IS2103 – Enterprise Systems Server-side Design and Development

AY 2019/20 Semester 1

Pair Project

Opening Narrative

Figure 1 – Promotional artwork of Merion Car Rental.

Merlion Car Rental (MCR) is a new car rental company with planned outlets strategically located at different parts of Singapore. The company will commence business operation in January 2020 and is currently preparing for its launch. As part of the preparation, MCR has engaged **Kent Ridge Technology (KRT)**, a global technology consulting and solution development company that is headquartered in Singapore, to develop a new **Car Rental Management System (CaRMS)** to support its core business processes. More specifically, the new CaRMS will be used by MCR to manage its car inventory, price packages, rentals, employees and customers. The customers of MCR will also be using the system to rent cars.

In addition to its own internal use, the CaRMS will also be exposed to external partners such as price comparison websites, online travel websites and travel agencies. This will allow MCR to sell its car inventory to a wider audience. At this juncture, MCR has already inked an agreement with **Holiday.com**. This would allow users of Holiday.com's **Holiday Reservation System** to search for and rent cars from MCR 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

MCR will have more than 50 cars available for rental at the time of launch and this will eventually increase to 200 cars. Each car is associated with a particular make and model, e.g., Toyota Corolla Altis, and categorised into one of four categories – Luxury Sedan, Family Sedan, Standard Sedan and SUV/Minivan. From time to time, the company may change its fleet of cars and product offerings. The new CaRMS is required to manage all enterprise data such as car-related records, employee-related records, customer-related records and more importantly rental records.

↳ store in DB

A car is available for rental during a particular period if it is not undergoing servicing or repair work, or not in transit from one outlet to another. Rental rates are typically quoted on a daily basis for a contiguous rental period of 24 hours. That is, if the rental duration is 24 hours or any part thereof, the customer will be charged for one day of rental. It is possible for multiple rental rates to be applied to a single rental. It is also possible for a particular date to have multiple rental rates in effect. In such cases, the lowest rental rate will apply.

For example, if a customer rents a car from Thursday to Saturday, an off-peak rate might apply for Thursday and a higher peak rate might apply for Friday and Saturday. But if there is a promotion on Saturday, the lower promotion rate will apply for Saturday instead of the peak rate. The total rental fee payable by a customer for a car rental is thus calculated by summing the applicable daily rate for each 24 hours period, or any part thereof, based on the pickup time.

When renting a car, customers can specify various options such as car type, and pickup and return locations. For the former option, customers can choose to rent a car of a specific make and model, or any car from a particular category. For the latter option, customers can specify the return location to be the same as the pickup location or a different one, subject to the opening hours of that outlet.

The rental fee can be paid upfront during the online reservation, or at the time of pickup at the outlet. For the latter option, reservation will need to be guaranteed by a credit card. Cancellation that is made at least 14 days before the pickup time will not incur any penalty. Otherwise, the following penalty rates will apply:

- Less than 14 days but at least 7 days before pickup – 20% penalty.
- Less than 7 days but at least 3 days before pickup – 50% penalty.
- Less than 3 days before pickup – 70% penalty.

For a reservation in which payment is to be made during pickup, the penalty amount will be deducted from the credit card that is used for securing that reservation.

In order to maximise business opportunities, MCR allows a customer to reserve a car that is last returned to a different outlet as long as the car can be moved to the required pickup outlet in time. To facilitate a timely pickup, a minimum transit time of 2 hours is enforced. MCR will assign an employee to drive the car from the last returned outlet to the new pickup outlet. For example, it should be possible for a customer to rent a car for pickup at outlet A on 3 October 10 am that is last returned to outlet B on 3 October 8 am. MCR will assign an employee to drive the car from outlet B to outlet A at an appropriate time taking into consideration the travel duration and pickup time.

The CaRMS is also responsible for handling the actual allocation of cars to fulfil the rental reservations at each outlet on a daily basis. When performing car allocation, it is important to take into consideration i) whether a rental reservation requires a car of a specific make and model, or just any car from a particular category; and ii) allocation of cars that are not currently in the particular outlet physically. As part of the car allocation process, the CaRMS also needs to generate the required transit driver dispatch records so that MCR can assign its employees to move cars from one outlet to another.

Finally, the processing of car rental reservations by external partners generally follows a similar process with the exception of having to capture details of the partners' customers separately since they are not directly registered with MCR.

High-level System Architecture

All software elements constituting the CaRMS are to be developed in Java using the Java Platform, Enterprise Edition (Java EE). In particular, Enterprise JavaBeans (EJB) and Java 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 the CaRMS is depicted in a block diagrams as shown in Figure 2 below. CaRMS Management Client is an application used by MCR's employees and CaRMS Reservation Client is used by customers. External partners will typically utilise their own client applications to connect to CaRMS's web services.

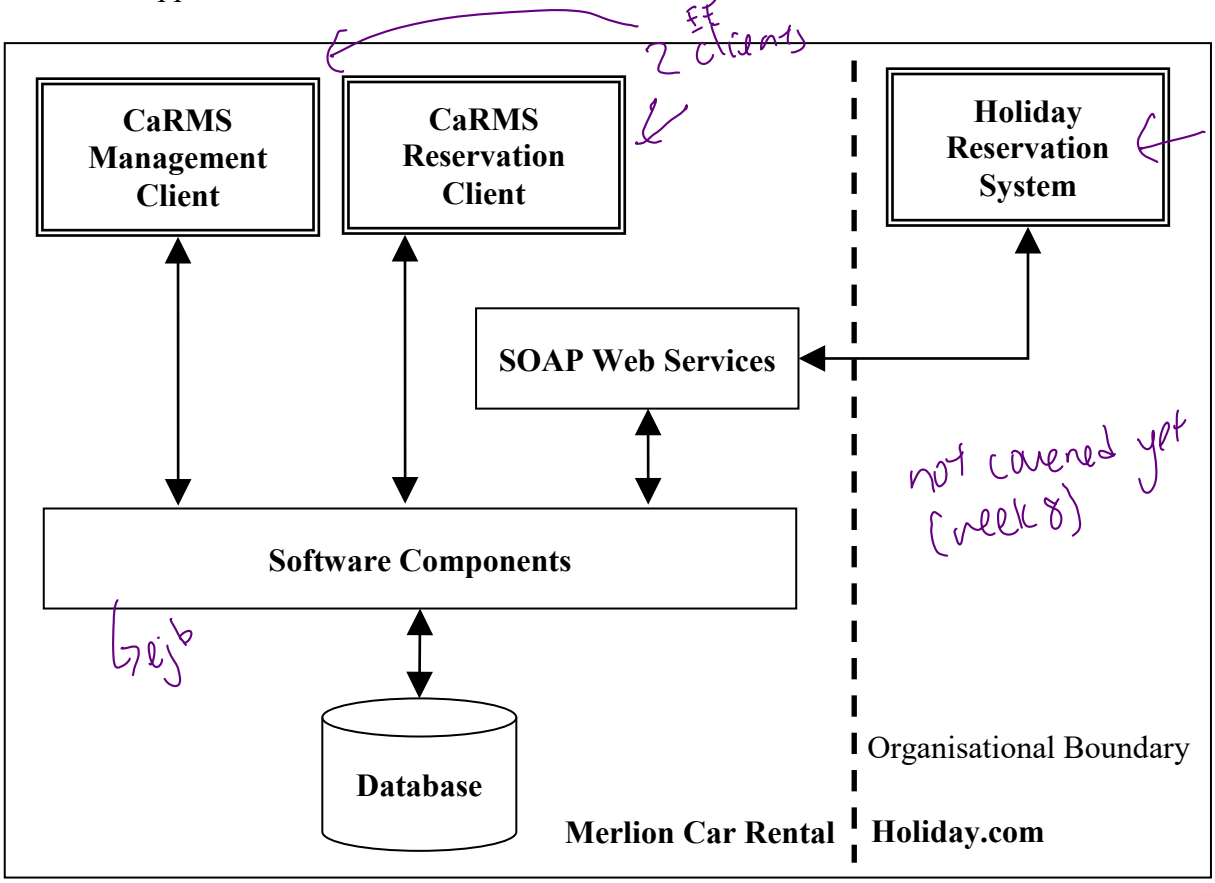


Figure 2 – High-level architecture of the Car Rental Management System (CaRMS).

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 CaRMS 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 in order 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 (assuming that the customer relationship attribute is mandatory):

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

Using Java EE, design a suitable **physical architecture** for the CaRMS software system taking reference from the high-level architecture shown in Figure 2. Your architecture should state clearly the various Java EE software elements that you are developing and the rationale.

Document the business rules, rationales and assumption for your logical data model as well as a brief description of the physical architecture 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 CaRMS Management Client

The UML use case diagram for the CaRMS 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, 2 and 3.

S/N	Use Case	Description/Business Rules
1	Create New Outlet	• Backend data initialisation only. • Create a new outlet record. • Basic attributes should include address and opening hours.
2	Create New Employee	• Backend data initialisation only. • Create a new employee record with the required credentials, user role (corresponds to use case actor) and outlet.
3	Create New Partner	• Backend data initialisation only. • Create a new partner record.
4	Create New Category	• Backend data initialisation only. • Creates a new (car) category record.
5	Allocate Cars to Current Day Reservations	• Retrieve a list of all car rental reservations for pickup on the current date and allocate an available car for the reserved car (make and) model or category. • When allocating cars, priority should be accorded to cars that are already in the pickup outlet, or will be returned to the pickup outlet in time. • Cars that are at a different outlet from the pickup outlet should be allocated only when necessary.
6	Generate Transit Driver Dispatch Records for Current Day Reservations	• Retrieve a list of car allocations for pickup on the current date that require movement from another different outlet. • Generate a transit driver dispatch record for each car. • Each outlet should only manage dispatch records for cars that are to be moved to itself.
7	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.
8	Employee Logout	• Logout the employee. • May only be performed if employee is currently login to the system.
9	Create Rental Rate	• Create a new car rental rate record for a particular car category. • Basic attributes should include name, car category, rate per day (i.e., 24 hours period), validity period (if applicable). • No rental reservation can be made if a rental rate is not available for a particular category for a particular day.
10	View All Rental Rates	• Display a list of all car rental rate records in the system. • Records should be sorted in ascending order by car category and validity period.

Table 1 – Use case descriptions and business rules for the CaRMS Management Client (Part 1 of 3).

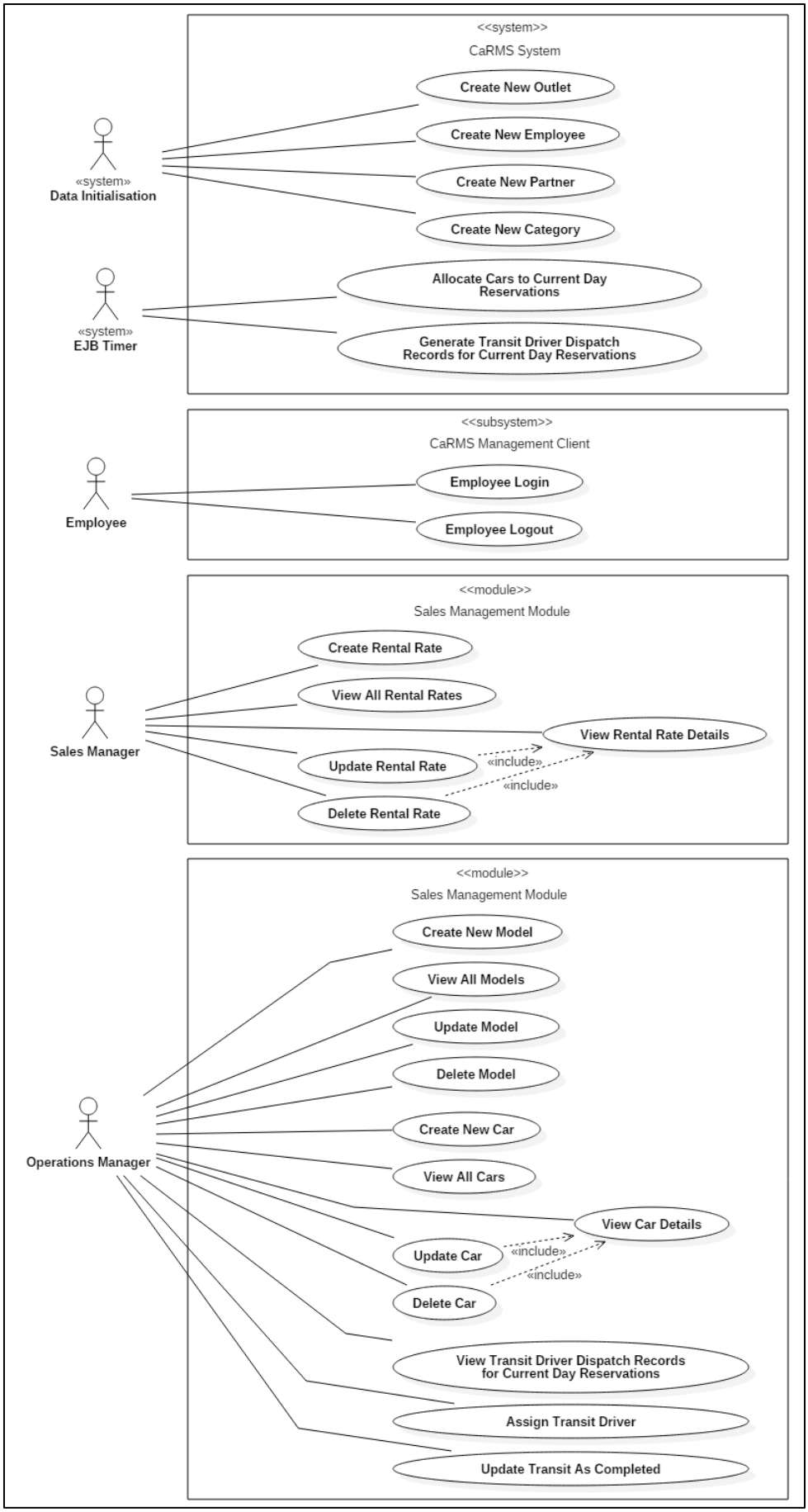


Figure 3 – UML use case diagram for the CaRMS Management Client (Part 1 of 2).

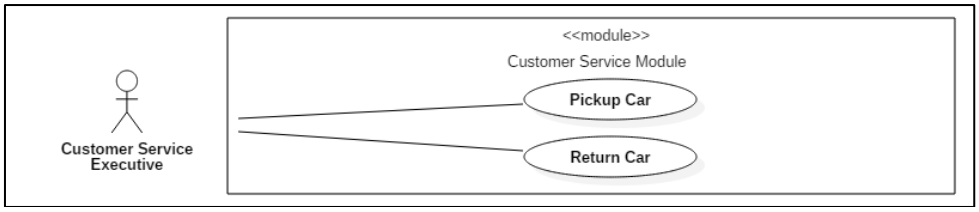


Figure 4 – UML use case diagram for the CaRMS Management Client (Part 2 of 2).

S/N	Use Case	Description/Business Rules
11	View Rental Rate Details	• View the details of a particular car rental rate record.
12	Update Rental Rate	• Update the details of a particular car rental rate record.
13	Delete Rental Rate	• Delete a particular car rental rate record. • A rental 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 rental rate.
14	Create New Model	• Create a new (make and) model record for a particular car category. • Basic attributes should include make and model.
15	View All Models	• Display a list of all (make and) model records in the system. • Records should be sorted in ascending order by car category, make and model.
16	Update Model	• Update the details of a particular (make and) model record.
17	Delete Model	• Delete a particular (make and) model record. • A (make and) model record can only be deleted if it is not used. • Otherwise, it should be marked as disabled and new car record should not be created with the disabled (make and) model.
18	Create New Car	• Create a new car record for a particular (make and) model. • Basic attribute should include license plate number, colour, status (in outlet or on rental) and location (specific customer or outlet).
19	View All Cars	• Display a list of all car records in the system. • Records should be sorted in ascending order by car category, make, model and license plate number.
20	View Car Details	• View the details of a particular car record.
21	Update Car	• Update the details of a particular car record.
22	Delete Car	• Delete a particular car record. • A car record can only be deleted if it is not used. • Otherwise, it should be marked as disabled and cannot be rented out.
23	View Transit Driver Dispatch Records for Current Day Reservations	• Retrieve a list of all transit driver dispatch records for the current day for the current outlet.
24	Assign Transit Driver	• Assign a driver to a particular transit driver dispatch record for the current day for the current outlet. • The driver may be any employee that is working in the current outlet.
25	Update Transit As Completed	• Update a particular transit driver dispatch record for the current day for the current outlet as completed after the transit driver has returned to the outlet with the car.

Table 2 – Use case descriptions and business rules for the CaRMS Management Client (Part 2 of 3).

S/N	Use Case	Description/Business Rules
26	Pickup Car	• Record a customer picking up a car. • If rental fee payment is deferred during online reservation, it must be paid before the car can be collected. • Status and location of the car must be updated.
27	Return Car	• Record a customer returning a car. • Status and location of the car must be updated.

Table 3 – Use case descriptions and business rules for the CaRMS Management Client (Part 3 of 3).

Project Tasks – Business Use Cases for CaRMS Reservation Client

The UML use case diagram for the CaRMS Reservation Client is shown in Figure 5. A brief description of each business use case together with the associated business rules are given in Table 4 and 5.

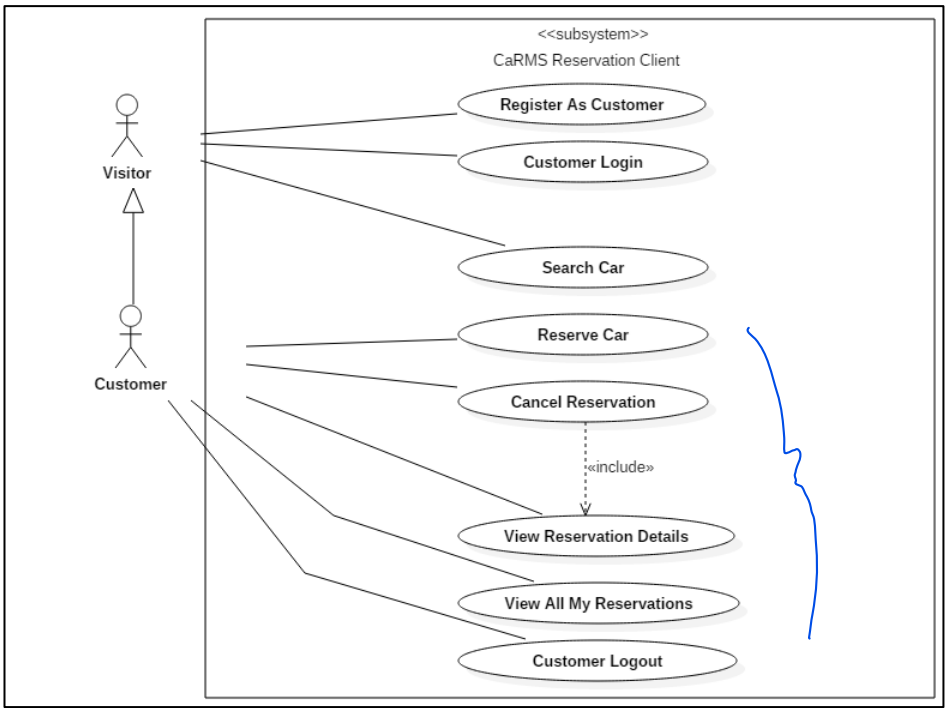


Figure 5 – UML use case diagram for the CaRMS Reservation Client.

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

Table 4 – Use case descriptions and business rules for the CaRMS Reservation Client (Part 1 of 2).

S/N	Use Case	Description/Business Rules
3	Search Car	• Search an available car across all category and (make and) model offered by MCR according to the pickup date/time, pickup outlet, return date/time and return outlet. • The rental fee amount should be calculated based on the available prevailing rental rate of that particular category. • The system needs to ensure that MCR has sufficient car inventory to fulfil the new reservation, including transiting cars between outlets in order to prevent overselling while maximising revenue.
4	Reserve Car	• Reserve a car offered in the search results (see use case 3). • You may assume that a customer can only reserve one car per transaction. • Record the credit card details of the customer for handling immediate or deferred rental fee payment.
5	Cancel Reservation	• Cancel a particular car rental reservation. • If the rental fee has already been paid, refund the balance after deducting the cancellation penalty amount. • If the rental fee has not been paid, charge the customer's credit card for the penalty amount.
6	View Reservation Details	• Display the details of a particular car rental reservation.
7	View All My Reservations	• Display a list of car rental reservation records for the customer.
8	Customer Logout	• Logout the customer. • May only be performed if customer is currently login to the system.

Table 5 – Use case descriptions and business rules for the CaRMS Reservation Client (Part 2 of 2).

Project Tasks – Business Use Cases for Holiday Reservation System

Java SE

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

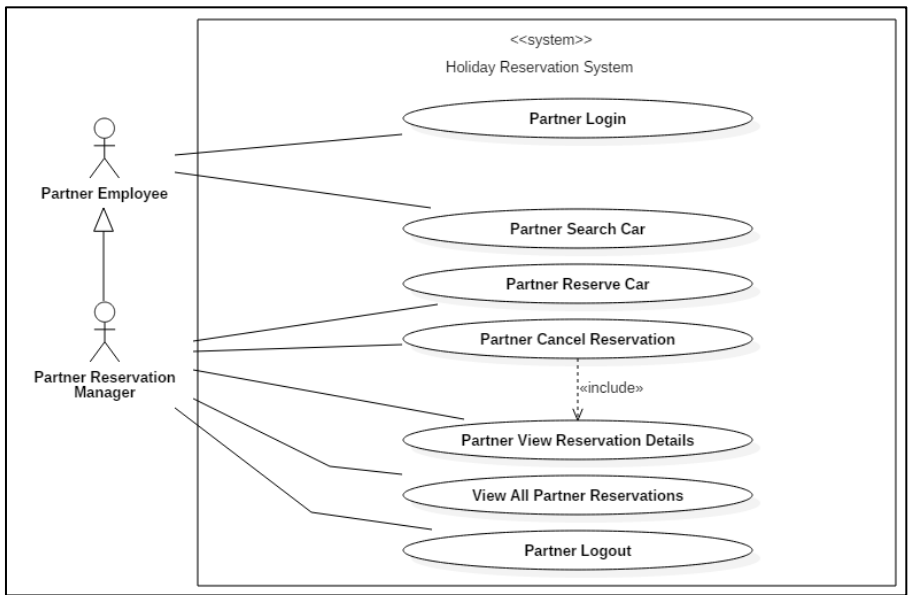


Figure 6 – UML use case diagram for the Holiday Reservation System.

S/N	Use Case	Description/Business Rules
1	Partner Login	• Allows a partner to login to the system. • May only be performed if partner is not currently login to the system. • Partner must be currently login to the system to perform reservation-related use cases. • Each partner only has one login account that is to be shared by all of the partner's employees.
2	Partner Search Car	• Search an available car across all category and (make and) model offered by MCR according to the pickup date/time, pickup outlet, return date/time and return outlet. • The rental fee amount should be calculated based on the available prevailing rental rate of that particular category. • The system needs to ensure that MCR has sufficient car inventory to fulfil the new reservation, including transiting cars between outlets in order to prevent overselling while maximising revenue.
3	Partner Reserve Car	• Reserve a car offered in the search results (see use case 2). • You may assume that a partner can only reserve one car per transaction. • Record the credit card details of the partner's customer for handling immediate or deferred rental fee payment. • Need to record additional details of the partner's customer.
4	Partner Cancel Reservation	• Cancel a particular car rental reservation. • If the rental fee has already been paid, refund the balance after deducting the cancellation penalty amount. • If the rental fee has not been paid, charge the partner's customer's credit card for the penalty amount.
5	Partner View Reservation Details	• Display the details of a particular car rental reservation.
6	View All Partner Reservations	• Display a list of car rental reservation records for the partner.
7	Partner Logout	• Logout the partner. • May only be performed if partner is currently login to the system.

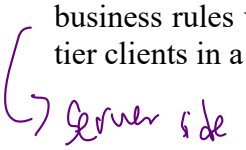
Table 6 – Use case descriptions and business rules for the Holiday Reservation System.

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 invocations, etc.

Assessment Criteria

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

The assessment criteria are listed in Table 7 below:

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 – CaRMS 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 – CaRMS Reservation Client Same as CaRMS Management Client	6 marks
System Functionalities – Holiday Reservation System Same as CaRMS Management Client Also include web services exposed on the server-side of CaRMS.	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 Java EE specific aspects such as correct use of annotations and the entity manager.	5 marks
Total	40 marks

Table 7 – Assessment criteria.

General Assignment Schedule

The general assignment schedule is shown in Table 8. Late submission will not be graded.

S/N	Week	Date	Activity	Remark
1	8	8 Oct	Release of pair project specification.	
2	13	TBC	Evaluation of Project System	
3	13	17 Nov	Submission of Project Deliverables Submission of Project Peer Review	Upload to LumiNUS by 11:59 pm

Table 8 – General assignment schedule.

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 your solution's architecture and design.
- **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 particular 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 IVLE Project for the correct group number. Upload the zip archive file to the LumiNUS folder: Deliverables Submission > Pair Project. The submission deadline is **Sunday, 17 November 2019, 11:59 PM**.

-- End of Pair Project Specification --