**Parking Lot Application:**

The Parking Lot application helps the customers to manage parking at the garage. The customer will be issued a ticket based on the availability of parking spots in the garage. When they exit the customer is charged according to duration of parking in the garage which he/she needs to pay.

**Roles:**

Customers: People who use the Parking Lot application for managing the parking at the parking garage

**Use Cases:**

**UC1:** The customer wants to know whether parking is available or not

**UC2:** The customer wants to get a ticket ID for the parking spot

**UC3:** The customer wants to know the amount owed for the parking spot

**UC4:** The customer wants to pay the ticket using credit card

**Functional Design Cases:**

**FD1:** Ability to check the availability of the parking spot

**FD2:** Ability to issue a parking spot with a ticket ID

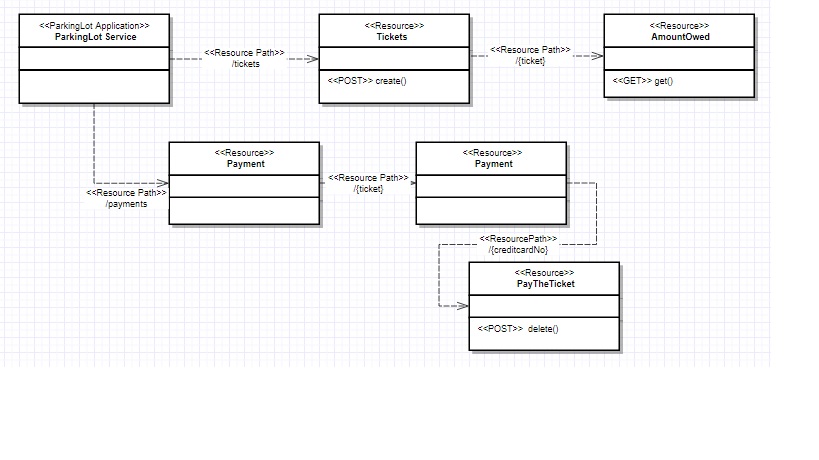
**FD3:** Ability to calculate the amount owed by the customer

**FD4:** Ability to get the payment through credit card

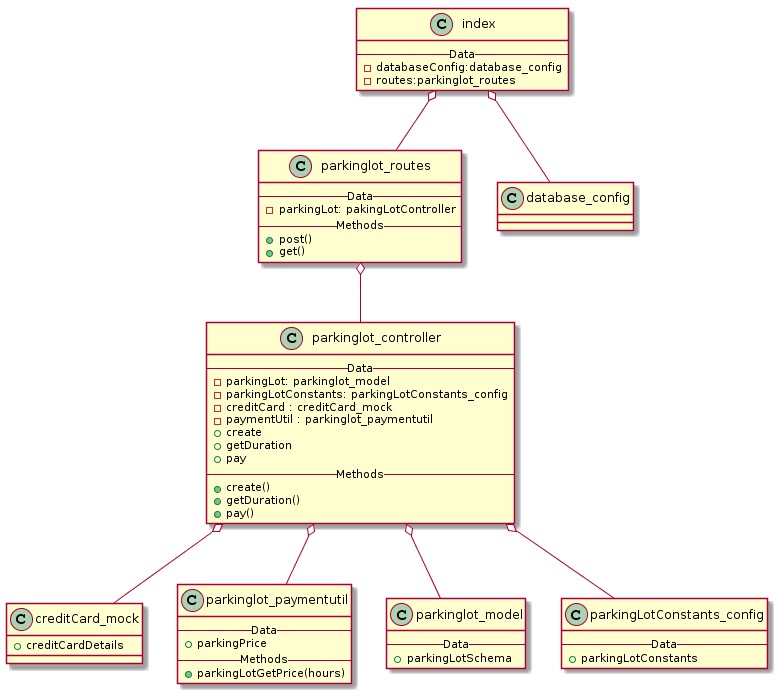
Mapping Function Design Cases to the Use Cases:

|  |  |  |
| --- | --- | --- |
| **S.No** | **Functional Design Case** | **Use Case ID** |
| 1. | Ability to check the availability of the parking spot | UC1 |
| 2. | Ability to issue a parking spot with a ticket ID | UC2 |
| 3. | Ability to calculate the amount owed by the customer | UC3 |
| 4. | Ability to get the payment through credit card | UC4 |

**REST Resource Diagram:**



**Class Diagram:**

****

**Design Cases:**

**1. Ticket ID generation**

**Design Case 1:** Use the MongoDB object ID as ticketID

**Design Case 2:** Generate a ticket ID with timestamp

**Ex:** ticket\_1519211809934

Note: DC2 is preferred as it makes ticket ID more readable

**2. Calculate the parking price**

**Design Case 1:**

/\*\*

\* Module for calculating parking price

\*/

let initialRateLevel= require(‘parkingLotConstants.js’);

calculateParkingPrice(hours){

if(hours <1){

//calculation

}

else if(hours >1 && hours <=3){

//calculations

}

else if(hours >3 && hours <=6){

//calculations

}

else if(hours>6 && hours <=24){

//calculations

}

else{

//calculations

}

}

**Design Case 2:**

/\*\*

\* Module for calculating parking price

\*/

calculateParkingPrice(hours){

let intialPriceRate = 3;

if(hours <1){

//calculation

}

else if(hours >1 && hours <=3){

//calculations

}

else if(hours >3 && hours <=6){

//calculations

}

else if(hours>6 && hours <=24){

//calculations

}

else{

//calculations

}

}

Note: DC1 is preferred as declaring constants in a configuration file will help for code maintenance.

**DataBase Schema:**

Collection: ParkingLot

Document: Parking

{

\_id : Object\_Id,

ticket\_id: String,

created\_At: Timestamp,

updated\_At: Timestamp

}