user.
23	Walk-in Search Room	• Search an available room across all room types offered by the hotel according to the check-in date and check-out date. • The reservation amount should be calculated based on the prevailing published rate of that room type. • The system needs to ensure that the hotel has sufficient room inventory to fulfil the new reservation taking into consideration other previously accepted reservations.

S/N	Use Case	Description/Business Rules
24	Walk-in Reserve Room	• Reserve a room offered in the search results (see use case 23). • It is possible for a walk-in guest to reserve more than one room. • A walk-in guest does not need to be a registered guest of the hotel. • For same day check-in, allocate the required room(s) immediately if reservation is made after 2 am.
25	Check-in Guest	• Check-in a guest by informing him/her of the allocated room(s). • Room allocation exception needs to be handled manually.
26	Check-out Guest	• Check-out a guest to indicate the end of his/her visit to the hotel.

Table 1 – Use case descriptions and business rules for the HoRS Management Client.

Project Tasks – Business Use Cases for HoRS Reservation Client

The UML use case diagram for the HoRS Reservation Client is shown in Figure 5. A brief description of selected business use cases together with the associated business rules are given in Table 2.

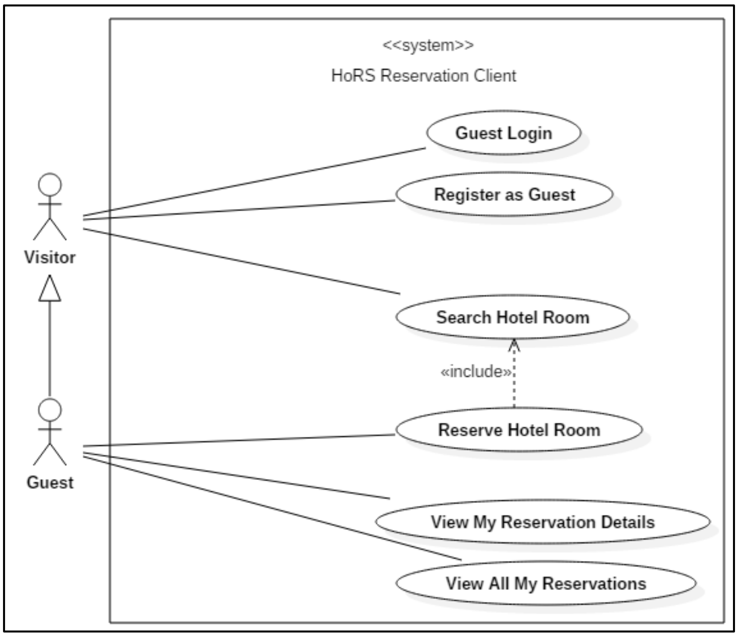

Figure 5 – UML use case diagram for the HoRS Reservation Client.

S/N	Use Case	Description/Business Rules
1	Guest Login	• Allows a guest to login to the system. • May only be performed if guest is not currently login to the system. • Guest must be currently login to the system to perform reservation-related use cases.
2	Register as Guest	• Allows a visitor to register as a guest of the hotel. • Each customer must be uniquely identifiable, e.g., email, mobile phone number or passport number.

3	Search Hotel Room	• Search an available room across all room types offered by the hotel according to the check-in date and check-out date. • The reservation amount should be calculated based on the available prevailing rate of that room type. • The system needs to ensure that the hotel has sufficient room inventory to fulfil the new reservation taking into consideration other previously accepted reservations.
4	Reserve Hotel Room	• Reserve a room offered in the search results (see use case 3). • It is possible for a guest to reserve more than one room. • For same day check-in, allocate the required room(s) immediately if reservation is made after 2 am.
5	View My Reservation Details	• Display the details of a particular guest reservation.
6	View All My Reservations	• Display a list of reservation records for the guest.

Table 2 – Use case descriptions and business rules for the HoRS Reservation Client.

Project Tasks – Business Use Cases for Holiday Reservation System

The UML use case diagram for the Holiday Reservation System is shown in Figure 6. A brief description of selected business use cases together with the associated business rules are given in Table 3.

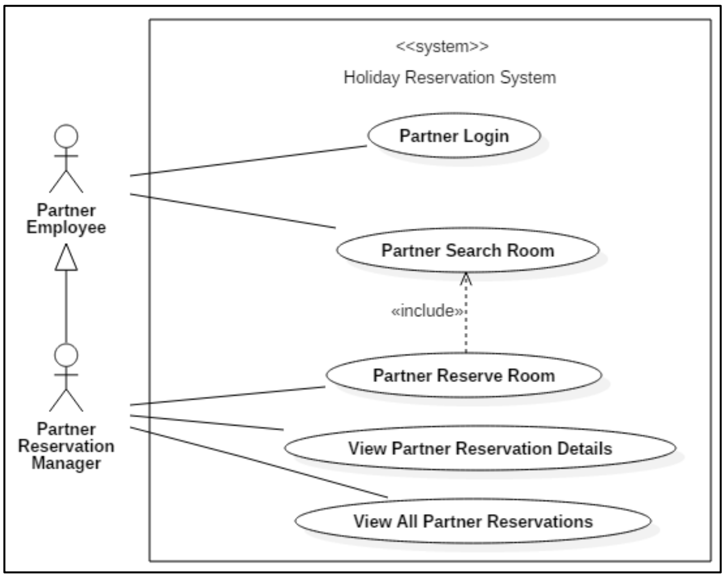
<<system>>
Holiday Reservation System
Partner Login
Partner Employee
Partner Search Room
«include»
Partner Reserve Room
Partner Reservation Manager
View Partner Reservation Details
View All Partner Reservations

Figure 6 – UML use case diagram for the Holiday Reservation System.

S/N	Use Case	Description/Business Rules
1	Partner Login	• Allows an employee of a partner organisation to login to the system. • May only be performed if partner employee is not currently login to the system. • Partner employee must be currently login to the system to perform reservation-related use cases.

2	Partner Search Room	• Search an available room across all room types offered by the hotel according to the check-in date and check-out date. • The reservation amount should be calculated based on the available prevailing rate of that room type. • The system needs to ensure that the hotel has sufficient room inventory to fulfil the new reservation taking into consideration other previously accepted reservations.
3	Partner Reserve Room	• Reserve a room offered in the search results (see use case 2). • It is possible to reserve more than one room. • The reservation is done by a partner employee on behalf of a customer of the partner organisation. • Such a reservation would not be visible in the HoRS Reservation Client when the partner customer login with his/her HoRS guest account. • For same day check-in, allocate the required room(s) immediately if reservation is made after 2 am.
4	View Partner Reservation Details	• Display the details of a particular partner reservation.
5	View All Partner Reservations	• Display a list of reservation records for the partner.

Table 3 – Use case descriptions and business rules for the Holiday Reservation System.

Project Tasks – User Interface

For each client application, implement a suitable CLI interface consisting of menus, prompts, cues and feedback messages (both informative and error messages). Users should be able to interact with the CLI interface with relatively ease.

Project Tasks – System Quality

All client applications should demonstrate some degree of input data validation, and all server-side business processing should demonstrate some degree of input data validation and business rules validation. In general, logic-tier components should not trust the presentation-tier clients in a distributed software system.

Error handling should be done properly using Java checked exceptions. In addition, error handling for each use case should reflect alternative and/or exceptional course(s) of action that deviate from the basic course of action as appropriate.

The quality of the coding will be assessed via criteria such as appropriate and correct usage of the JPA EntityManager API methods, EJB component types, number of remote session bean business method invocation, etc.

Assessment Criteria

This pair project is worth 40 marks out of the overall assessment for the whole course. In general, both members in the group will get the same score if there is no negative peer review.

The assessment criteria are listed in Table 4.

Architecture and Design Appropriateness and correctness of logical data model, physical data model and physical system architecture. Also includes the implementation of the entity classes and the forward engineering of the underlying relational database table.	10 marks
System Functionalities – HoRS Management Client Implementation of use cases with respect to the logical data model as well as complexity of the use cases that are implemented as measured by the fulfillment of the business rules.	15 marks
System Functionalities – HoRS Reservation Client Same as HoRS Management Client	6 marks
System Functionalities – Holiday Reservation System Same as HoRS Management Client Also include web services exposed on the server-side of HoRS.	4 marks
System Quality Two sub-criteria will be evaluated: 1. The quality of the user interface (Command Line Interface) including cues and input data validation. 2. The quality of the coding including general aspects such as proper exception handling and Jakarta EE specific aspects such as correct use of annotations and the entity manager.	5 marks
Total	40 marks

Table 4 – Assessment criteria.

General Project Schedule

The general project schedule is shown in Table 5. Late submission will not be graded.

S/N	Week	Date	Activity	Remark
1	7	3 Oct	Release of pair project specification.	
2	12	8 Nov	Release of test data.	
3	13	15 Nov	Release of test cases. Evaluation of Project System	
4	13	17 Nov	Submission of Project Deliverables Submission of Project Peer Review	Upload to Canvas by 11:59 pm

Table 5 – General project schedule.

General Evaluation Process

The evaluation of your project system would be done using a set of test cases that cover the greater majority of the use cases. The test cases assume that your project system is preloaded with a set of test data that would be released in Week 12. Failure to preload this set of test data might result in your project system failing one or more test cases.

For use cases that are not covered by the test cases, they would be evaluated after the submission of the project system.

Deliverables Submission Instructions

Place all the deliverables into a **single zip archive file** with the following folder structure:

- **doc folder**:
  - A Microsoft Word document containing the write-up of the business rules, rationales and assumption of your logical data model (**optional**).

- **src folder**:
  - Place all your NetBeans project folder(s) inside here.
  - You should "clean" the projects beforehand, the file size will be much smaller.
  - Test data, if any, such as the default system administrator account and all backend data initialisation should be loaded using a singleton session bean decorated with the `@Startup` annotation. Do not use a SQL script to load the test data.
  - Ensure that your main enterprise application's EJB module contains a glassfish-resources.xml file and a persistence unit persistence.xml.

- **readme.txt**:
  - A text file containing the personal particulars of all group members, including full name, matriculation number and email.

Name the zip archive file with your project group name in the format "**PPXX.zip**" where "XX" is your group number. Please double check your Canvas Project for the correct group number. Upload the zip archive file to the Canvas Assignment: Assignments > Pair Project. The submission deadline is **Sunday, 17 November, 11:59 PM**.

-- End of Pair Project Specification --