Online Parking Information System

Process Modeling and Data Modeling /   
System Proposal / Analysis Phase   
(Homework No.3)

Project team: Team 07

Instructor: Dr. Araz Yusubov

Submitted in partial fulfillment of the requirements of the INFT 2303: Systems Analysis and Design course project

|  |  |
| --- | --- |
| GitHub repository | https://github.com/ADA-SITE-INFT2303-2023-Spring/sys-dev-project-team-07.git |
| 19.04.23 | ERD (without relations) |
| 20.04.23 | ERD (final version) |
| 21.04.23 | Context and level-0 diagram |
| 22.04.23 | Level-1 diagrams and introduction |
| 23.04.23 | Definitions and CRUD matrix |

| Other documents in the package | |
| --- | --- |
| ERD.drawio | Entity Relationship Diagram of the system |
| Context Diagram.drawio | Context diagram of the system |
| DFD Level 0.drawio | Level 0 diagram of the system |
| DFD Level1(1).drawio | Level 1 DFD for process 1 |
| DFD Level1(2).drawio | Level 1 DFD for process 2 |

|  |  |  |
| --- | --- | --- |
| Team member | Contribution to this homework (NOT the project) | Estimated % |
| Emil Hajiyev | Entity Relationship Diagram and its description | 20% |
| Lala Mahmudova | Level 1 diagrams, definitions, references | 20% |
| Faraz Bagher Nezhad | Context diagram, Bonus 1 | 20% |
| Orkhan Ismayilov | Introduction, CRUD matrix | 20% |
| Jamaladdin Talishinskiy | Level 0 diagram | 20% |

# Table of Contents

Table of Contents

[1. Introduction 3](#_Toc133182167)

[Definitions 3](#_Toc133182174)

[2. Process Modeling 4](#_Toc133182185)

[Context diagram 4](#_Toc133182187)

[Level 0 diagram 4](#_Toc133182191)

[Level 1 diagrams 5](#_Toc133182196)

[3. Data Modeling 6](#_Toc133182209)

[4. References 9](#_Toc133182238)

# Introduction

This is part of the System Proposal for a hypothetical project Online Parking Information System submitted for partial fulfillment of the requirements of the Systems Analysis and Design course in the School of Information Technologies and Engineering at ADA University, Baku, Azerbaijan.

The purpose of the document is to provide detailed information of the different components, processes and data involved in the smart parking system. This includes different sections such as process modeling, level 0 and level 1 diagrams, data modeling and a CRUD matrix.

The process modeling section provides a clear description of the different processes involved in the smart parking system, including registration, payment and processing. These flowcharts and diagrams will show how processes work and interact with each other.

The level 0 and level 1 diagrams provide a high-level view of the smart parking system architecture. These diagrams illustrate different components of a system and their interactions between the data flow diagrams.

The data modeling section provides an overview of the data elements in the smart parking system, including how they are stored, processed, and analyzed.

The CRUD matrix provides an overview of the different operations in the smart parking system, including whether they can be created, read, and updated or deleted. This matrix ensures the system is properly designed and meets requirements in both functionality and security.

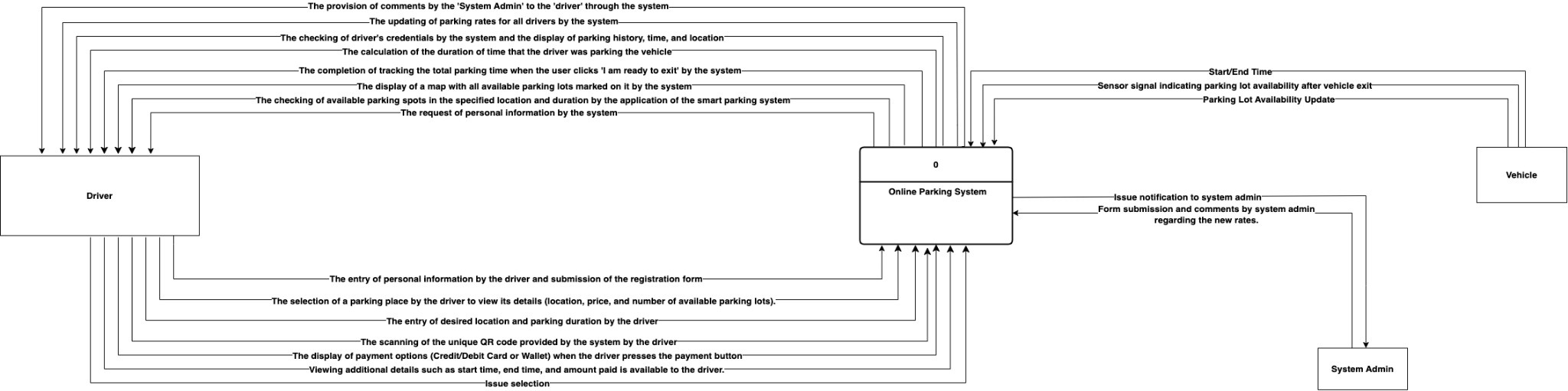
## Definitions

|  |  |  |
| --- | --- | --- |
| Term | Definition |  |
| ​​Parking bollard | Physical devices installed on the ground that can be remotely controlled to allow or restrict access to a specific area. |  |
| Payment processing system | The platform enables businesses to process electronic payments, such as credit card transactions |  |
| Quick Response (QR) code | A type of two-dimensional barcode that can be scanned and read by a smartphone camera or a QR code reader. |  |
| ​​User authentication | User authentication |  |

# Process Modeling

Below is the representation of Context, Level 0 and Level 1 diagrams for the system.

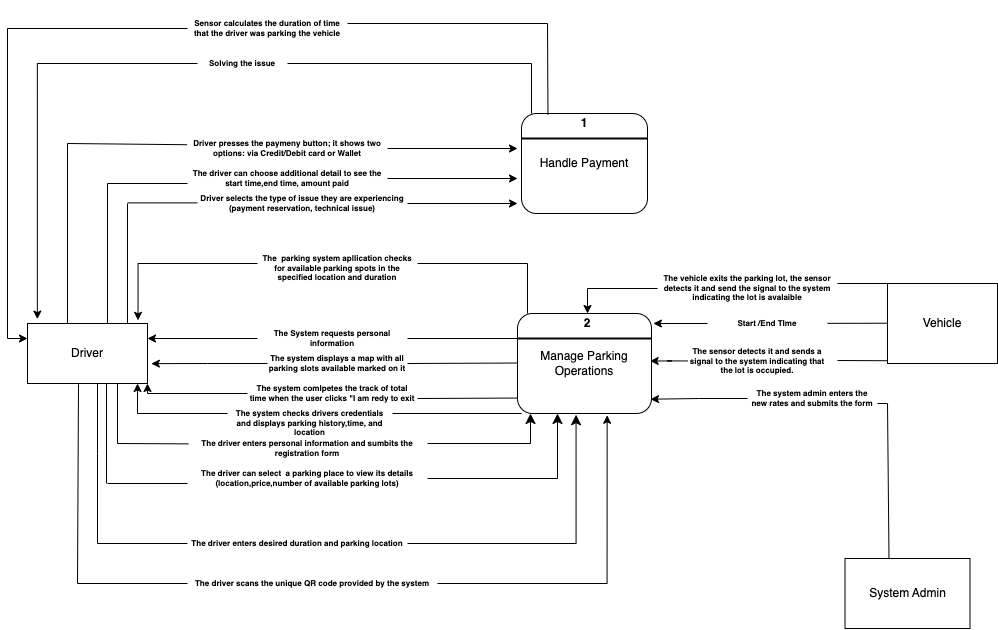
## Context diagram



In the Context diagram, there is one process:

1. Online Parking System: In the Context diagram, the "Online Parking System" process oversees the registration of drivers, Available Parking places, Duration and costs, and support.

## Level 0 diagram



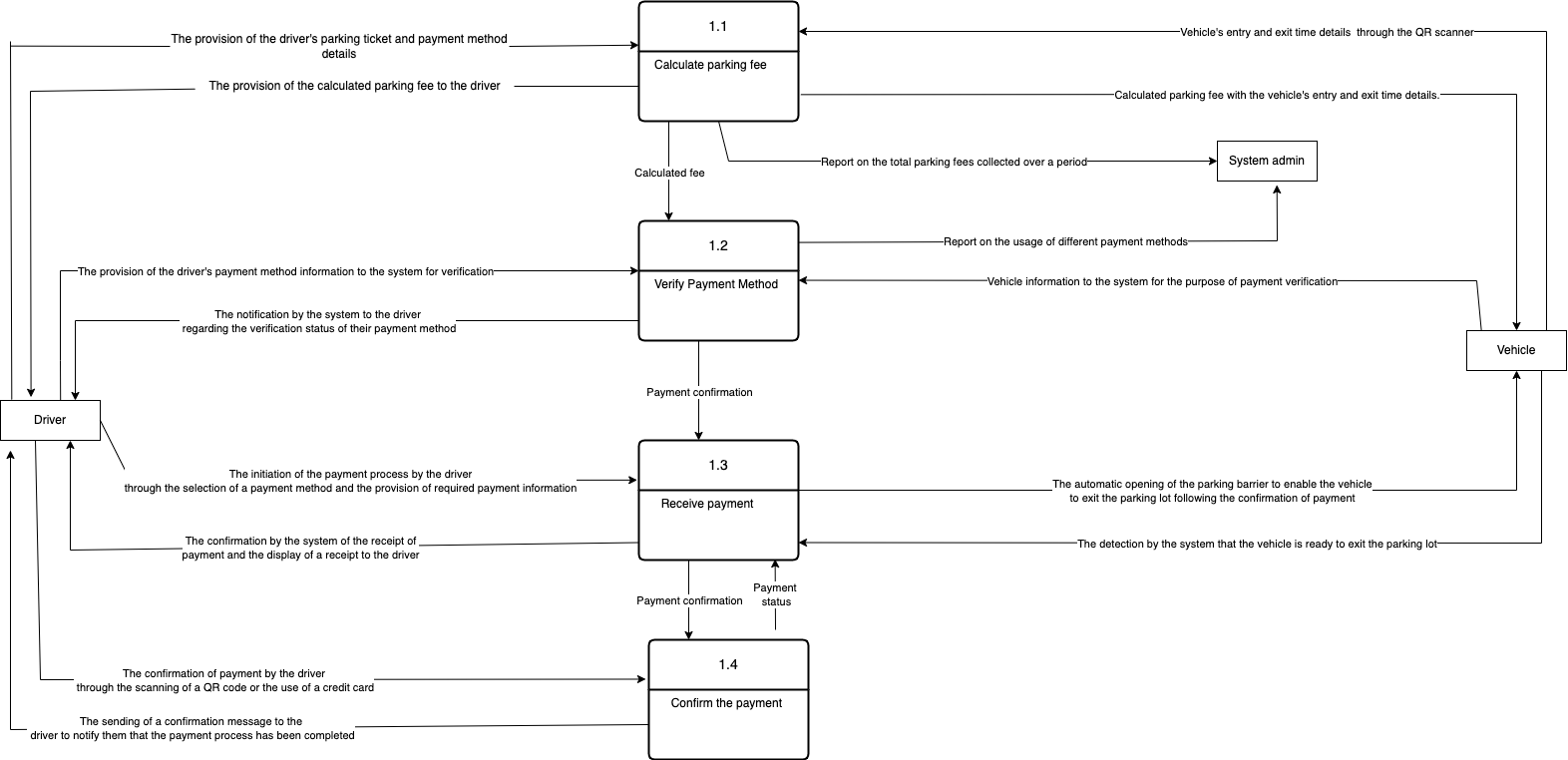
In the Level 0 diagram, there are two processes.

1.Handle payment - In the Level 0 diagram, the "Handle Payment" process oversees handling the payment process from start to end, including verifying the payment method, receiving the payment, and confirming the payment.

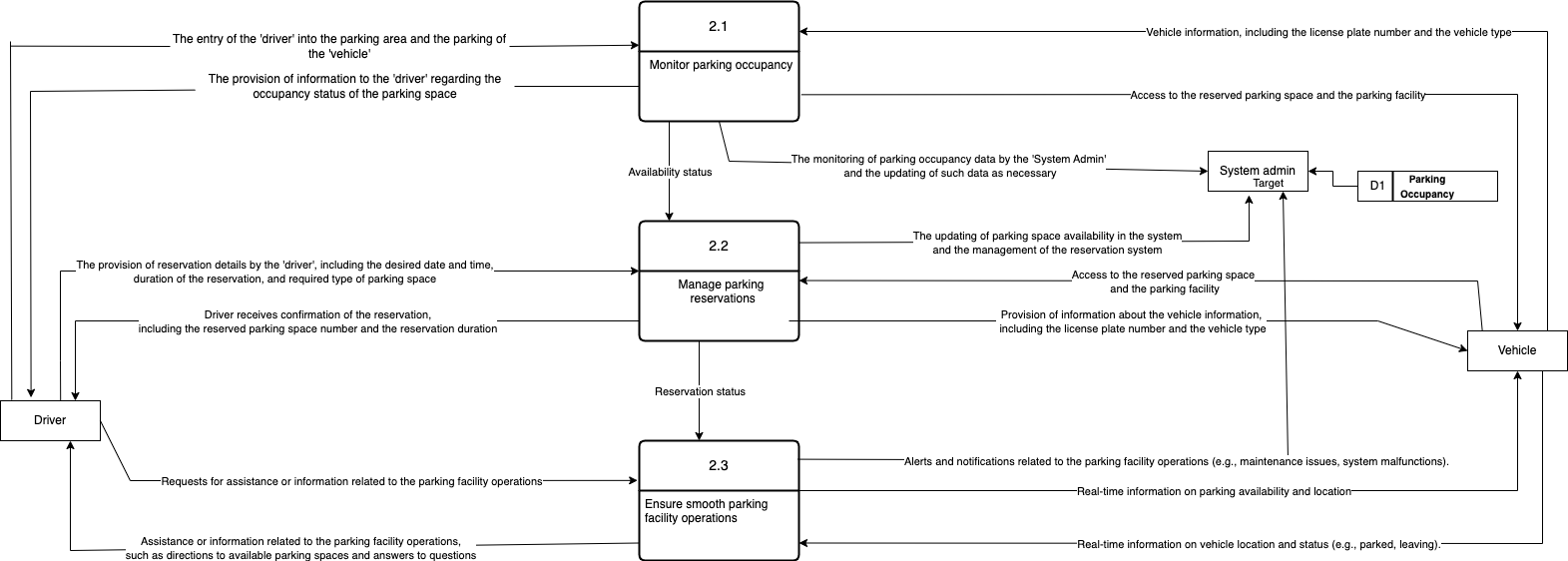
2.Manage Parking operations - In the level 0 diagram, the "Manage Parking Operations" process involves managing overall parking operations, such as monitoring parking occupancy, handling reservations, and ensuring that the parking facility functions smoothly. It is a high-level mechanism that manages all parking-related activities within the facility. The process guarantees that parking activities are carried out efficiently and that any concerns or problems are handled as soon as possible.

## Level 1 diagrams

**Level 1 DFD for Process 1**



**Level 1 DFD for Process 2**



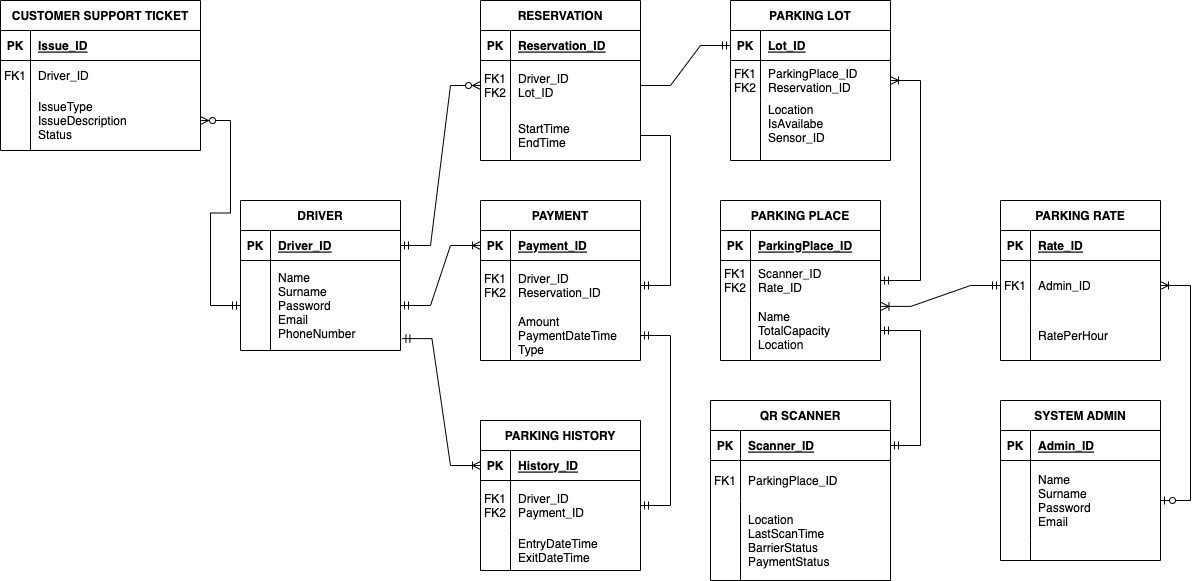
In the Level 1 diagrams, there are following subprocesses:

1. Calculate parking fee (UC-07) - this subprocess is used to calculate parking fee of vehicles.
2. Verify payment method (UC-07) - this subprocess verifies either a confirmation or a rejection of the payment method.
3. Receive payment (UC-07) - this subprocess processes the payment.
4. Confirm the payment (UC-07) - this subprocess verifies that the payment was received and approved.
5. Monitor parking occupancy (UC-02)- this subprocess is about managing the availability of parking slots.
6. Manage parking reservations (UC-03) - this subprocess ensures the available spaces are reserved by drivers.
7. Ensure smooth parking facility operations (UC-10) - this subprocess involves ensuring that the parking facility runs smoothly. This process includes various activities such as managing parking signage, ensuring proper traffic flow, and resolving any issues related to parking operations.

Additionally, there are three entities: Driver, vehicle, and system admin. The external entities that interact with the system are the driver and vehicle. The driver reserves a parking place, parks the car, and pays for the parking through the system. Vehicle information, such as license plate number, is logged in the system to guarantee that only approved cars may park.

The Parking occupancy data store demonstrates the system's storage of information on the number of parking spaces available and occupied in the parking facility at any given moment. It might comprise information such as the number of available parking spaces, the number of spots occupied, the location of the spaces, and any other relevant data required to monitor and manage parking occupancy. This data store is likely to be updated on a frequent basis by various processes that monitor parking occupancy, such as "Monitor Parking Occupancy" and "Manage Parking Reservations".

# Data Modeling

**

Entities used in Entity Relationship Diagram:

1. **Driver**: The driver is one of the primary entities of the system, participating in many processes. Attributes are information about the driver obtained during the registration, such as name, surname, phone number, email, and account password.
2. **Reservation**: It is an event that happens before the driver enters the parking lot, therefore the Reservation entity has the primary keys of the Driver and Parking lot entities as its foreign keys. Additionally, start/end time of reservation should be tracked for payment entity.
3. **Payment**: There is a Driver\_ID attribute as a foreign key because payment is made by the driver. Reservation time must also be paid, therefore Reservation\_ID is a foreign key of the Payment entity. Information about the amount of payment time and the type of payment (cash/online) should be stored.
4. **Parking History**: After payment is done, information about parking will be visible to the driver in the “Parking history” section, therefore Driver\_ID is the foreign key. As it contains information about payment, Payment\_ID is the foreign key of Parking History entity. It will also display the time of entry and exit to the parking lot.
5. **Parking Lot**: Since the parking lot is in the parking place, ParkingPlace\_ID is the foreign key. During reservation, the parking lot is shown as unavailable to the others, therefore Reservation\_ID should be used as a foreign key of the Parking Lot entity. Parking lot has 2 statuses (available/unavailable) which are represented in IsAvailable attribute. Since each lot has one sensor, which affects IsAvailable and Location attributes, Sensor\_ID is an attribute of this entity.
6. **Parking Place**: Parking place consists of many parking lots and usually it is considered as limited ground or underground parking with a barrier at the entrance. Since there is the QR Scanner at the entrance of the parking place, Scanner\_ID is the foreign key of Parking Place entity. Every parking place has its own parking rate, so Rate\_ID is also the foreign key. Name, location, and total capacity of the parking place are attributes of this entity as this information is important for proper working of the system.
7. **QR Scanner**: QR Scanner helps to identify the driver’s account at the entrance, and it monitors if the driver paid or not at the exit. Location of this QR Scanner should be visible in the system as it helps the driver to find the entrance of the parking place on the map quickly. If the barrier is open, QR Scanner will not identify the user and will wait until the barrier status changes. Using this information, we get the following attributes: ParkingPlace\_ID as the foreign key, Location, Last Scan Time, Barrier Status, and Payment Status.
8. **Parking Rate**: Parking Rate for parking place is managed by system admin so Admin\_ID is the foreign key. Additionally, this entity has a Rate per Hour attribute.
9. **System Admin**: System Admin has basic attributes which contain personal data such as name, surname, email, and account password.
10. **Customer Support Ticket**: It is a special section in the system, where drivers can address their issues and get feedback. Driver indicates the type of issue (e.g payment issue, time tracking error) and issue description. Status of issue will change after resolving the problem.

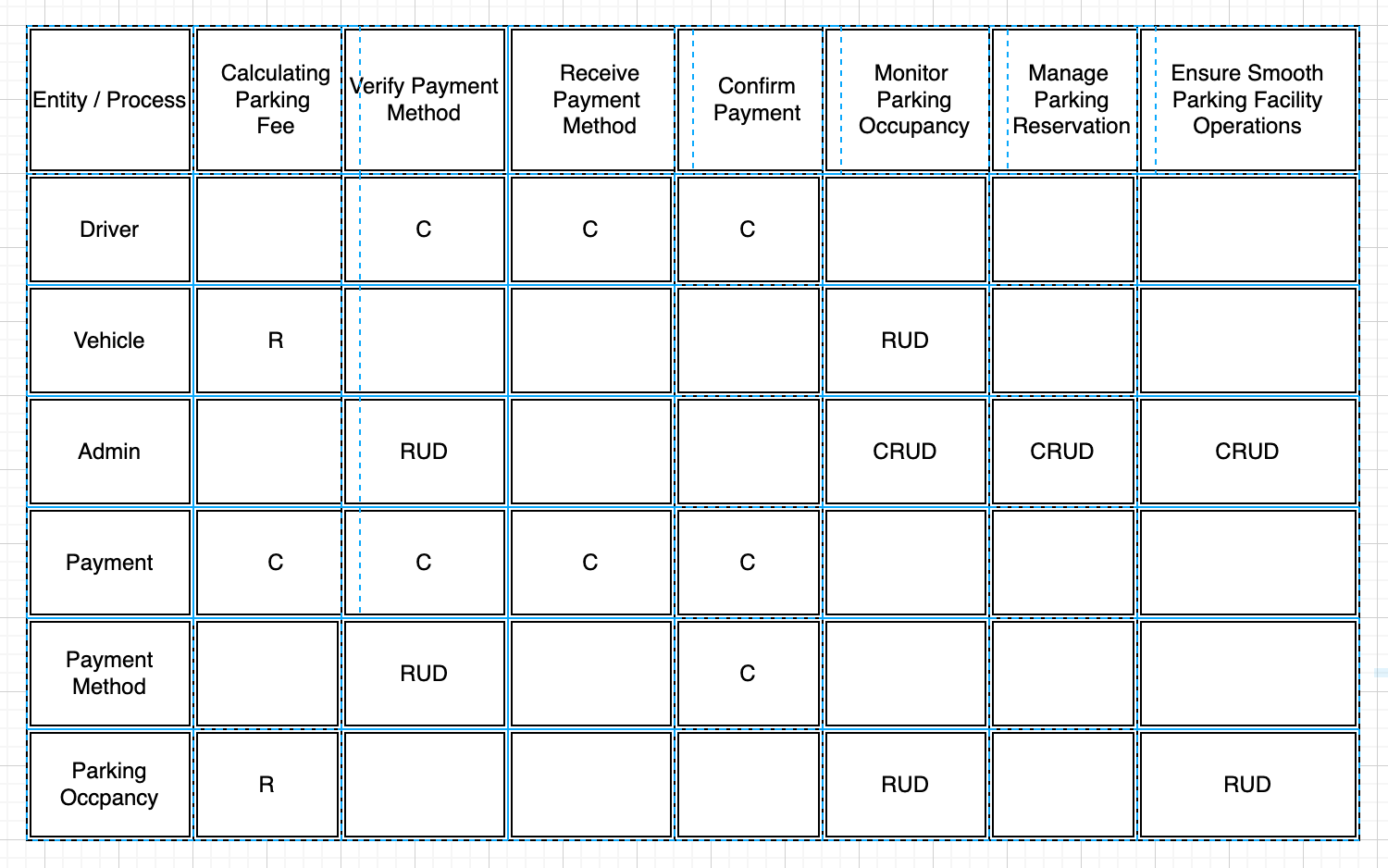
Relations between entities:

1. **Driver - Reservation:** One driver can have many reservations, but each reservation belongs to only one driver. The driver may not reserve parking. (One to Many, Mandatory to Optional)
2. **Driver - Payment:** One driver can have many payments, but each payment belongs to only one driver. (One to Many, Mandatory to Mandatory)
3. **Driver - Parking History:** One driver can have multiple parking histories associated with them, but each parking history is associated with only one driver. (One to Many, Optional to Mandatory)
4. **Driver - Customer Support Ticket:** Each driver can have many customer support tickets, but one customer support ticket belongs to only one driver. Driver may not have customer support ticket. (One to Many, Mandatory to Optional)
5. **Reservation - Payment:** One payment belongs to only one reservation and each reservation can have only one payment. Payment must be made even if there is no reservation. ( One to One, Optional to Mandatory)
6. **Reservation - Parking Lot:** One reservation belongs to only one parking lot and each parking lot can have only one reservation. There may not be a reservation on the parking lot. (One to One, Optional to Mandatory)
7. **Parking Place - Parking Lot:** One parking place can have many parking lots, but each parking lot belongs to only one parking place. (One to Many, Mandatory to Mandatory)
8. **Parking Place - QR Scanner:** One parking place can have only one QR Scanner and each QR Scanner belongs to one parking place. (One to One, Mandatory to Mandatory)
9. **Payment - Parking History:** Payment can be linked to one parking session and each parking session can only have one payment. (One to One, Mandatory to Mandatory)
10. **Parking Rate - Parking Place:** Each parking rate can belong to many parking places, but one parking place can have only one parking rate (One to Many, Mandatory to Mandatory)
11. **System Admin - Parking Rate:** One system admin can set many parking rates, but each parking rate is set by only one system admin. System admin may not set parking rate. (One to Many, Optional to Mandatory)

## CRUD matrix

This CRUD matrix helps to ensure that all the necessary interactions between processes and entities are accounted for, and that the system is designed to support the required functionality and security requirements.

The interactions between the processes and the entities are as follows:



# References

1. Abhishek, K. (2021). *Smart parking management system* . ResearchGate. Retrieved from: https://www.researchgate.net/publication/353924045\_Smart\_Parking\_Management\_System
2. Whitten, J. L., Bentley, L. D., & Barlow, J. D. (2018). The systems analyst and information systems development. Cengage Learning