Saint Mary's University – Master's in Computing and Data Analytics

Car Rental Project

Software Requirements Specification

Version <1.0>

By: Mehar Singh
A00434701

Table of Contents

1. Introduction:

1.1. Purpose:

The Software Design Document is made with the purpose of outlining the software architecture and design of Car Rental System in detail. This document will provide the developer with several views of system design. It focuses on the high-level view of the architecture of our system, and on the interaction between the user and the system.

1.2. Scope:

This document will explain how the system is going to accomplish the functional and non-functional requirements as mentioned in SRS document. The document will explain the framework to the programmers by describing the high-level components and architecture, database design and algorithm design.

1.3. Definitions, Acronyms and Abbreviations:

SRS	System Requirement Document
EER	Enhanced Entity-Relationship
IIS	Internet Information Services
SQL	Structured Query Language
MS	Microsoft
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet

1.4. References:

1. https://www.sciencedirect.com/topics/computer-science/context-diagram
2. https://www.edrawsoft.com/architecture-diagram.php

1.5. Overview:

The next chapters of the document will give us the information regarding System characteristics,System Architecture, Infrastructure Services, System context, Standards and Conventions, Software Development Tools while the 3rd chapter will explain the description of each component.

2 System Overview:

The Elite Car Rental Project allows the user to book car by adding their location and pickup date. The user is supposed to create its profile once by providing its information like name, contact number and driver’s licence. While booking the car the system matches the user’s location and all the filter options entered by the user. It then provides all the possible car options available at that location. The user then has an option to add extra features to the car like child's seat, winter tyres etc. Once the booking is completed the user also gets an option to modify or cancel the booking.

2.1 System Characteristics:

2.1.1 User Friendly Interface: The system allows the user to book cars by simply putting in the dates and the pickup location.

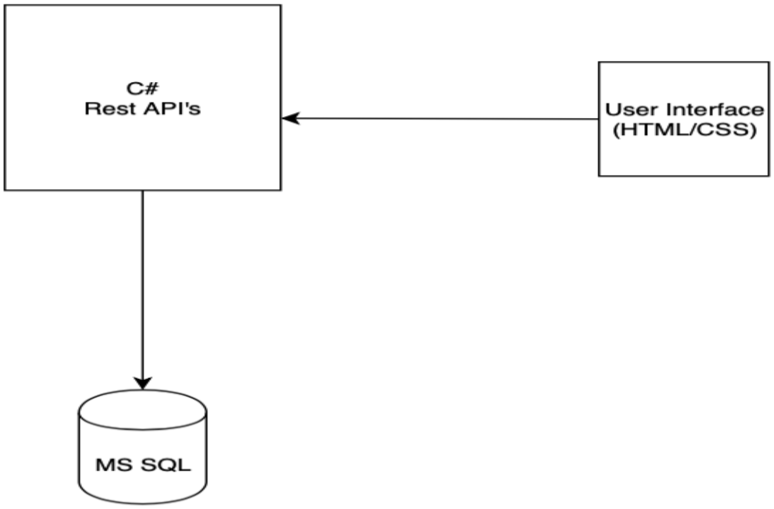
2.1.2 Easily Maintainable and Scalable in future: The system has clearly defined schema and user interface. The project follows MVC framework therefore additional filter options and tables can be easily added to the project in future.

2.1.3 The system uses entity framework therefore it can handle up to hundreds of users.

2.2 System Architecture:

2.2.1 Architecture Diagram:

The Architecture diagram represents the overall structure of the system is going to be implemented.



2.3 Infrastructure Services:

Hardware Specifications:

- Apple MacBook Pro- 3.1 GHz Dual-Core Intel Core i5
- MSSQL server

The system also fulfills the following requirements:

Error Handling: The application provides the necessary error handling for the app to work properly.

Logging: The application logs the necessary details to monitor the performance.

Scalable: The application is scalable as it can handle hundreds of users at a time.

2.4 System Context:

A zero-level context diagram shows the relationship of the system with the external entities. Following is the zero-level context diagram for the system of our project:

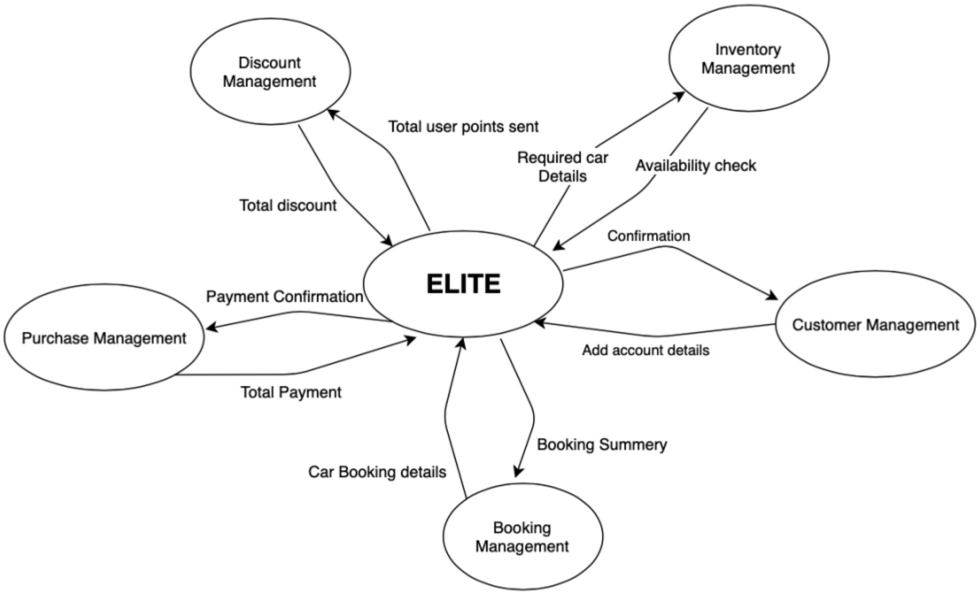


Figure: Zero level Diagram

A First level context diagram depicts basic modules in the system and the flow of data among various modules. Following is the first-level diagram for our project:

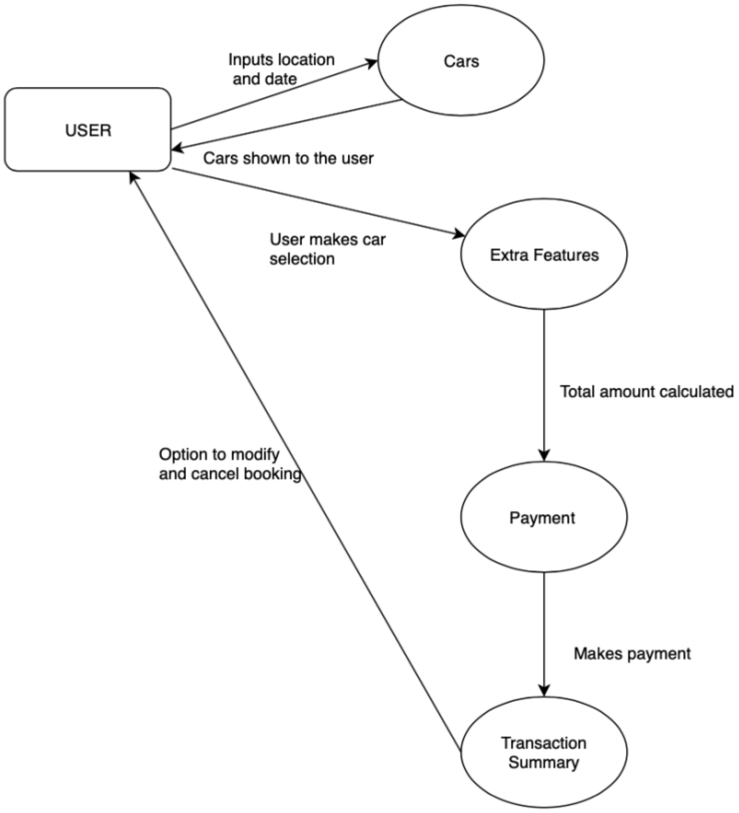


Figure: First level diagram

2.5 Standards and Conventions:

The coding conventions helps to give a consistent look to code which helps the user focus more on the content than the layout. This project will follow the following conventions:

Layout Conventions: Only one statement per line is written and only one declaration is made per line. At least one line is added between method definition and property definition.

Comment Convention: The comments are added in the separate line and not on the same line of the code. The comments end with a period at the end.
Exceptional Handling: The try-catch block is used for handling the exceptions.
Unsigned Data Type: In general, int is used rather than unsigned type, since the use of int is common in C#.
Naming Convention: Camel case terminology is adopted while doing the document.

2.6 Software Development Tools:

2.6.1 SQL Server Management Studio
2.6.2 Microsoft Visual Studio
2.6.3 GIT
2.6.4 IIS

2.7 Outstanding Issues:

2.7.1 Since we don’t have access to the bank’s database, the app wouldn’t be able to confirm if the information entered by the user is correct or not.
2.7.2 The system can store limited data due to limited storage space provided by the server.

3 Component Description:

3.1 Identification: Sign up

3.1.1 Type: Application

3.1.2 Purpose: This module is required to store the user information into database.

3.1.3 Function: The sign-up element brings the user to the sign-up screen where user enters its Email ID, Password, Full name, Last name, Contact number, Driver’s licence. This information is stored in the Account_details table. This information can later be retrieved from the database once the user sign-in.

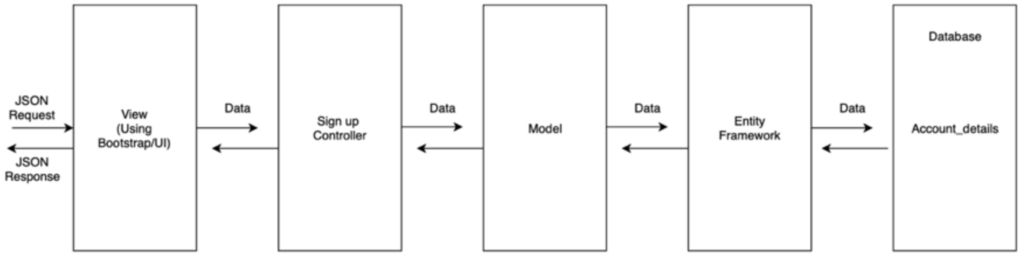
3.1.4 Interface:

The request is sent on the form of JSON request

JSON Request

```
{
    Email: “ ”,
    Password: “ ”,
    Confirm_password: “ ”,
    First_name: “ ”,
    Last_name: “ ”,
    Phone_number: “ ”
```

}



The response is received in the form of JSON object.

JSON Response
{
Status: “ ”
}

The status can be returned either as true or false depending upon the details entered by the user.

3.2 Component Identifier: Sign in

3.2.1 Type: Application

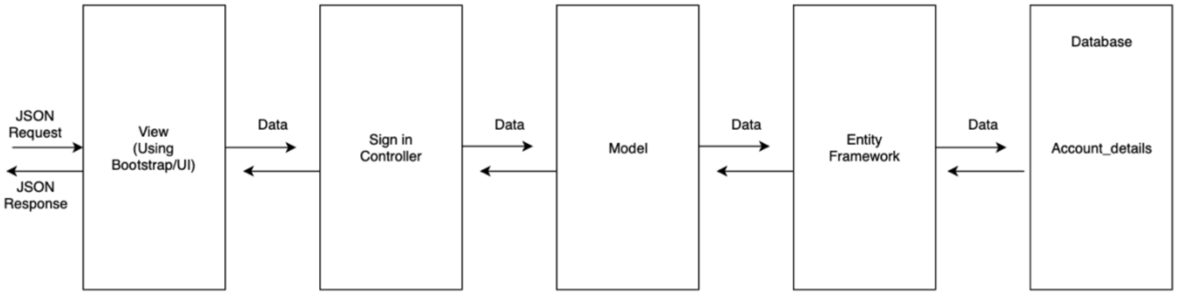
3.2.2 Purpose: This component allows the registered user to save time by directly accessing the information entered by the user before.

3.2.3 Functions: The sign-in element helps the user to access the portal which retrieves the user details from the database which was stored at the time of sign-up. This includes things like First name, last name, contact number and points the user have collected so far.

3.2.4 Interface: The user sends the request in form of JSON request

```
JSON Request {

        Email: “ ”,
        Password: “ ”
        }
```



JSON Response {
Status: “ ”
}

The status can be true or false depending upon the details entered by the user.

3.3 Component Identifier: Car Booking

3.3.1 Type: Application

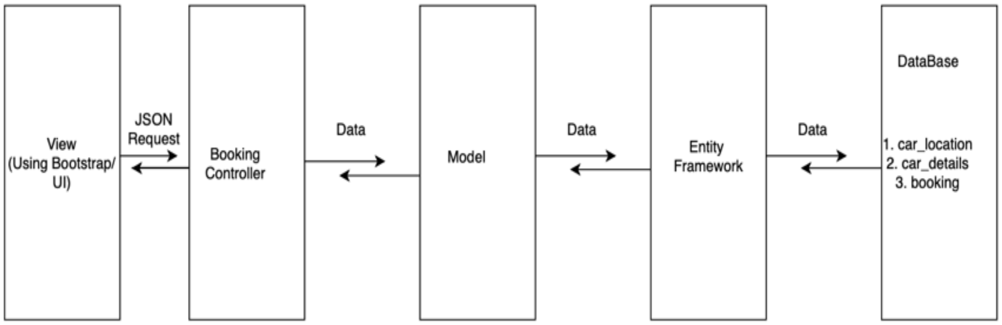
3.3.2 Purpose: This module provides the customer an option to have a look at the options available on the basis of location and dates input by the user.

3.3.3 Function: The Car Booking element helps the user to find the details of all the cars and its details present at the location entered by the user. The user is also given options to add extra features to their car if they like.

3.3.4 Interface:

The request is sent in the form of a JSON object

```
JSON Request
{
    pickUpLoc: “ ”,
    pickupDate: “ ”,
    DropOffDate: “ ”
}
```



The response is sent in the form of JSON object

```
JSON Response
{
    Car_model: " ",
    Vehicle_type: " ",
    Passenger_type: " ",
    Price: " ",
    Color: " "
}
```

3.4 Component Identifier: Payment

3.4.1 Type: Application

3.4.2 Purpose: This module provides the customer to pay via card. The customer is supposed to enter the card details to pay.

3.4.3 Function: The Payment module receives the card information to make the transaction. The required validation test is made, and an error is shown in case the validation rules are not satisfied. The information entered by the user is supposed to be verified with the bank's database but due to the inaccessibility of that we will not make the backend framework for this component, therefore, system assumes that the card information entered by the user is correct and makes the transaction.

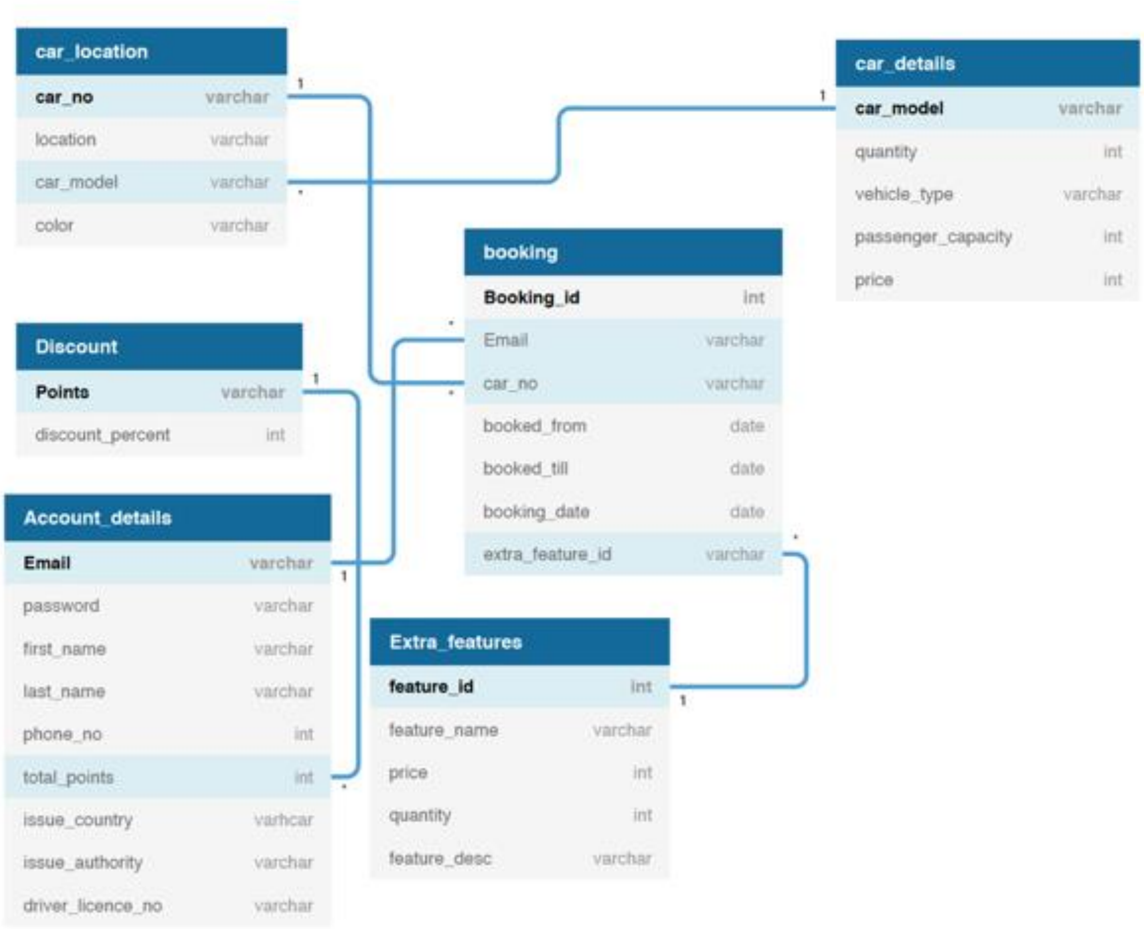
3.5 Component Identifier: Confirmation

3.5.1 Type: Application

3.5.2 Purpose: The confirmation module will show the summary of the car booked by the user.

3.5.3 Function: The module accesses the booking database and show the summary. It provides an option to modify and delete the booking. In case of delete, the entry is removed from Booking table and in case of modify, the entry is replaced by the new one. The effect is accordingly reflected in Account_details table.

4 Class Diagram



car_location
car_no varchar
location varchar
car_model varchar
color varchar
car_details
car_model varchar
quantity int
vehicle_type varchar
passenger_capacity int
price int
booking
Booking_id int
Email varchar
car_no varchar
booked_from date
booked_till date
booking_date date
extra_feature_id varchar
Discount
Points varchar
discount_percent int
Account_details
Email varchar
password varchar
first_name varchar
last_name varchar
phone_no int
total_points int
issue_country varhcar
issue_authority varchar
driver_licence_no varchar
Extra_features
feature_id int
feature_name varchar
price int
quantity int
feature_desc varchar
1
*
1
*
1
*
1
*
1
*