**CS631-103 Term Project  
  
Car Rental System**

|  |  |
| --- | --- |
| **Phase 3** | |
| **Group 26:** | Jairo A. Perez  Manavkumar  Patel |
| **Date:** | 11/18/2022 |

Table of Contents

[**PART 1 – Business Requirements:** 3](#_Toc121777953)

[Customer requirements: 3](#_Toc121777954)

[Reservation requirements: 3](#_Toc121777955)

[Location requirements: 3](#_Toc121777956)

[Car requirements: 3](#_Toc121777957)

[Model Requirements: 3](#_Toc121777958)

[Car Class Requirements: 3](#_Toc121777959)

[Agreement Requirements: 3](#_Toc121777960)

[Invoice Requirements: 4](#_Toc121777961)

[a. Comparing Reservation’s rental period and Agreement’s return date 4](#_Toc121777962)

[b. Extra days: 4](#_Toc121777963)

[c. Rates: 4](#_Toc121777964)

[d. Late Fee: 4](#_Toc121777965)

[**PART 2 – ER Diagram:** 5](#_Toc121777966)

[**PART 3 – Rational Schema Diagram:** 6](#_Toc121777967)

[**PART 4 – Application Program Design** 7](#_Toc121777968)

[**Annex A** 8](#_Toc121777969)

[SQL Statements: 8](#_Toc121777970)

[SQL Execution Proof: 11](#_Toc121777971)

[**Annex B** 12](#_Toc121777972)

[Requirements: 12](#_Toc121777973)

[SQL Statements: 12](#_Toc121777974)

[SQL Execution Proof: 13](#_Toc121777975)

[**Annex C** 16](#_Toc121777976)

[Requirements: 16](#_Toc121777977)

[**Annex D** 17](#_Toc121777978)

[SQL Queries: 17](#_Toc121777979)

[a. Group By 17](#_Toc121777980)

[b. Group By & Having 17](#_Toc121777981)

[c. In 17](#_Toc121777982)

[d. All 18](#_Toc121777983)

[**PART 8 – Version Control** 19](#_Toc121777984)

[**PART 9 – Version Control** 20](#_Toc121777985)

# **PART 1 – Business Requirements:**

The Following are the business requirements for the Car Rental system. These requirements will be grouped into sections to provide a better understanding from each component.

## Customer requirements:

1. A customer must make a reservation prior to rent a car.
2. If customer decides to proceed with the rental, she/he will get a rental agreement.
3. The customer must provide name, address, and phone at reservation time.
4. Lastly, the customer must provide a valid driver’s License and credit card if she/he gets a rental agreement.

## Reservation requirements:

1. The reservation must specify a type of class car.
2. The reservation will be made for a specific pick-up and drop-off location.
3. Pick-up and drop-off location do not have to be the same but can.
4. The reservation will include a rental period. This period consists of the start date and time and end date and time.
5. Reservation will have a type. To know if customer reserved by phone, walk-in or online. \*
6. The reservation will have a status. \*
7. Status will be active if customer shows up at pick-up location on start date.
8. Reservation will be cancelled if customer does not appear on start date and/or calls to cancel.

## Location requirements:

1. A location can hold several cars.
2. Not all locations will have the same number of cars or classes of cars.
3. Each location is associated with a reservation either in pick-up or drop-off form.

## Car requirements:

1. A car must be either at a location or with a customer (under a rental agreement).
2. If a car is rented it must be associated with a rental agreement.
3. Once a car is returned it becomes available for a new rental agreement.
4. A car is associated with a specific model

## Model Requirements:

1. A model has several cars associated to it.
2. A model is associated with a single car class.

## Car Class Requirements:

1. A class car has several models associated to it.
2. A class car has several cars associated to it.
3. Each reservation has a car class specified.

## Agreement Requirements:

1. Getting a rental agreement is only possible if there is a reservation.
2. Agreements can only be made to one person.
3. Agreements can only be associated with one specific car.
4. An agreement’s rental period is based on reservation rental period.
5. If a car is returned earlier, the Agreement will still base its rental period of the reservation.
6. If a car is returned late (after reservation rental period) the Agreement will save its new return date and time.
7. The agreement specifies the car’s initial odometer reading.
8. Once the car is return the agreement will store the ending or returning odometer reading.

## Invoice Requirements:

1. An invoice must be associated with a rental agreement.
2. Invoices will have a status. This indicate if it has been paid or not.
3. The invoice will calculate its amount due based on:
   1. Comparing Reservation’s rental period and Agreement’s return date.
      1. If return dates are equal no extra fees are applied
      2. If Agreement’s return date is less than Reservation’s, amount is based on Reservation’s period.
      3. If Agreement’s return date is older than Reservation’s, amount is based on Reservation’s period, extra days, and late fees.
   2. Extra days:
      1. Extra days are solely charge on daily rate
   3. Rates:
      1. Car rates are based on the car class selected
      2. If a reservation period is a week period, the weekly rate is applied.
      3. If a reservation period is less than a week period, the daily rate is applied.
      4. If a reservation period is more than a week period:
         1. A 7-day period will have a weekly rate
         2. Days outside of a week period will be charge with a daily rate.
   4. Late Fee:
      1. A late fee is 15% of a daily rate for a type of class car.
      2. A late fee will be applied to every single extra day.

# **PART 2 – ER Diagram:**

Below you can find the Entity Relationship diagram based on the business requirements.

Diagram

Description automatically generated

# **PART 3 – Rational Schema Diagram:**

Diagram

Description automatically generatedBelow you can find the Relational Schema based on the ER diagram.

# **PART 4 – Application Program Design**

# **PART 5 – Normalization**

# **Annex A**

**From Phase 2: PART 4 – Database and Table Creation**

## SQL Statements:

Below please find SQL statements used to create the database (NJIT\_cs631103\_carproject) and the appropriate tables for the database.

These statements were created based on the design from Phase 1.

-- -----------------------------------------------------

-- Schema NJIT\_cs631103\_carproject

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `NJIT\_cs631103\_carproject` DEFAULT CHARACTER SET utf8;

USE `NJIT\_cs631103\_carproject`;

-- -----------------------------------------------------

-- Table `NJIT\_cs631103\_carproject`.`ClassCar`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `NJIT\_cs631103\_carproject`.`ClassCar` (

  `Name` VARCHAR(45) NOT NULL,

  `RateWeekly` DECIMAL(4) NOT NULL,

  `RateDaily` DECIMAL(4) NOT NULL,

  PRIMARY KEY (`Name`))

UNIQUE INDEX `Name\_UNIQUE` (`Name` ASC) VISIBLE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `NJIT\_cs631103\_carproject`.`CarModel`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `NJIT\_cs631103\_carproject`.`CarModel` (

  `carmodelID` INT NOT NULL AUTO\_INCREMENT,

  `Year` INT NULL DEFAULT NULL,

  `Name` VARCHAR(45) NULL DEFAULT NULL,

  `Make` VARCHAR(45) NULL DEFAULT NULL,

  `ClassCar\_Name` VARCHAR(45) NOT NULL,

  PRIMARY KEY (`carmodelID`),

  UNIQUE INDEX `fk\_CarModel\_ClassCar1\_idx` (`ClassCar\_Name` ASC) VISIBLE,

  CONSTRAINT `fk\_CarModel\_ClassCar1`

    FOREIGN KEY (`ClassCar\_Name`)

    REFERENCES `NJIT\_cs631103\_carproject`.`ClassCar` (`Name`)

    ON DELETE RESTRICT

    ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `NJIT\_cs631103\_carproject`.`Car`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `NJIT\_cs631103\_carproject`.`Car` (

  `VIM` VARCHAR(100) NOT NULL,

  `Color` VARCHAR(45) NOT NULL,

  `ClassCar\_Name` VARCHAR(45) NOT NULL,

  `CarModel\_carmodelID` INT NOT NULL,

  PRIMARY KEY (`VIM`),

  UNIQUE INDEX `VIM\_UNIQUE` (`VIM` ASC) VISIBLE,

  INDEX `fk\_Car\_ClassCar1\_idx` (`ClassCar\_Name` ASC) VISIBLE,

  INDEX `fk\_Car\_CarModel1\_idx` (`CarModel\_carmodelID` ASC) VISIBLE,

  CONSTRAINT `fk\_Car\_ClassCar1`

    FOREIGN KEY (`ClassCar\_Name`)

    REFERENCES `NJIT\_cs631103\_carproject`.`ClassCar` (`Name`)

    ON DELETE RESTRICT

    ON UPDATE CASCADE,

  CONSTRAINT `fk\_Car\_CarModel1`

    FOREIGN KEY (`CarModel\_carmodelID`)

    REFERENCES `NJIT\_cs631103\_carproject`.`CarModel` (`carmodelID`)

    ON DELETE RESTRICT

    ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `NJIT\_cs631103\_carproject`.`Location`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `NJIT\_cs631103\_carproject`.`Location` (

  `LocationID` INT NOT NULL AUTO\_INCREMENT,

  `Phone` INT NOT NULL,

  `AStreet` VARCHAR(45) NOT NULL,

  `ACity` VARCHAR(45) NOT NULL,

  `AState` VARCHAR(2) NOT NULL,

  `AZip` VARCHAR(4) NOT NULL,

  `Car\_VIM` VARCHAR(100) NOT NULL,

  PRIMARY KEY (`LocationID`),

  UNIQUE INDEX `fk\_Location\_Car1\_idx` (`Car\_VIM` ASC) VISIBLE,

  CONSTRAINT `fk\_Location\_Car1`

    FOREIGN KEY (`Car\_VIM`)

    REFERENCES `NJIT\_cs631103\_carproject`.`Car` (`VIM`)

    ON DELETE RESTRICT

    ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `NJIT\_cs631103\_carproject`.`Reservation`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `NJIT\_cs631103\_carproject`.`Reservation` (

  `ReservationID` INT NOT NULL AUTO\_INCREMENT,

  `Status` VARCHAR(45) NOT NULL,

  `StartDate` DATE NULL DEFAULT NULL,

  `StartTime` TIME NULL DEFAULT NULL,

  `EndDate` DATE NULL DEFAULT NULL,

  `EndTime` TIME NULL DEFAULT NULL,

  `Type` VARCHAR(45) NULL DEFAULT NULL,

  `Pickup\_LocationID` INT NOT NULL,

  `Return\_LocationID` INT NOT NULL,

  `ClassCar\_Name` VARCHAR(45) NOT NULL,

  PRIMARY KEY (`ReservationID`),

  UNIQUE INDEX `ReservationID\_UNIQUE` (`ReservationID` ASC) VISIBLE,

  INDEX `fk\_Reservation\_Location1\_idx` (`Pickup\_LocationID` ASC) VISIBLE,

  INDEX `fk\_Reservation\_Location2\_idx` (`Return\_LocationID` ASC) VISIBLE,

  INDEX `fk\_Reservation\_ClassCar1\_idx` (`ClassCar\_Name` ASC) VISIBLE,

  CONSTRAINT `fk\_Reservation\_Location1`

    FOREIGN KEY (`Pickup\_LocationID`)

    REFERENCES `NJIT\_cs631103\_carproject`.`Location` (`LocationID`)

    ON DELETE RESTRICT

    ON UPDATE CASCADE,

  CONSTRAINT `fk\_Reservation\_Location2`

    FOREIGN KEY (`Return\_LocationID`)

    REFERENCES `NJIT\_cs631103\_carproject`.`Location` (`LocationID`)

    ON DELETE RESTRICT

    ON UPDATE CASCADE,

  CONSTRAINT `fk\_Reservation\_ClassCar1`

    FOREIGN KEY (`ClassCar\_Name`)

    REFERENCES `NJIT\_cs631103\_carproject`.`ClassCar` (`Name`)

    ON DELETE RESTRICT

    ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `NJIT\_cs631103\_carproject`.`Customer`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `NJIT\_cs631103\_carproject`.`Customer` (

  `customerID` INT NOT NULL AUTO\_INCREMENT,

  `FName` VARCHAR(45) NOT NULL,

  `LName` VARCHAR(45) NOT NULL,

  `Phone` VARCHAR(45) NOT NULL,

  `LcNumber` VARCHAR(45) NULL DEFAULT NULL,

  `LcState` VARCHAR(2) NULL DEFAULT NULL,

  `CCType` VARCHAR(45) NULL DEFAULT NULL,

  `CCNumber` INT NULL DEFAULT NULL,

  `AStreet` VARCHAR(45) NULL DEFAULT NULL,

  `ACity` VARCHAR(45) NULL DEFAULT NULL,

  `AState` VARCHAR(2) NULL DEFAULT NULL,

  `Azip` VARCHAR(4) NULL DEFAULT NULL,

  `Reservation\_ReservationID` INT NULL,

  PRIMARY KEY (`customerID`),

  UNIQUE INDEX `customerID\_UNIQUE` (`customerID` ASC) VISIBLE,

  INDEX `fk\_Customer\_Reservation\_idx` (`Reservation\_ReservationID` ASC) VISIBLE,

  UNIQUE INDEX `LcNumber\_UNIQUE` (`LcNumber` ASC) VISIBLE,

  CONSTRAINT `fk\_Customer\_Reservation`

    FOREIGN KEY (`Reservation\_ReservationID`)

    REFERENCES `NJIT\_cs631103\_carproject`.`Reservation` (`ReservationID`)

    ON DELETE SET NULL

    ON UPDATE RESTRICT)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `NJIT\_cs631103\_carproject`.`Invoice`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `NJIT\_cs631103\_carproject`.`Invoice` (

  `InvoiceID` INT NOT NULL AUTO\_INCREMENT,

  `Status` VARCHAR(45) NOT NULL,

  `Date` DATE NULL DEFAULT NULL,

  `Taxes` DECIMAL NULL DEFAULT NULL,

  `Total` DECIMAL NULL DEFAULT NULL,

  PRIMARY KEY (`InvoiceID`),

  UNIQUE INDEX `InvoiceID\_UNIQUE` (`InvoiceID` ASC) VISIBLE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `NJIT\_cs631103\_carproject`.`Agreement`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `NJIT\_cs631103\_carproject`.`Agreement` (

  `ContractNumber` INT NOT NULL AUTO\_INCREMENT,

  `StartTime` TIME NULL DEFAULT NULL,

  `ReturnTime` TIME NULL DEFAULT NULL,

  `StartDate` DATE NULL DEFAULT NULL,

  `ReturnDate` DATE NULL DEFAULT NULL,

  `StartOdometer` INT NULL DEFAULT NULL,

  `EndOdometer` INT NULL DEFAULT NULL,

  `Reservation\_ReservationID` INT NOT NULL,

  `Invoice\_InvoiceID` INT NULL,

  `Car\_VIM` VARCHAR(100) NOT NULL,

  PRIMARY KEY (`ContractNumber`),

  UNIQUE INDEX `ContractNumber\_UNIQUE` (`ContractNumber` ASC) VISIBLE,

  INDEX `fk\_Agreement\_Reservation1\_idx` (`Reservation\_ReservationID` ASC) VISIBLE,

  INDEX `fk\_Agreement\_Invoice1\_idx` (`Invoice\_InvoiceID` ASC) VISIBLE,

  INDEX `fk\_Agreement\_Car1\_idx` (`Car\_VIM` ASC) VISIBLE,

  CONSTRAINT `fk\_Agreement\_Reservation1`

    FOREIGN KEY (`Reservation\_ReservationID`)

    REFERENCES `NJIT\_cs631103\_carproject`.`Reservation` (`ReservationID`)

    ON DELETE CASCADE

    ON UPDATE CASCADE,

  CONSTRAINT `fk\_Agreement\_Invoice1`

    FOREIGN KEY (`Invoice\_InvoiceID`)

    REFERENCES `NJIT\_cs631103\_carproject`.`Invoice` (`InvoiceID`)

    ON DELETE RESTRICT

    ON UPDATE RESTRICT,

  CONSTRAINT `fk\_Agreement\_Car1`

    FOREIGN KEY (`Car\_VIM`)

    REFERENCES `NJIT\_cs631103\_carproject`.`Car` (`VIM`)

    ON DELETE RESTRICT

    ON UPDATE CASCADE)

ENGINE = InnoDB;

## SQL Execution Proof:

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application

Description automatically generatedThe images below shows the successful execution of the SQL statements.

# **Annex B**

**From Phase 2: PART 5 – Data Insertion.**

## Requirements:

Insert 5 rows of data in each table

## SQL Statements:

show databases;

use NJIT\_cs631103\_carproject;

show tables;

INSERT INTO ClassCar (Name, RateWeekly, RateDaily)

VALUES

("Van",100.00,20.00),

("Minivan",90.00,20.00),

("Pickup",700.00,100.00),

("Sedan",80.00,15.00),

("Sports",5000.00,1000.00);

INSERT INTO CarModel (Make,Name,Year,ClassCar\_Name)

VALUES

("Toyota","Sienna",2023,"Minivan"),

("Dodge","Sprinter",2009,"Van"),

("Tesla","Model 3",2022,"Sedan"),

("GMC","Hummer ",2023,"Pickup"),

("Cadillac","CTS-V",2016,"Sports");

INSERT INTO Car(VIM, Color, ClassCar\_Name, CarModel\_carmodelID)

VALUES

("xyz0987abc1776","red","Sports", 5),

("abc1234xyz1492","purple","Pickup", 4),

("njitm02d09y1881","black","Sedan", 3),

("newark1666nj2022","yellow","Van", 2),

("idont000konw111vim","Green","Minivan", 1);

INSERT INTO Location(AStreet, ACity, AState, AZip, Phone, Car\_VIM)

VALUE

("719 E 11th Ave", "Anchorage", "AK","99501", "123-456-7890", "idont000konw111vim"),

("8000 Gtwy Blvd E", "El Paso", "TX", "79907", "987-654-3210","abc1234xyz1492"),

("180 Washington St", "Newark", "NJ", "07102", "159-357-8520","newark1666nj2022"),

("500 Truman Ave", "Key West", "FL", "33040", "1-800-456-4562","njitm02d09y1881"),

("650 Airport Dr", "Presque Isle", "ME", "04769", "1-800-7416", "xyz0987abc1776");

INSERT INTO Reservation(Status, StartDate, StartTime, EndDate, EndTime, Type, Pickup\_LocationID, Return\_LocationID, ClassCar\_Name)

VALUES

("open", "2022-04-03", "06:20:15", "2022-04-13", "22:22:22", "walkin", 4, 2, "Minivan"),

("open", "2022-05-03", "06:20:15", "2022-05-13", "22:22:22", "phone", 3, 3, "Pickup"),

("open", "2022-06-03", "06:20:15", "2022-06-13", "22:22:22", "walkin", 2, 4,"Sedan"),

("open", "2022-06-03", "06:20:15", "2022-07-13", "22:22:22", "phone", 1, 5, "Sports");

INSERT INTO Customer(FName, LName, Phone, Reservation\_ReservationID )

VALUES

("Jairo","Perez","1234567890", 11),

("Sebastian","Correa","9876543210", 12),

("Ana","Palacio","1597534862", 13),

("Camila","Salderriaga","9638527418", 14),

("Richy","Perez","1472583692", 15);

INSERT INTO Invoice (Status, Date,Taxes,Total)

VALUE

("open", "2022-03-03", 0.25, 0.00),

("open", "2022-04-03", 0.26, 0.00),

("open", "2022-05-03", 0.27, 0.00),

("open", "2022-06-03", 0.28, 0.00),

("open", "2022-06-03", 0.29, 0.00);

INSERT INTO Agreement (StartDate, StartTime, ReturnDate, ReturnTime, StartOdometer, Reservation\_ReservationID, Car\_VIM)

VALUE

("2022-03-03", "06:20:15", "2022-03-13", "22:22:22", 1001, 15,"xyz0987abc1776" ),

("2022-04-03", "06:20:15", "2022-04-13", "22:22:22", 2001, 14,"abc1234xyz1492"),

("2022-05-03", "06:20:15", "2022-05-13", "22:22:22", 3001, 13,"njitm02d09y1881"),

("2022-06-03", "06:20:15", "2022-06-13", "22:22:22", 4001, 12,"newark1666nj2022"),

("2022-06-03", "06:20:15", "2022-07-13", "22:22:22", 5001, 11,"idont000konw111vim");

## SQL Execution Proof:

Write SQL statements with the “**SELECT \***” to proof data is stored in each corresponding table.

SELECT \* FROM Customer;

Table

Description automatically generated

SELECT \* FROM Reservation;

Graphical user interface, application

Description automatically generated

SELECT \* FROM Location;

Graphical user interface, application, table

Description automatically generated

SELECT \* FROM Car;

Graphical user interface, application

Description automatically generated

SELECT \* FROM CarModel;

Table

Description automatically generated

SELECT \* FROM ClassCar;

Table

Description automatically generated

SELECT \* FROM Agreement;

Table

Description automatically generated

SELECT \* FROM Invoice;

Table

Description automatically generated

# **Annex C**

**From Phase 2: PART 6 – Data Update & Deletion**

## Requirements:

1. Update 1 column in each table.
2. Delete 1 row in each table.

SQL Statements:

UPDATE Customer SET FName=" Ana Maria" WHERE customerID=3;

UPDATE Reservation SET EndDate="2022-05-23" WHERE ReservationID=13;

UPDATE Location SET AStreet="38 Access Hwy", ACity="Caribou", AZip=0473 WHERE LocationID=5;

UPDATE Car SET Color="pink" WHERE VIM="xyz0987abc1776";

UPDATE CarModel SET Year=2020 WHERE carmodelID=3;

UPDATE ClassCar SET RateWeekly=5010, RateDaily=1010 WHERE Name="Sports";

UPDATE Agreement SET ReturnDate="2022-05-23" WHERE ContractNumber=9;

UPDATE Invoice SET Taxes=0.35 WHERE InvoiceID=3;

DELETE FROM Customer WHERE customerID=5;

DELETE FROM Reservation WHERE ReservationID=11;

DELETE FROM Location WHERE LocationID=5;

DELETE FROM Car WHERE VIM="xyz0987abc1776";

DELETE FROM CarModel WHERE CarmodelID=5;

DELETE FROM ClassCar WHERE Name="Sports";

DELETE FROM Invoice WHERE InvoiceID=5;

# **Annex D**

**From Phase 2: PART 7 – SQL Queries**

## SQL Queries:

1. Group By

|  |  |
| --- | --- |
| Get a count of the classes of cars being rented out. | SELECT COUNT(ReservationID) AS Count , ClassCar\_Name  FROM Reservation  GROUP BY ClassCar\_Name; |

Graphical user interface, application

Description automatically generated

1. Group By & Having

|  |  |
| --- | --- |
| Get all reservations that have only 1 class of car reserved. | SELECT COUNT(ReservationID) AS Count , ClassCar\_Name  FROM Reservation  GROUP BY ClassCar\_Name  HAVING COUNT(ReservationID) < 2; |

Graphical user interface, text, application

Description automatically generated

1. In

|  |  |
| --- | --- |
| Get all the customer who have a current reservation with a luxury car. | SELECT \*  FROM Customer  WHERE Reservation\_ReservationID IN (  SELECT ReservationID  FROM Reservation  WHERE ClassCar\_Name="Luxury"); |

Table

Description automatically generated with medium confidence

1. All

|  |  |
| --- | --- |
| Get all the customer who have a current reservation with a luxury car. | SELECT CustomerID, FName, LName, Phone  FROM Customer  WHERE Reservation\_ReservationID = All (  SELECT ReservationID  FROM Reservation  GROUP BY ClassCar\_Name  HAVING ClassCar\_Name = "Sedan"); |

Graphical user interface, text, application

Description automatically generated

# **PART 8 – Version Control**

# **PART 9 – Version Control**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Amendment Number | Date | Requestor | Comment | Results |
| 1 | 1.1 | 11/2/2022 | Prof. Canan | MODEL is a separate entity, not just an attribute. CAR has MODEL, MODEL is associated with CLASS. Please revise your diagram. | ER diagram has been modified to reflect changes. Entity MODEL has been added.  Changes reflected on:  Part 2, page 5. |
| 2 | 2.1 | 11/16/2022 | Jairo A. Perez | Current “Relational Schema” diagram is difficult to read. Arrows are difficult to follow. | Switched from Microsoft Word to MySQL Workbench for relational schema diagram creation. |
| 3 | 3.1 | 12/2/2022 | Jairo A. Perez | Based on the 3rd phase requirements, re-organization of some section will be required.  However, the new restructuring will not omit any of the previous “Parts” displayed in version/phase 2 of the project. | Referencing version “Phase 2” of the project, the following parts will become annexes:  Part 4 becomes Annex A  Part 5 becomes Annex B  Part 6 becomes Annex C  Part 7 becomes Annex D |