**IS2103 Enterprise Systems Server-side Design and Development**

**PAIR PROJECT**

****

**Pair Group: PP01**

**Merlion Car Rental (MCR) Management System**

Tan Jia Le Damien (A0183560E)

Wong Jin Fu Shaun (A0189408R)

**Table of Contents**

[**1.** **Introduction** 3](#_Toc24756760)

[1.1. Components in System 3](#_Toc24756761)

[1.2. Benefits of System 3](#_Toc24756762)

[**2.** **Project Scope** 4](#_Toc24756763)

[2.1. Subsystem/Modules in CaRMS 4](#_Toc24756764)

[2.1.1. CaRMS Management System 4](#_Toc24756765)

[2.1.1.1. CaRMS Management Client 5](#_Toc24756766)

[2.1.1.2. Sales Management Module 5](#_Toc24756767)

[2.1.1.3. Customer Service Module 5](#_Toc24756768)

[2.1.2. CaRMS Reservation System 6](#_Toc24756769)

[2.1.2.1. CaRMS Reservation Client 6](#_Toc24756770)

[2.1.3. Holiday Reservation System 7](#_Toc24756771)

[**3.** **High-level Systems Architecture** 8](#_Toc24756772)

[3.1. Logical Data Model (Class Diagram) 9](#_Toc24756773)

[3.2. Physical Data Model 11](#_Toc24756774)

[**4.** **Business Requirements Analysis** 12](#_Toc24756775)

[4.1. CaRMS Management System 12](#_Toc24756776)

[4.1.1. Business Rules, Assumptions & Rationales 12](#_Toc24756777)

[4.2. CaRMS Reservation System 16](#_Toc24756778)

[4.2.1. Business Rules, Assumptions & Rationales 16](#_Toc24756779)

[4.3. Holiday Reservation System 18](#_Toc24756780)

[4.3.1. Business Rules, Assumptions & Rationales 18](#_Toc24756781)

# **Introduction**

**Merlion Car Rental (MCR)** is a new car rental company that will begin business operations in January 2020. In preparation for its launch, MCR has engaged Kent Ridge Technology (KRT) to develop a new Car Rental Management System (CaRMS).

My partner and I have documented, analysed and reviewed the various core business processes that CaRMS will need to support for MCR, including inventory management, rental reservations, account management for customers and employees as well as a web service link with Holiday.com.

In this write-up, we list all the business rules, rationales and assumptions made in the system that we have coded. We also explain about the high-level systems architecture and the logical data model used to structure our coding process.

## Components in System

CaRMS comprises three main systems—*CaRMS Management System*, *CaRMS Reservation System* and *Holiday Reservation System*.

The *CaRMS Management System* handles the bulk of the core internal business processes required by MCR. This ranges from employee login / logout, CRUD of cars and rental rates and assigning of transit drivers. The *CaRMS Reservation System* handles all rental reservations made by customers of MCR. The core business processes include customer login / logout, searching and reserving of a car as well as reviewing past reservation details. The *Holiday Reservation System* handles all rental reservations made through Holiday.com’s website. It allows MCR to reach a larger audience and achieve higher volume of sales.

## Benefits of System

Automation of procurement, payment, and inventory will establish clear and accessible digital records that can streamline future procurement processes and accurately track inventory levels. Some key benefits include:

* Eliminating the need for frequent manual inventory-taking and updating of its car fleet
* Facilitating sales audits through automated rental reservation records

CaRMSalso optimizes customer and employee management and facilitates internal audits. Some key benefits include:

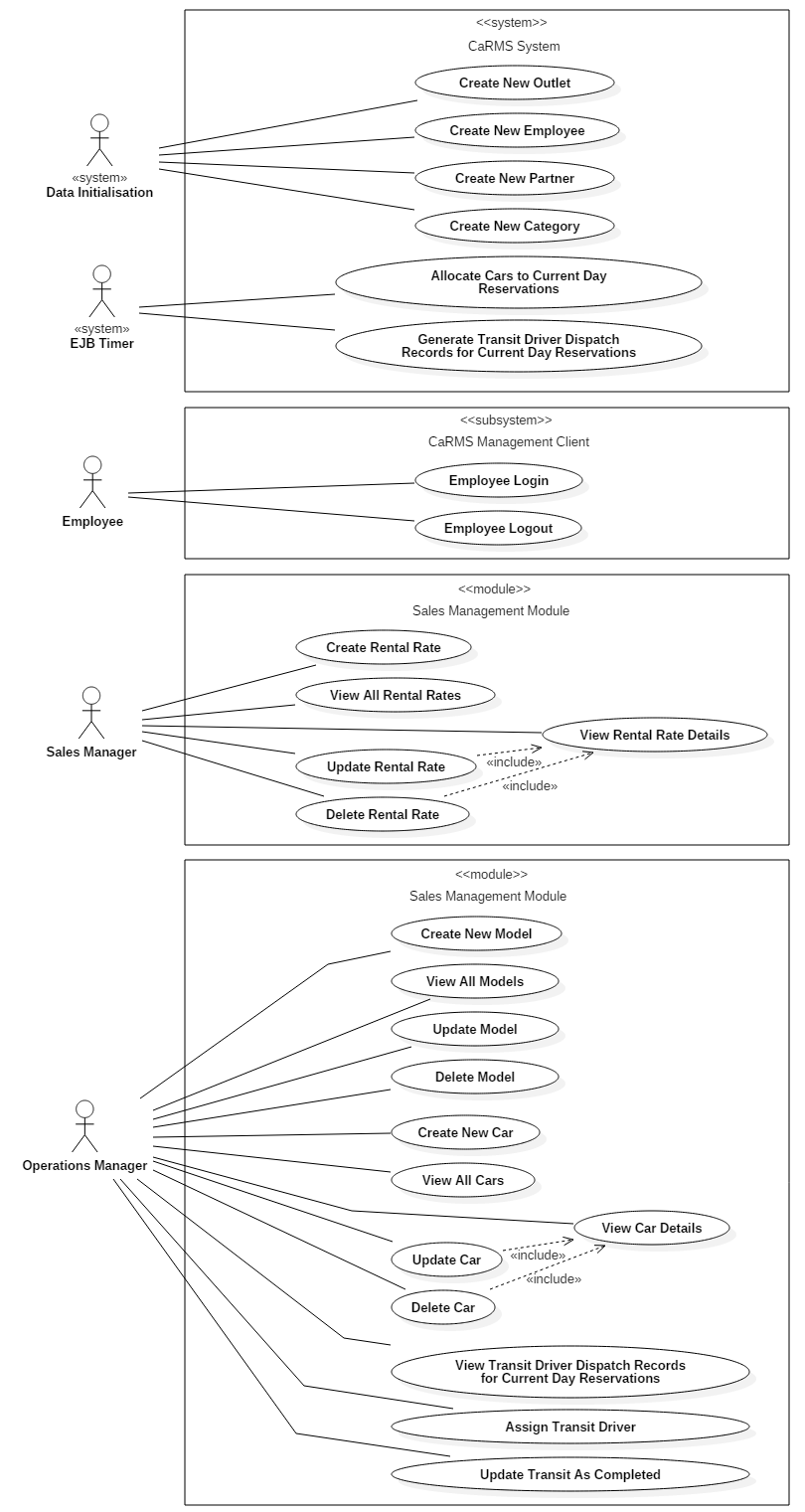
* Allowing MCR to sell to both customers through its system as well as online through partner websites like Holiday.com
* Automating the process of verifying and checking rental reservations and whether any car or model is available
* Allow CRUD of all enterprise data such as car-related records and employee-related records
* Generation of various reports, created from a combination of data collected by all subsystems

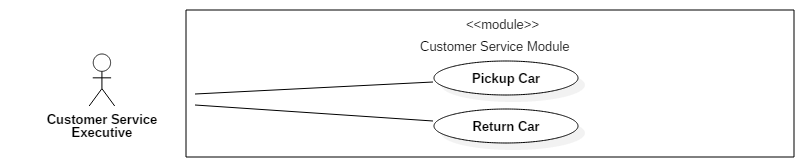
# **Project Scope**

Analysis of MCR’s needs have led to the creation of three systems to handle their requirements. They are the *CaRMS Management System*, *CaRMS Reservation System* and *Holiday Reservation System*.

## Subsystem/Modules in CaRMS

### CaRMS Management System





The CaRMS System itself handles data initialisation which allows the system to load the necessary data the first time it is deployed and run. This is handled through the singleton session bean and covers the creation of new outlets, employees, partners and car categories.

The system also consists of the EJB timer which is responsible for allocating the cars to according to each given day’s rental reservations stored in the system database, and then generating the transit driver dispatch records if necessary for that day.

#### CaRMS Management Client

The CaRMS Management Client allows employees to login and logout of the system in order to perform their business-related tasks.

#### Sales Management Module

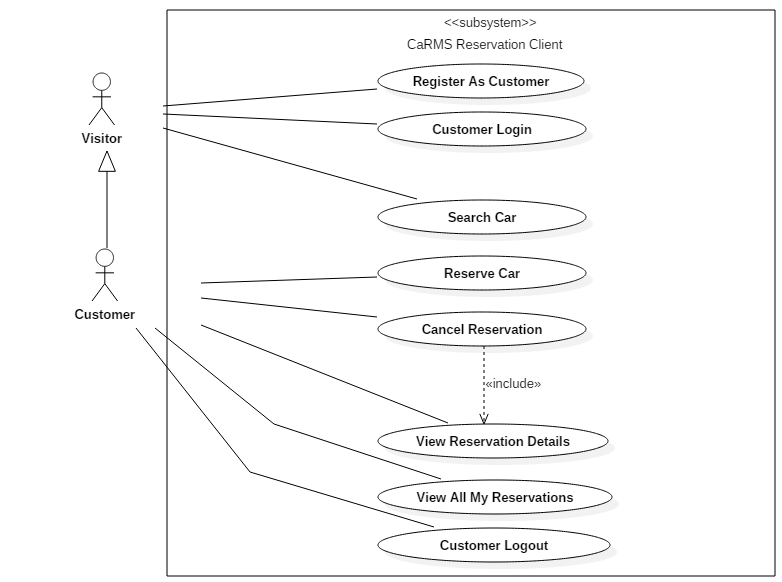
The Sales Management Module allows the outlet’s sales manager to perform CRUD functions of car rental rates. By logging into the system, the sales manager is able to create new rental rates, update existing rental rates, delete any existing rental rates and view a specific rental rate’s details.

This module also allows the operations manager to perform CRUD functions car models and cars. The operations manager is also able to view transit driver dispatch records for any given day’s reservations in order to assign a transit driver or update a record as completed according to the business of that day.

#### Customer Service Module

The Customer Service Module handles the front-of-desk business process required by MCR. Through this module, MCR’s customer service executives are able to process customers’ pickup and return car requests directly at any physical outlet.

### CaRMS Reservation System



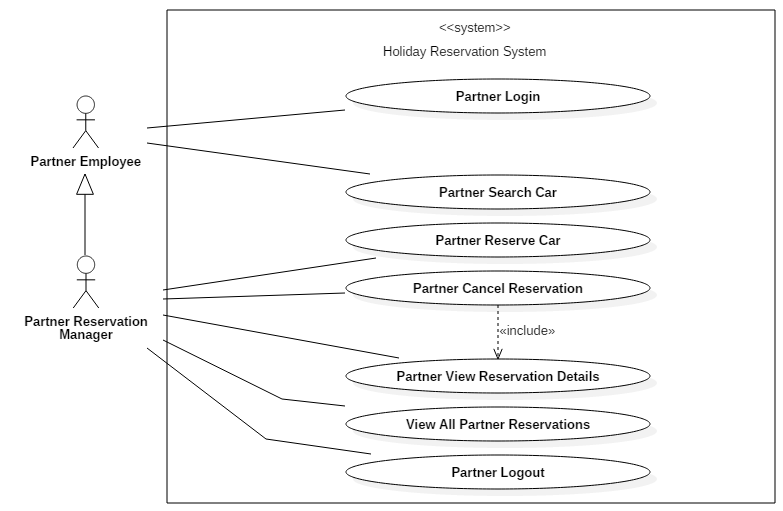
#### CaRMS Reservation Client

The CaRMS Reservation Client is responsible for handling all customer or visitor requests to reserve or search for a car. Any visitor can register as a customer and will be able to search for a car to see if it is available for rental at a given date.

However, in order to make a rental reservation, the customer must be logged in and he will then be able to reserve a car that he has searched for. He is also able to cancel any past reservations made, and to view his past reservations and reservation details.

### Holiday Reservation System

The Holiday Reservation Client is responsible for handling all reservations made through Holiday.com’s website. The system is managed by Holiday.com’s reservation manager, who is an employee of Holiday.com and is able to login, search for cars, make a reservation, cancel a reservation, view any reservation record details and to view all past partner reservations.



Any customer that wishes to rent a car as part of Holiday.com’s holiday packages may do so through this system.

# **High-level Systems Architecture**

**Software Components**

**SOAP Web Services**

**CaRMS**

**Management**

**Client**

**CaRMS**

**Reservation**

**Client**

**Holiday**

**Reservation**

**System**

**Database**

Organisational Boundary

**Merlion Car Rental**

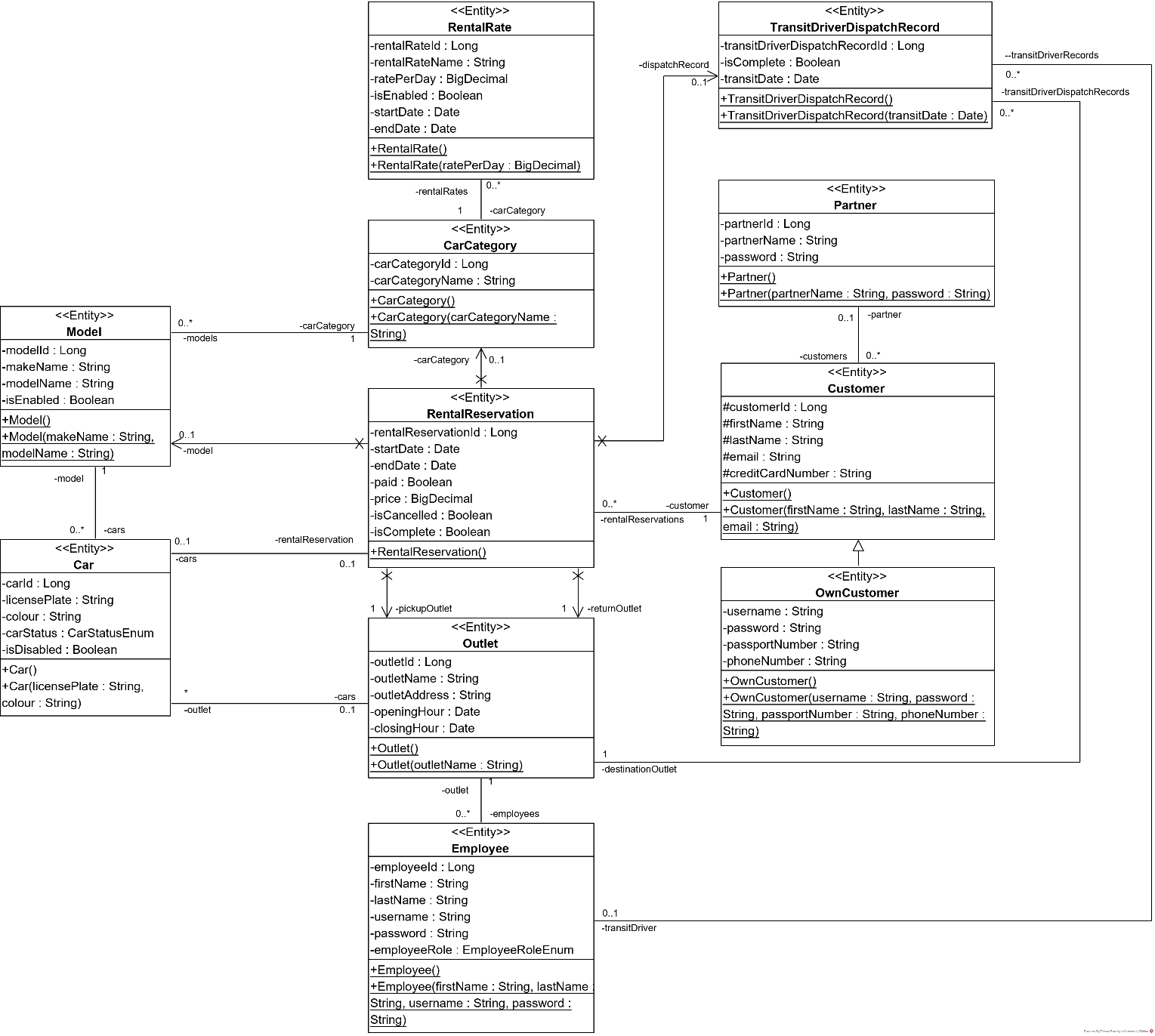
**Holiday.com**

The CaRMS Management Client and CaRMS Reservation Client are Java EE clients while the Holiday Reservation System is a Java SE application that will interact with the overall system through SOAP Web Services.

The software components of the system comprise the Enterprise JavaBeans (EJB) and Java Persistence API (JPA) technologies that was used in conjunction with the platform-independent object-oriented query language—Java Persistence Query Language (JPQL)—as a Relational Database Management System (RDBMS) to make queries to our database of entities.

All client applications are Command-line Interface (CLI) only.

# Logical Data Model (Class Diagram)



**Rationale of General Structure**

We designed the entire system centred around the *RentalReservation* entity as it is the entity class with the greatest number of relationships with all the other classes. The nature of these relationships are indicated by the multiplicity value as stated in the above class diagram.

The *OwnCustomer* entity is inherited from *Customer* entity to separate between the MCR’s customers (*OwnCustomer* entity) from the CaRMS Reservation System and Holiday.com’s customers (*Customer* entity) who are instead interacting with the Holiday Reservation System. The use of inheritance also makes sense as *OwnCustomer* has attributes that are common with *Customer*.

We did not include a *RentalDay* entity as we felt the association was unnecessary for us to achieve the use case functionality. Instead, we calculate the number of days between using Java’s GregorianCalendar API methods.

**Rationale of Attributes Used**

All entity classes have their respective IDs as a Long object.

For price of *RentalReservation* entity, we used the BigDecimal object because it would help maintain accuracy when dealing with money.

All dates are stored as simple Java Date objects as it is easier to manipulate. We used DateFormatter and GregorianCalendar when it was necessary to receive input from the client side and parse it in a readable format. When dealing with the corresponding web service methods, we used a XMLGregorianCalendar format instead.

We used a *CarStatusEnum* enumeration to record the state of a car as a Boolean would not be appropriate (more than two states). The three possible states are AVAILABLE, ON\_RENT and REPAIR.

**Description of Relationships**

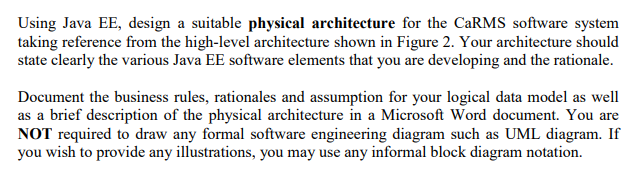
The entity class relationships were modelled after the business requirements stated in the project specifications document. Each entity class has its corresponding session bean (all stateless session bean methods) to perform CRUD tasks when necessary as well as some more complex use cases (such as retrieveCheapestRentalRate and searchCarByCategory) which requires traversal across multiple relationships.

The multiplicity notations indicate the relationships that we have developed and are correspondingly annotated in the submitted code (e.g., @ManyToOne, @OneToOne).

The business rules and assumptions associated with our logical data model are further extrapolated upon in **4. Business Requirements Analysis**.

# Physical Data Model

put the database schema



Some information on physical data model (replace some things from logical data model, Damien do take a look): <https://www.1keydata.com/datawarehousing/physical-data-model.html>

# **Business Requirements Analysis**

## CaRMS Management System

### Business Rules, Assumptions & Rationales

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Use Case** | **Business Rules** | **Business Assumptions** |
| 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. | * Assumes that allocation is done at 2AM daily |
| 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. | * Assumes that transit can still be done outside of the outlet’s operation hours |
| 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. | - |
| 11 | View Rental Rate Details | * View the details of a particular car rental rate record 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. | * Rental rate record does not need to be deleted even if not used. |
| 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 (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. | * Model record does not need to be deleted even if not used. |
| 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. | * Car record does not need to be deleted even if not used. |
| 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. | - |
| 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. | * Customer pays upfront to customer service executive. |
| 27 | Return Car | * Record a customer returning a car. * Status and location of the car must be updated. | - |

## CaRMS Reservation System

### Business Rules, Assumptions & Rationales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S/N** | **Use Case** | | **Business Rules** | **Business Assumptions** |
| 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. | * Customer data input is valid (e.g., email is a valid email format) |
| 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. | - |
| 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. | * Searching can be done by either car category or model, but not both at the same time. |
| 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. | * If the number of cars queried from the database is more than the number of rental reservations queried, the user is able to reserve a car. * Assumes that credit card details provided by the customer is always valid. |
| 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. | - |

## Holiday Reservation System

### Business Rules, Assumptions & Rationales

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Use Case** | **Business Rules** | **Business Assumptions** |
| 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. | * Searching can be done by either car category or model, but not both at the same time. |
| 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. | * If the number of cars queried from the database is more than the number of rental reservations queried, the user is able to reserve a car. * Assumes that credit card details provided by the customer is always valid. |
| 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. | - |