بسم الله الرحمن الرحيم


Palestine Polytechnic University

College of Information Technology and Computer Engineering

# Smart Parking

Team:

Omar Takrouri

Yusra lahaleeh

Wagdi kaki

Supervisor:

Eng. Ezdehar Jawabreh

December-2018

# الإهداء:

بسم الله الرحمن الرحيم

"سبحانك اللهم لا علم لنا إلا ما علمتنا إنك أنت السميع العليم"

و الصلاة والسلام على أشرف الخلق والمرسلين سيد البشر أجمعين سيدنا وحبيبنا النبي الأمي محمد بن عبد الله صلوات الله وسلامه عليه وعلى صحابته الميامين

، إلى مسرى محمد صلى الله عليه وسلم ومهد المسيح عليه السلام

إلى أرض الشهداء .. والجرحى .. والأسرى

إلى وطننا **فلسطين**

إلى القدس يا منارة الشرائع .. حزينة عيناكِ يا مدينة البتول .. يا واحة ظليلة مرّ بها الرسول .. ستبقين عاصمة فلسطين الأبدية و التاريخية

إلى من علمننا وعانين الصعاب لنصل إلى ما نحن فيه .. إلى منبع الحب و الحنان..

إلى **أمهاتنا** الحبيبات

إلى من يدفعوننا قدماً نحو الأمام لنيل المبتغى .. إلى من يمتلك الإنسانية بكل قوة .. إلى من يسهر على تعليمنا بتضحيات جسام .. إلى مدرستنا الأولى في الحياة ..

إلى **أبائنا** الأفاضل

إلى من كانت شمعة تنير دربنا .. وتأخذ بأيدينا لتقودنا إلى بر الأمان .. متجاوزتاً بنا أمواج الفشل والقصور

إلى **مشرفتنا الفاضلة م. إزدهار جوابرة**

إلى **زملائنا وزميلاتنا** ، رفقاء درب مسيرتنا التعليمية

إليكم جميعاً .. نهديكم نجاحنا هذا راجيين من المولى عز وجل القبول و النجاح

# الملخص:

يركز هذا المشروع على إنشاء طريقة جديدة وسهلة للمستخدمين لركن سياراتهم بطريقة أكثر أمان و فعالية . فمع نمو مدينة مثل الخليل بسرعة ، أصبح تحديد أماكن لوقوف السيارات أكثر صعوبة خاصة في أماكن وقوف السيارات داخل المباني. فمن شأن النظام الجديد والمبتكر أن يفيد المستهلكين بشكل كبير عن طريق توفير الوقت والطاقة،وذلك من خلال جعل المستخدم يقوم بحجز مكان له في المصف قبل الوصول اليه ،حتى يتفادى مشكلة عدم وجود مكان فارغ بعد وصوله،وأيضا إذا لم يقم بالحجز يستطع عند تواجده على مدخل المصف و من خلال الشاشة الرئيسية الموجودة على المدخل معرفة أين يوجد مكان فارغ وفي أي طابق والذهاب له مباشرة بدلاً من القيادة عبر مساحة وقوف السيارات بأكملها ، خاصةً إذا كانت المنطقة تتكون من عدة طوابق.

يفيد هذا المشروع المالك أيضًا بشكل كبير ، من خلال منحهم القدرة على عرض الكاميرات الأمنية ، وكذلك عرض التقارير اليومية والأسبوعية والشهرية والسنوية عن عدد السيارات التي دخلت موقف انتظار السيارات والحصول على إيرادات إجمالية،هذا من شأنه أن يلغي الحاجة لموظفي الأمن متعددة ، ومحاسب.

مع وجود مشكلة ستزداد بمرور الوقت ، يحل مشروعنا هذه المشكلة عن طريق استخدام تقنية جديدة ودمجها في مواقف السيارات لإنشاء نظام إنترنت للأشياء ، مما يجعل مواقف السيارات أكثر تقدمًا ، وهناك عن طريق جعل المدينة أكثر تقدما من الناحية التكنولوجية.

# Abstract:

This project focuses on creating a new and easy way for users to park their cars in a more time efficient way. With a city like Hebron rapidly growing, parking spaces become more and more difficult to locate inside buildings parking spaces. A new and innovative system would benefit the consumers greatly by saving them time and energy. If a user was able to locate a parking spot as soon as they enter the parking lot, instead of driving through the entire parking space, especially if the area consists of multiple floors.

This project also benefits the owner greatly, by giving them the ability to view the security cameras, and also viewing the daily, weekly, monthly and yearly reports on how many cars entered the parking lot, and getting a total revenue. This would eliminate the need for multiple security employees, and an accountant.

With a problem that is just going to get bigger as time goes by, our project solves this problem by using new technology and integrate it with the parking lots to create an internet of things system, that would make the parking lots more advanced, and there by make the city more technologically advanced.

# Table of Contents

# List of Figures



# List of table

# Abbreviations

**IOT:** Internet OF Things

**IR:** Infrared sensor

**LED:** Light Emitting Diode

**PC:** Personal Computer

**WAMP:** Windows Apache MySQL PHP

**PHP:** Hyper Text Preprocessor

**WI-FI:** Wireless

**SD:** Secure Digital

**CPU:** Central Processing Unit

**USB:** Universal Serial Bus

**LCD:** Liquid Crystal Display

**QR:** Quick Response Code

# Chapter 1

## Introduction:

### Overview of the project:

smart parking is a way to help consumers to park their cars in the most efficient way possible, especially in growing cities like Hebron. Adopting (IOT) concept, we strive to create a smart way to ease the parking experience for the average user.

We also wish to create an easy and simple way for the owner of the parking IOT to manage this new parking system through a very easy to use administrator website.

### Motivation and importance:

With a city like ours rapidly growing, parking spaces become more and more difficult to locate inside buildings parking spaces. A new innovative system would benefit the consumers greatly by saving them time and energy. If a user was able to locate a parking spot as soon as they enter the parking lot, instead of driving through the entire parking space, especially if the area consists of multiple stories.

This project also benefits the owner, by giving them the ability to see the security cameras, also viewing the daily, weekly, monthly and yearly reports on how many cars entered the parking lot, and getting a total revenue. This would eliminate the need for multiple security employees, and an accountant.

### Objective:

The main objective is to create an advanced way to park our cars, to keep up with this evolving environment we live in. by using a combination of sensors, controllers, large displays and a website, our goal would be achieved.

### Description:

In this project, we plan on using both hardware and software to achieve IOT.

The hardware part is going to consist of mainly a raspberry pi 3 to control the hardware parts.

The software part is going to be a two-part website, the first website is a smartphone compatible page that the average consumer can use to see how many parking spaces are available before getting to the physical location, and they also have the ability to reserve a parking space for a certain amount of time.

The other website is going to be for the owner of the parking lot, where they manage the parking lot. Using this project would lower employees cost, where there wouldn't be as many security employees, and there wouldn't be a need an employee at the main entrance, it would also benefit the owner by getting the exact amount of money made each day.

This project is most affective in large parking spaces and in multi-leveled parking spaces as well.

## Problem analysis:

The main purpose of this project is to solve a rapidly increasing problem, where it's becoming hard to locate a parking spot in certain places, like shopping malls, or office buildings. This project would decrease the human efforts in finding an empty space, and use our technology to locate and park their car, in a time and effort efficient way.

## List of requirements:

This project requires both hardware and software components to achieve:

### *hardware equipment:*

1. IR sensor or ultrasonic sensor
2. LED light
3. Raspberry pi 3
4. monitor
5. entrance gate
6. cameras
7. PC

### *Software components:*

1. Visual studio
2. Sublime editor
3. Wamp
4. Browser

## Programming Language:

1. PHP for programming the website
2. C++ for programming the raspberry pi

System constrains:

One of the most important constraints is the internet connection and the electricity, if either one gets cut off, the entire system would shut down, also if the QR code was misread, the gate would not open and wouldn't let any customers in.

# Chapter 2:

## Background:

1. **Raspberry pi 3:**

**Description:** raspberry pi 3 contains a 1.2 GHz quad-core processor with a 1 GB RAM, along with a built in WI-FI and Bluetooth 4.1. it also contains all the necessary I/O ports and a SD card slot for memory.

**Alternative:** Arduino.

**Reason for using this product:** with a faster CPU, raspberry pi 3 can handle our project without any hiccup, and with the wireless chip sets built in, it would make it much easier to connect the hardware and the software together, without any extra modules.

2. **Ultra sonic sensors:**

**Description:** As the name indicates, ultrasonic sensors measure distance by using ultrasonic waves. The sensor head emits an ultrasonic wave and receives the wave reflected from the target. The sensor measures the distance to the target by measuring the time between the emission and reception of the signal.

**Alternative:** IR sensor

**Reasons for using it:** can handle a wider range of distance and doesn't get affected by sun light.

3. **RGB LED:**

**Description:** It's an RGB (Red, Green, Blue) LED that contains the three primary colors.
**Alternative:** using a single color LED.
**Reasons for using it:** instead of using 2 different LED (Green, Red), we can use a single LED that contains those 2 colors.

4. **Raspberry pi camera module:**


**Description:** It's a module that can connect to the raspberry pi 3 without an adapters or extra components, it contains a 5MPixel camera sensor.
**Alternative:** using a USB camera.
**Resones for using it:** there is a dedicated slot on the raspberry pi for this camera module, so I wouldn't need to occupy a USB port for the camera.

5. **9g micro servo:**

**Description:** A small 9 grams' motor to control the gate at the entrance.

6. **Jumper wires:**

**Description:** used to connect all the components to the Raspberry pi 3 pins.

7. **LCD display:**

**Description:** A small programmable LCD screen used to show the number of available spaces inside the parking lot.

8. **QR scanner:**

**Description:** A scanning device that uses laser to read both barcode and QR code.

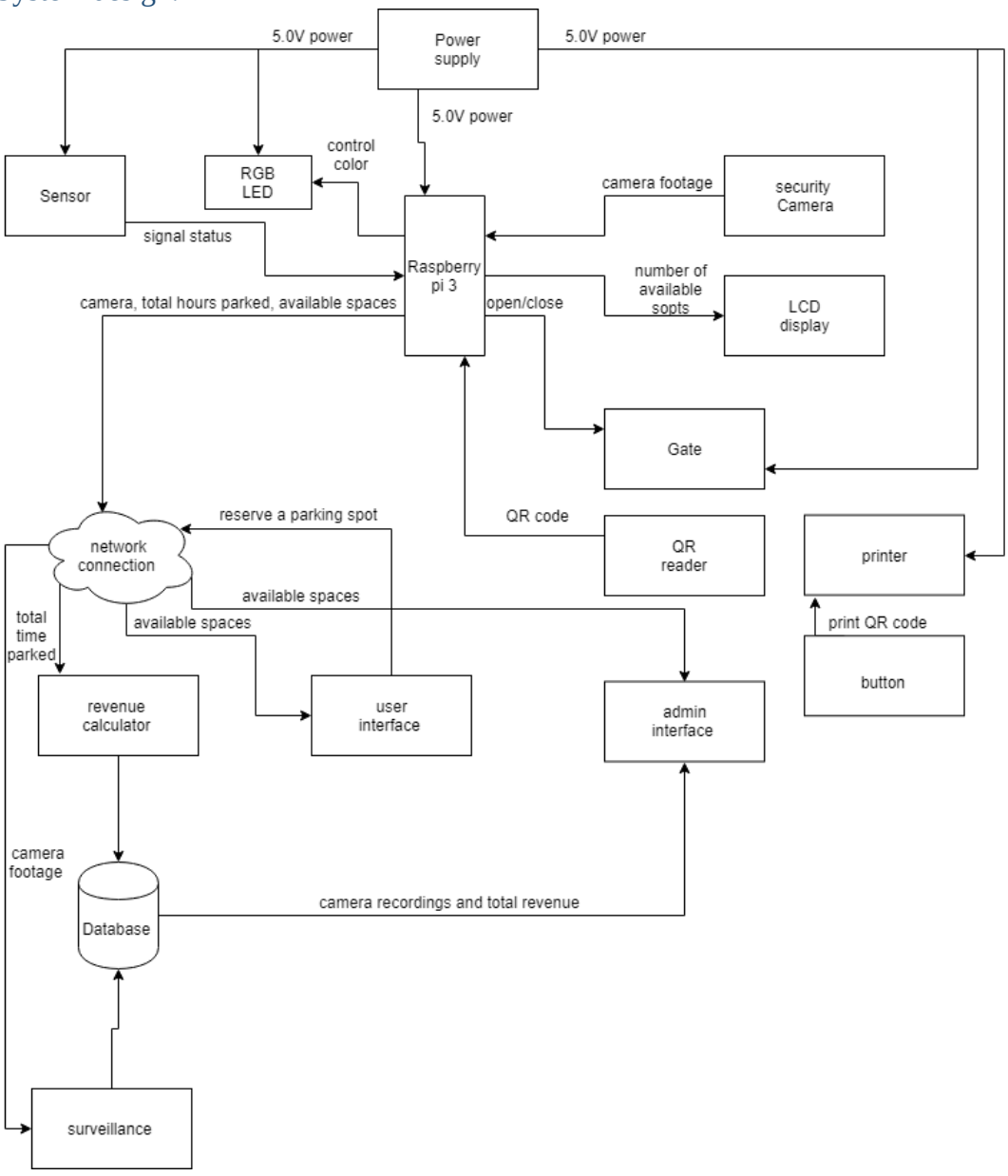
**Alternative:** USB camera.

**Reasons for using it:** more accurate for close and further distances.

# Chapter 3

## Software and hardware design

### System design:




*Fieger 1: system block diagram*

## Description:

The Raspberry pi is used as the main controller, it has multiple input and output components, the main input component is the ultrasonic sensor, which sends its signal to the raspberry pi to process. The raspberry pi controls the RGB LED depending on the signal it got from the ultrasonic sensor. The ultrasonic sensor also has an effect on the screen attached to the raspberry pi. After processing the signal from the raspberry pi, and the QR code sent from the QR reader, the number of available spaces shows up on the screen that is placed on the entrance of the parking lot.

A gate is placed at the front of the entrance, and is controlled by the raspberry pi, depending on whether the owner has reservation, or if there is an available space inside the parking lot. If the user didn't have a reservation, they can press a button on the entrance and get a QR code printed out to them.

The last hardware input to the raspberry pi is the security camera witch sends the footage to the pi to be processed and sent to storage.

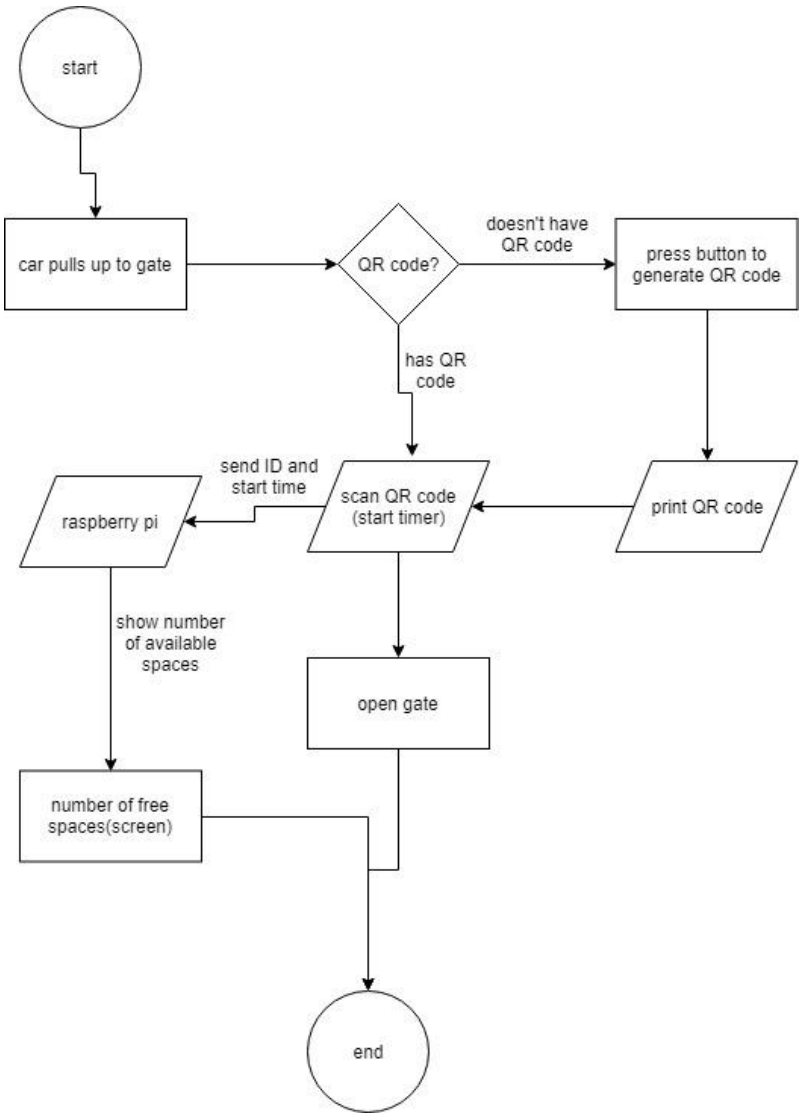
The raspberry pi is connected to the internet, and it has different types of data being sent back and forth between the website and the raspberry pi, such as the camera footage and the total number of hours parked and the number of available spaces are all sent over the internet to the server to be processed and stored.

The total number of hours parked are processed through a function to produce the total revenue, which then is stored in a database. The camera footage goes through a function to either show live footage on the administrator's page or to store the footage in the database.

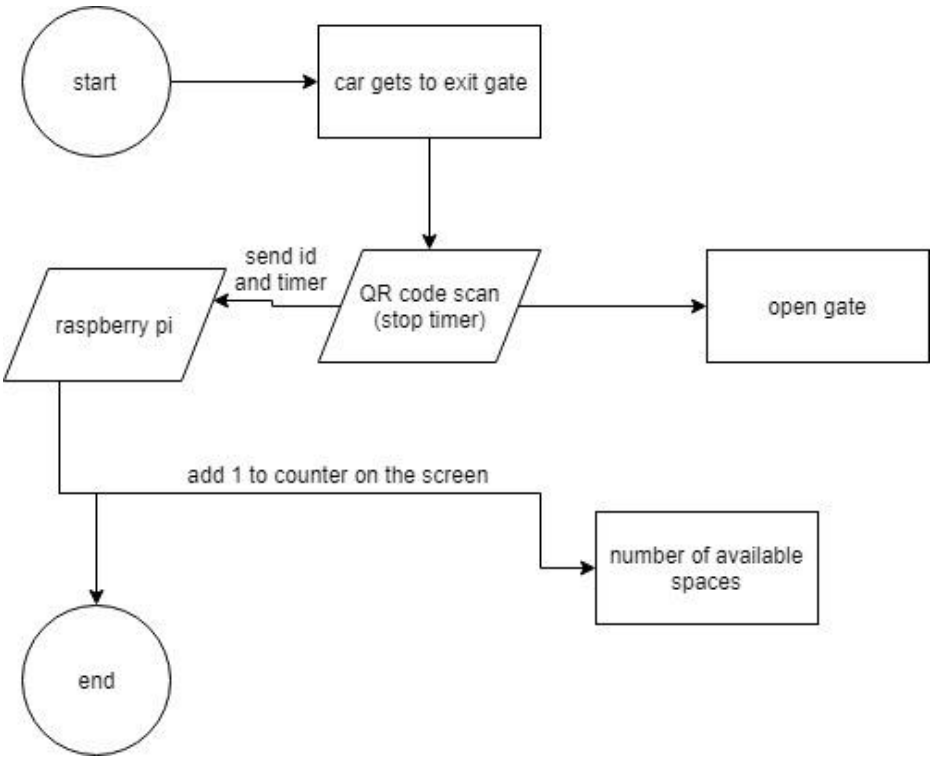
The webpage for the user shows the number of available spaces, and also gives the option to reserve a parking spot, which produces a QR code to be used at the entrance of the parking spot.

The administrator's webpage also shows the number of available spaces, and it also has the ability to access the database to view the revenue of the parking lot, and to view stored footage of the security cameras, it also has the ability to view live footage from the parking lot.

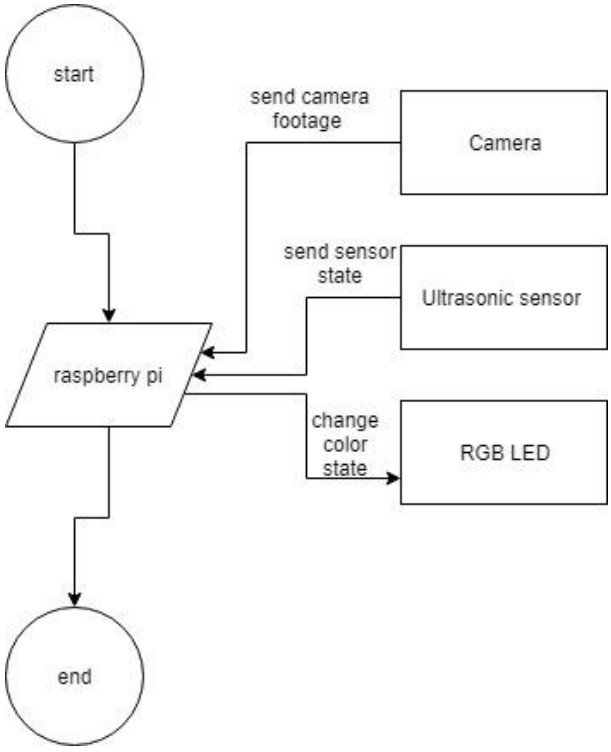
## Hardware flow chart:



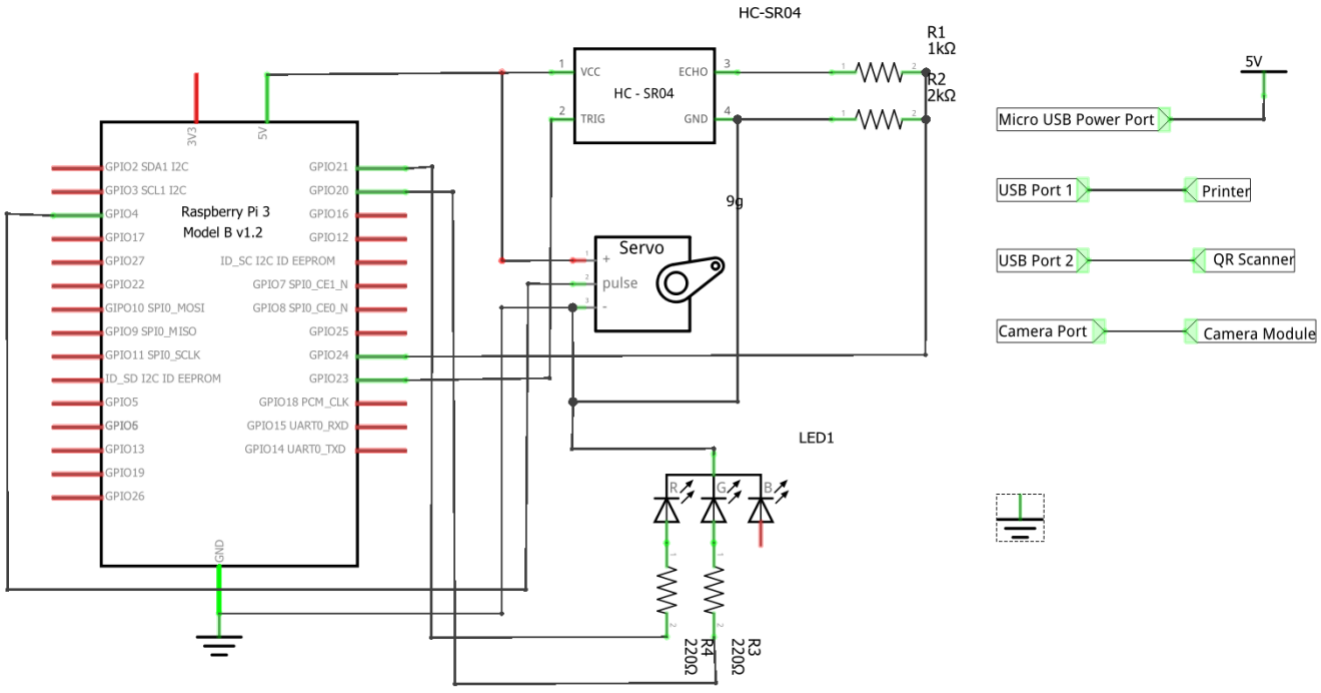

*Fieger 2: car entering flow chart.*




*Fieger 3: car exiting*




*Fieger 4: parking lot*




*Fieger 5: schematic diagram*

## Software design:

Actor: this system has two actor

1. Admin
2. User

### Functional Requirements for Admin:

1. show the number of free spaces:
   shows the number of available spaces in the parking lot.

2. Show Surveillance cameras:
   The admin can watch the camera recordings or live stream them.

3. Shows the financial returns and reports:
   The administrator can view the financial returns for a day, a month, or a year.

4. Log in to the system:
   The administrator need to log in to access the system

5. Log out from the system:
   When the administrator is finished with their work on the system, they have the ability to log out from their account.

### Functional Requirements User:

1. Log in:

   The user need to have an account to be able to reserve a parking space

2. Log out:

   The user has the ability to log out from their account.

   ‘

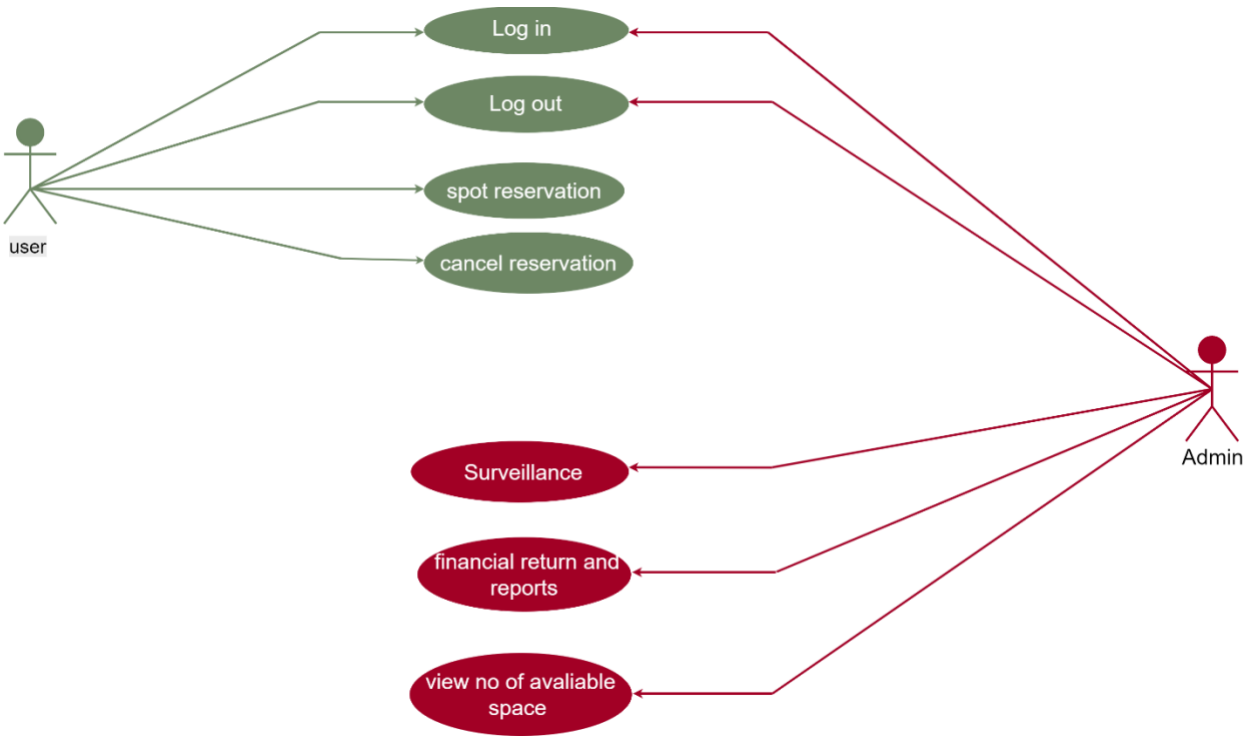
3. See number of position:

   Any user, whether they are logged on or off, can view the number of available spaces.

4. Reserve position:

   The logged on user has the ability to reserve a parking spot for up to 30 minutes.

5. Cancel reservation:
   The user can cancel the reservation before the 30 minutes are up.




*Fieger 6: Use-case diagram:*

## Use-cases descriptions:

### *Admin Use case:*

| Use Case | Login |
|---|---|
| Primary Actor | Admin |
| Goal in Context | Allow the admin to access the system |
| Precondition | The administrator must have a valid username and password to login |
| Trigger | The admin requires access to the system |
| Scenario | 1-The admin goes to the login page<br>2-The admin enter their username and password<br>3-The site validates the username and password before granting them access to the system |
| Exception | The admin enters an invalid username or password |

*Table 1: Admin – Login*

| Use Case | Log out |
|---|---|
| Primary Actor | Admin |
| Goal in Context | Disconnect from the system |
| Precondition | The admin is already logged on the system |
| Trigger | The admin requests to logout |
| Scenario | 1-The admin select the “log out” button from the website |
| Exception | The admin is not logged on. |

*Table 2: Admin – Log out*

| Use Case | Surveillance cameras |
|---|---|
| Primary Actor | Admin |
| Goal in Context | view the records of the cameras |
| Precondition | The admin has to be logged on |
| Trigger | The admin requires access to the security system |
| Scenario | After logging in, the admin select camera from the navigation bar.<br>The admin gets transferred to the surveillance page to either view a live feed or view the recordings. |
| Exception | The admin isn’t logged on. |

*Table 3: Admin – Surveillance cameras*

| | |
|---|---|
| Use Case | show the number of free spaces |
| Primary Actor | Any user |
| Goal in Context | View free places in the park |
| Precondition | Have internet connection |
| Trigger | Go to the home page |
| Scenario | The user goes to the webpage to view the available spaces. |
| Exception | There is no internet connection to reach the site. |

*Table 4: Admin – show the number of free spaces*

| | |
|---|---|
| Use Case | Financial return |
| Primary Actor | Admin |
| Goal in Context | View earnings daily, monthly, or yearly |
| Precondition | Login to access the page of the Financial return |
| Trigger | Admin requests the financials page from the menu bar. |
| Scenario | 1-The admin access the Financial return<br>2-View the earnings every day, month, or year |
| Exception | No internet connection. |

*Table 5: Admin – Financial return*

*User Use case:*

| | |
|---|---|
| Use Case | Log in |
| Primary Actor | User |
| Goal in Context | Enable user to access to the system |
| Precondition | The user has a valid username and password and is not already logged in |
| Trigger | The user requires access to the system for reserve position or unreserved position or see the number of position. |
| Scenario | 1-The user select log in from the home page in website.<br>2-the user enter their username and password<br>3-the site enable access to the system according to access control |
| Exception | The user enters invalid username and password |

*Table 6: User – Log in*

| | |
|---|---|
| Use Case | Log out |
| Primary Actor | User |
| Goal in Context | The user needs to log out from the website. |
| Precondition | The user is already logged in to the website . |
| Trigger | The user requests to log out |
| Scenario | The user clicks on “log out” button from the menu bar |
| Exception | The user doesn’t have an account in the website. |

*Table 7: User – Log out*

| Use Case | Reserve position |
|---|---|
| Primary Actor | User |
| Goal in Context | Reserve a spot before getting to the physical location. |
| Precondition | The user has an account in the website and is already logged in |
| Trigger | The user request a parking spot prior to getting to the physical location |
| Scenario | 1-The user log in the website.<br>2-the user check if the park has a position or not<br>3- if the park has position the user reserve it else the website display message to the user “Sorry we don’t have a space” and user log out from the website.<br>4-after the user choose reserved the position it determines the time he needs to arrive until the park.<br>5-after it reserved position the website display image for QR code and the user download from his mobile.<br>6-the use Log out from the website. |
| Exception | The user don’t has internet. |

*Table 8: User – Reserve position*

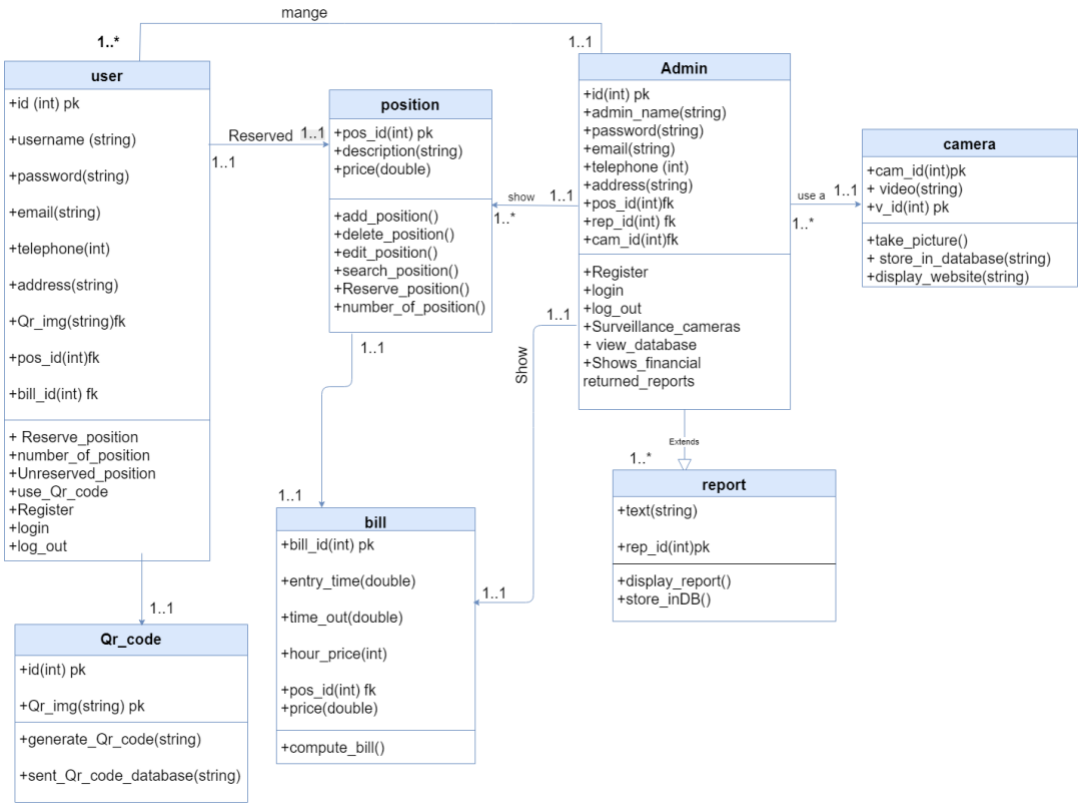
| Use Case | Cancel reservation |
|---|---|
| Primary Actor | User |
| Goal in Context | Cancel reservation before the expiration time is out. |
| Precondition | The user has a reservation |
| Trigger | The user requests to cancel the reservation form the website. |
| Scenario | 1-The user log in the website.<br>2-The user go to the reserved menu.<br>3-the user clicks on the cancel reservation button.<br>4-afer clicking on the cancel but, a message box show asking if the user is sure they want to cancel their reservation.<br>5-the user clicks on “Yes”. |
| Exception | The user doesn’t has a reserved position. |

*Table 9: User – Cancel reservation*

| Use Case | See number of position |
|---|---|
| Primary Actor | User |
| Goal in Context | The users want the check the number of available spaces before going the physical location. |
| Precondition | Have internet connection |
| Trigger | The user goes the main web page |
| Scenario | 1-The user log in the website.<br>2-the user checks if the parking lot has a free space or not |
| Exception | The user doesn’t have an internet connection, |

*Table 10: User – See number of position*

## *System classes:*




*Fieger 7: System classes*

Mapping:

Described for each table:

- User: the table contain this information:
  - id: a unique integer value of a user and it's the primary key of this table (auto-increment).
  - username: a string value for the user .
  - password: a string value to the password for the user, through this value the user can enter on the parking website.
  - email: a string value for the user.
  - telephone: an integer value for the user.
  - address: a string value for the user.
  - Qr_img: a string value (image) for the user, the user use it for to inter in the parking, and it's a foreign key of this table.
  - pos_id: an integer value for the position on the parking, and it's a foreign key of this table.
  - bill_id: an integer value for the bill and it's a foreign key of this table.

- Qr_code: the table contain this information:
  - id: a unique integer value for the Qr_code, and it's the primary key of this table (auto-increment).
  - Qr_img: a string value for (Qr_code) and the website save it on the user mobile, and user use it when he enters the parking.

- position: the table contain this information:
  - pos_id: a unique integer value for the position on the parking, and its primary key of this table (auto-increment).
  - description: a string value for a position and it's tell the user description about his position like (in any floor, and the time of booking end).
  - price: a double value for the bill, it tell the user how much he must to pay, for the booking.

- bill: the table contain this information:
  - bill_id: a unique integer value for the bill and its primary key of this table (auto-increment).
  - entry _time: a double value for time when the car enter in the parking.
  - time_out: a double value for the time when the car out of the parking.
  - hour_price: an integer value for price booking in hour.

- pos_id: an integer value for the position on the parking, and it's a foreign key of this table.

- price : a double value for the bill, and its compute the price for booking and it display to the user how much he pay.

- camera: the table contain this information:
  - cam_id: a unique integer value for each camera and its primary key of this table (auto-increment).
  - v_id: a unique integer value for each video the camera takes it, and its primary key of this table (auto-increment).
  - video: a string value for recording video, the cameras on the parking takes it and store it in the database.

- report: the table contain this information:
  - rep_id: a unique integer value for each report and its primary key of this table (auto-increment).
  - text: a string value for the report, and its describe the all information about the financial return, it stores in database, the admin see it any time, and its be for a week, month or year.

- Admin: the table contain this information:
  - id: a unique integer value for each admin and its primary key of this table (auto-increment).
  - admin_name: a string value for the admin.
  - password: a string value for the admin, through this value the admin can enter on the system.
  - email: a string value for the admin.
  - telephone: an integer value for the admin.
  - address: a string value for the admin.
  - pos_id: an integer value for the position on the parking, admin use it to access to position, and it's a foreign key of this table.
  - cam_id: an integer value for each camera, admin use it to display a special video if any problem its happen and its foreign key of this table.
  - rep_id: an integer value for each report, admin use it to allow the reports and its foreign key of this table.

*Database tables:*

| Name | Type | Length | Null | Auto increment | PK | FK | default |
|---|---|---|---|---|---|---|---|
| Id | Int | 10 | No | Yes | Yes | No | - |
| username | String | 40 | No | No | No | No | - |
| password | String | 40 | No | No | No | No | - |
| Email | String | 40 | No | No | No | No | - |
| telephone | Int | 20 | No | No | No | No | - |
| Address | String | 40 | No | No | No | No | - |
| Qr_img | String | 50 | No | No | Yes | No | - |
| pos_id | Int | 10 | No | Yes | No | Yes | - |
| bill_id | Int | 10 | No | Yes | No | Yes | - |

*Table 11: User Table in database*

| Name | Type | Length | Null | Auto increment | Pk | Fk | default |
|---|---|---|---|---|---|---|---|
| Id | Int | 10 | No | Yes | yes | No | - |
| Qr_img | String | 50 | No | No | yes | No | - |

*Table 12: Qr_code table in database*

| Name | Type | Length | Null | Auto increment | Pk | Fk | default |
|---|---|---|---|---|---|---|---|
| bill_id | Int | 10 | No | Yes | Yes | No | - |
| entry_time | Double | 10 | No | | No | No | - |
| time_out | Double | 10 | No | | No | No | - |
| hour_price | Int | 10 | No | No | No | No | - |
| pos_id | Int | 10 | No | Yes | No | Yes | - |
| Price | Double | 20 | No | No | No | No | - |

*Table 13: Bill table in database*

| Name | Type | Length | Null | Auto increment | Pk | Fk | default |
|---|---|---|---|---|---|---|---|
| pos_id | Int | 10 | No | Yes | Yes | No | - |
| description | String | 50 | No | No | No | No | - |
| Price | Double | 20 | No | No | No | No | - |

*Table 14: Position table in database*

| Name | Type | Length | Null | Auto increment | Pk | Fk | Default |
|---|---|---|---|---|---|---|---|
| text | String | 100 | No | No | No | No | - |
| rep_id | Int | 10 | No | Yes | Yes | No | - |

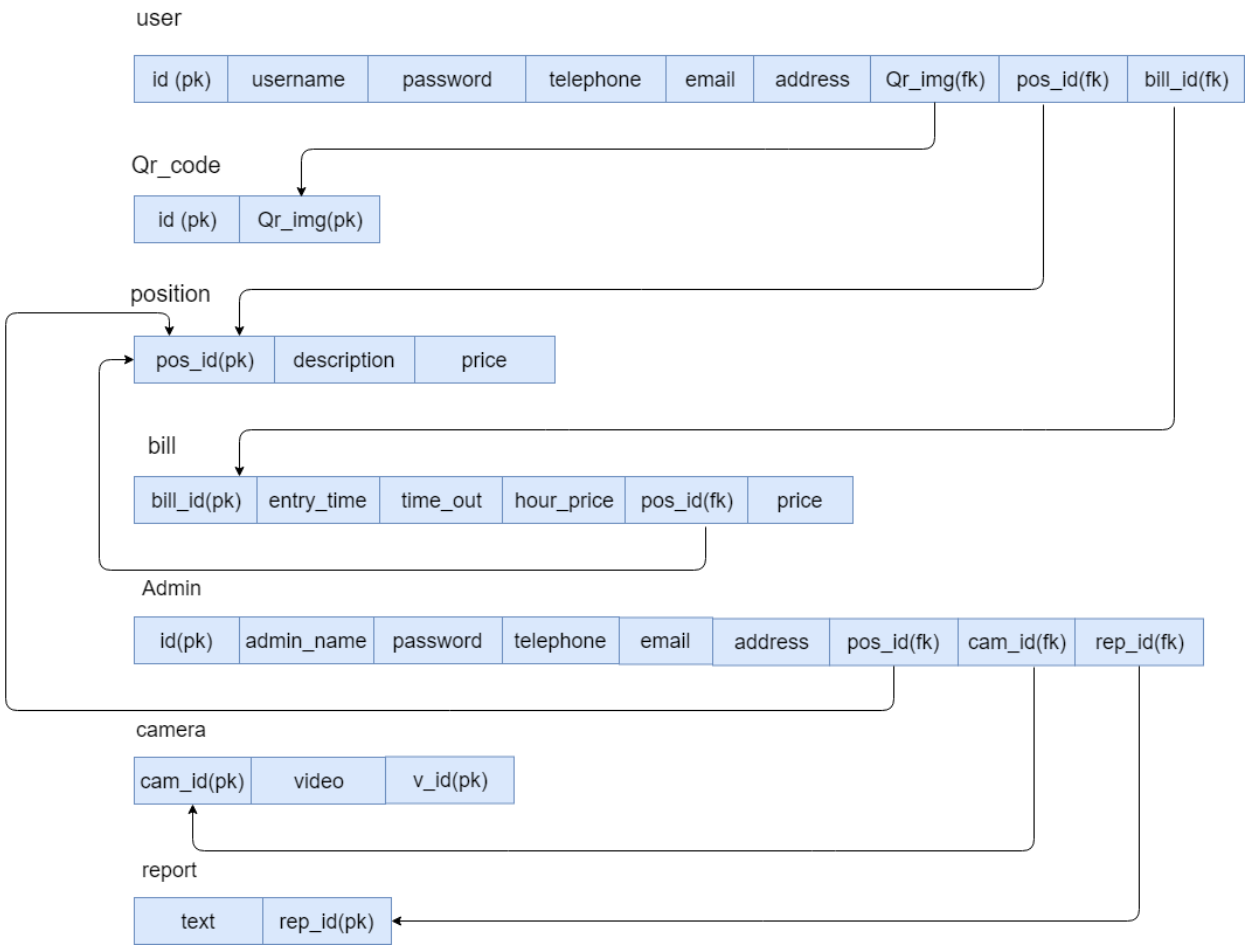
*Table 15: Report table in database*

| Name | Type | Length | Null | Auto increment | Pk | Fk | Default |
|---|---|---|---|---|---|---|---|
| id | Int | 10 | No | Yes | Yes | No | - |
| admin_name | String | 40 | No | No | No | No | - |
| password | String | 40 | No | No | No | No | - |
| email | String | 40 | No | No | No | No | - |
| telephone | Int | 20 | No | No | No | No | - |
| address | String | 40 | No | No | No | No | - |
| pos_id | Int | 10 | No | Yes | No | yes | - |
| rep_id | Int | 10 | No | Yes | No | Yes | - |
| cam_id | Int | 10 | No | Yes | No | Yes | - |

*Table 16: Admin table in database*

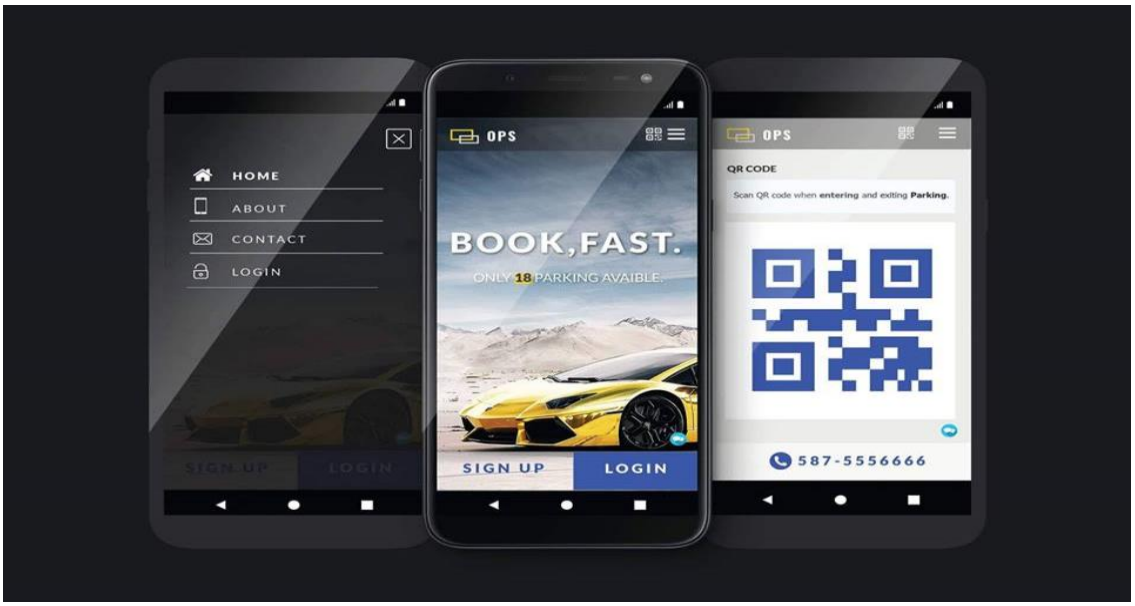
| Name | Type | Length | Null | Auto increment | Pk | Fk | Default |
|---|---|---|---|---|---|---|---|
| cam_id | Int | 10 | No | Yes | Yes | No | - |
| video | String | 100 | No | No | No | No | - |
| v_id | Int | 10 | No | Yes | Yes | No | - |

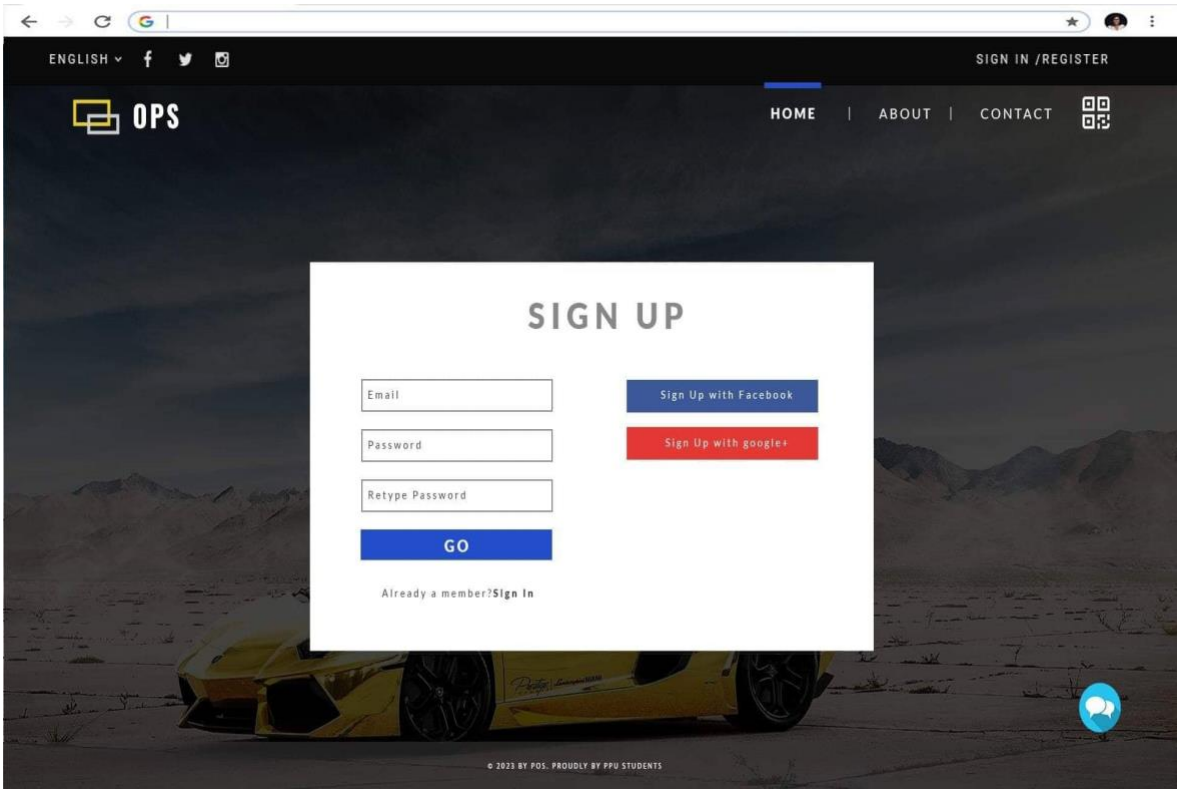
*Table 17: Camera table in database*

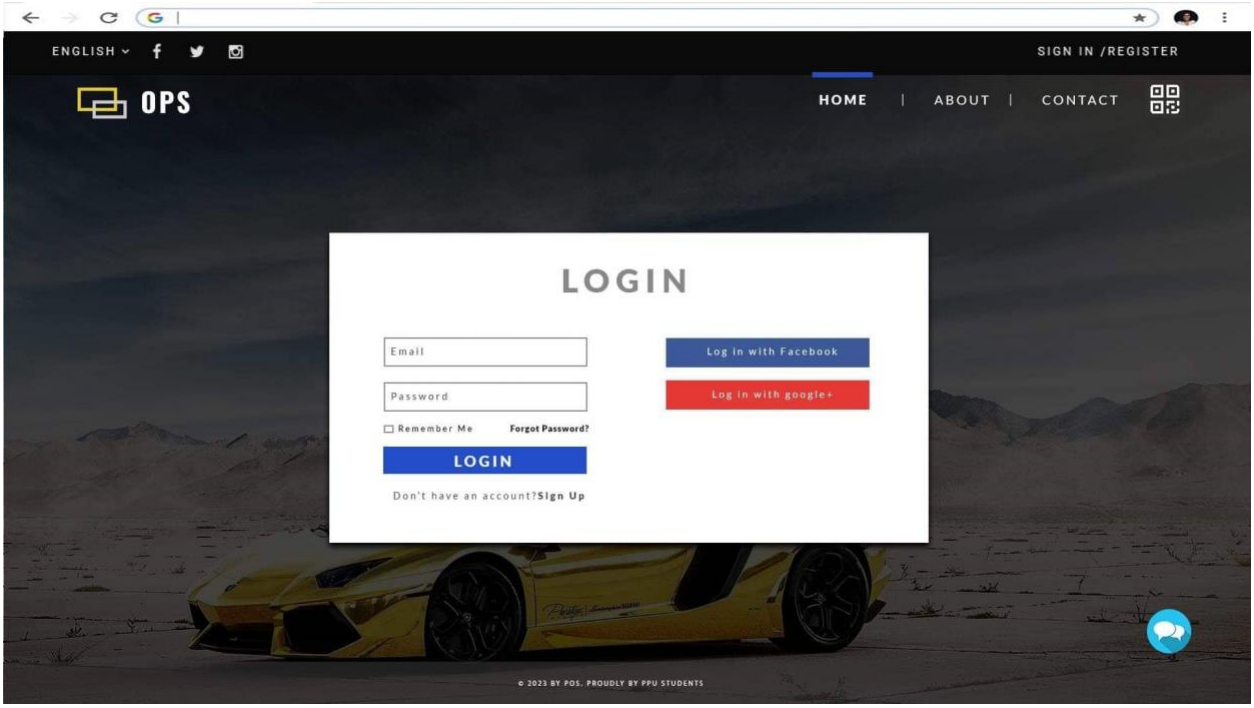


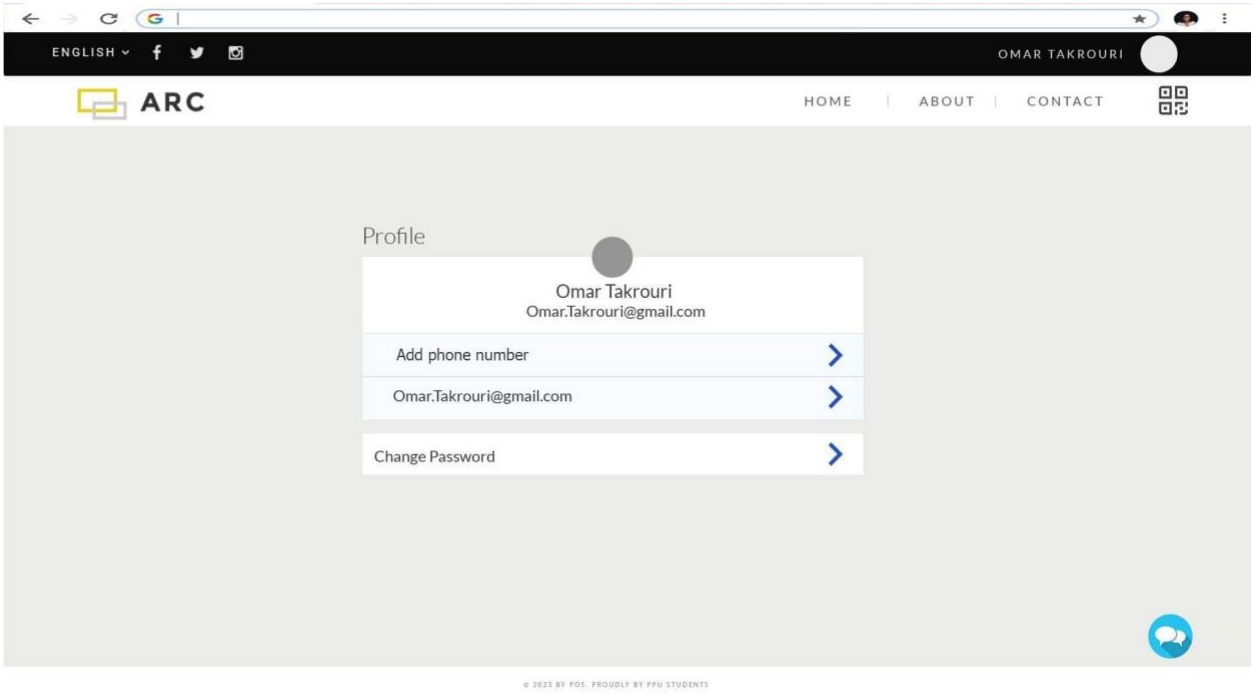

*Fieger 8: Relation between the database table:*

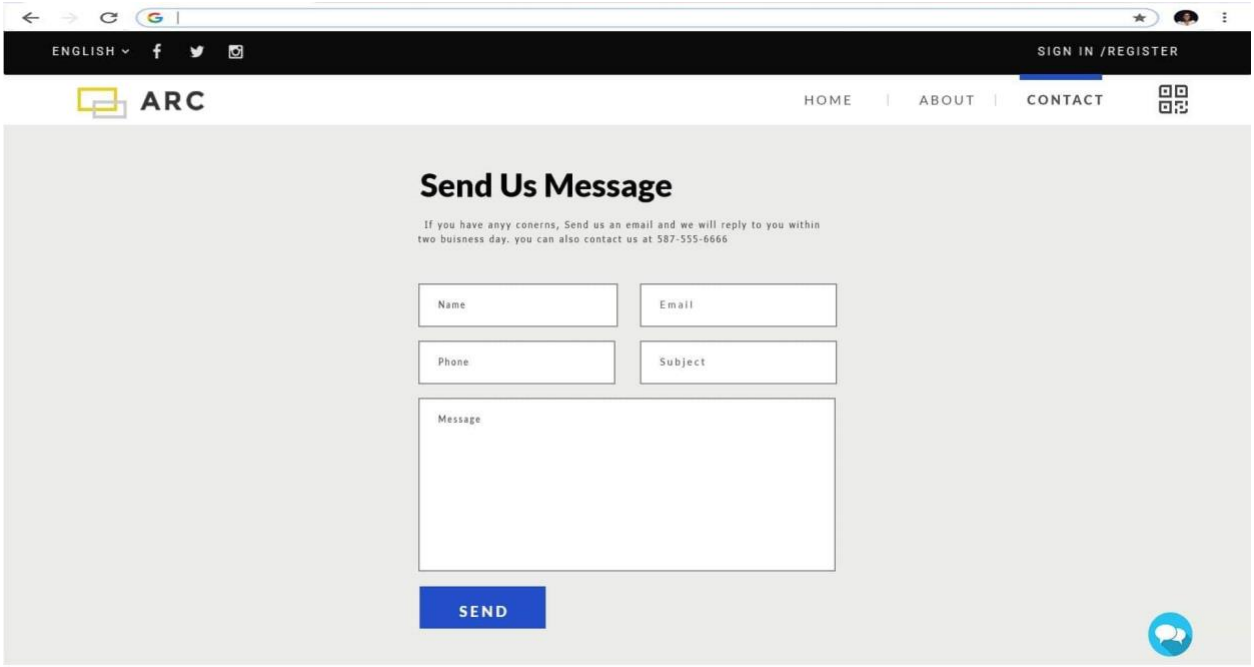
## Layout:

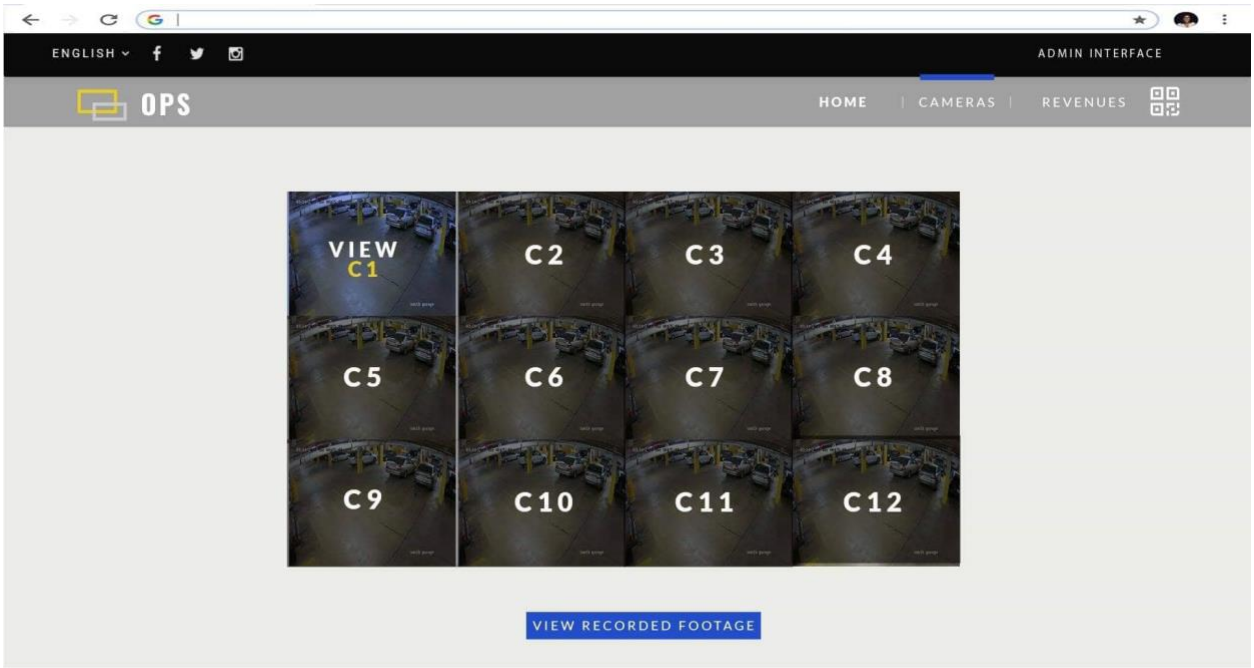

HOME
ABOUT
CONTACT
LOGIN
OPS
BOOK,FAST.
ONLY 18 PARKING AVAIBLE.
SIGN UP
LOGIN
OPS
QR CODE
Scan QR code when entering and exiting Parking.
587-5556666

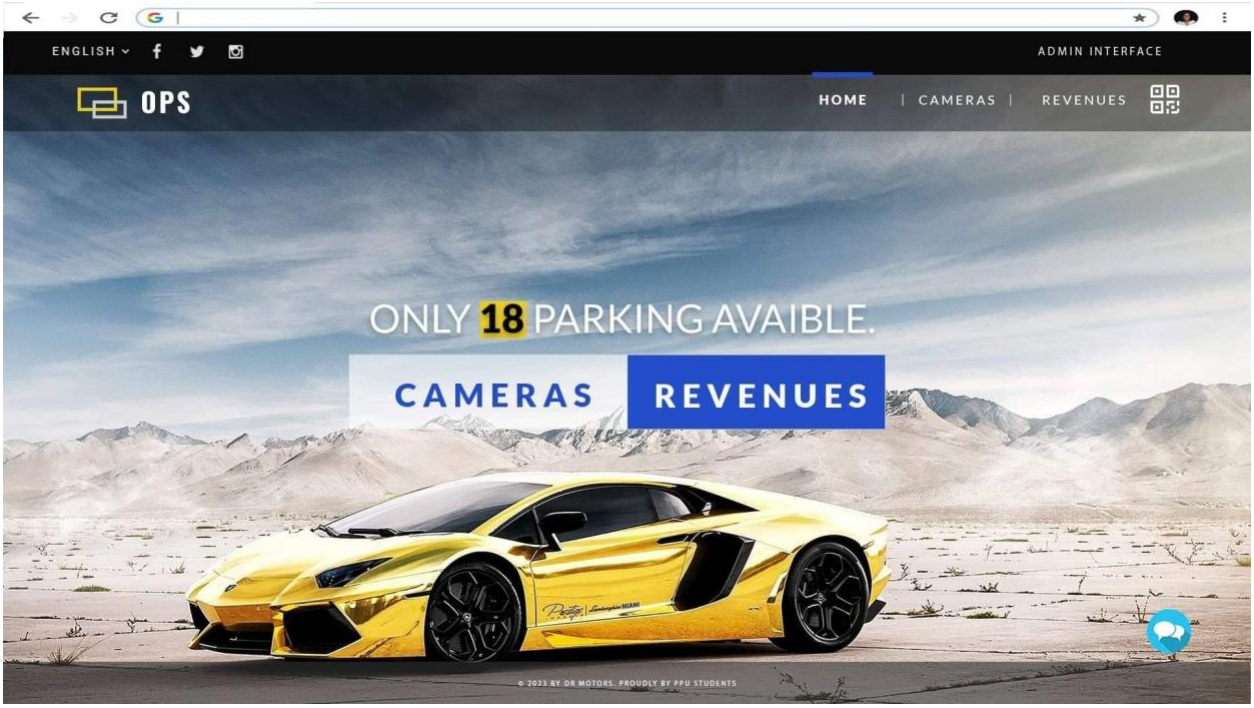

ENGLISH
SIGN IN /REGISTER
OPS
HOME | ABOUT | CONTACT
SIGN UP
Email
Password
Retype Password
GO
Already a member?Sign In
Sign Up with Facebook
Sign Up with google+
© 2023 BY POS. PROUDLY BY PPU STUDENTS

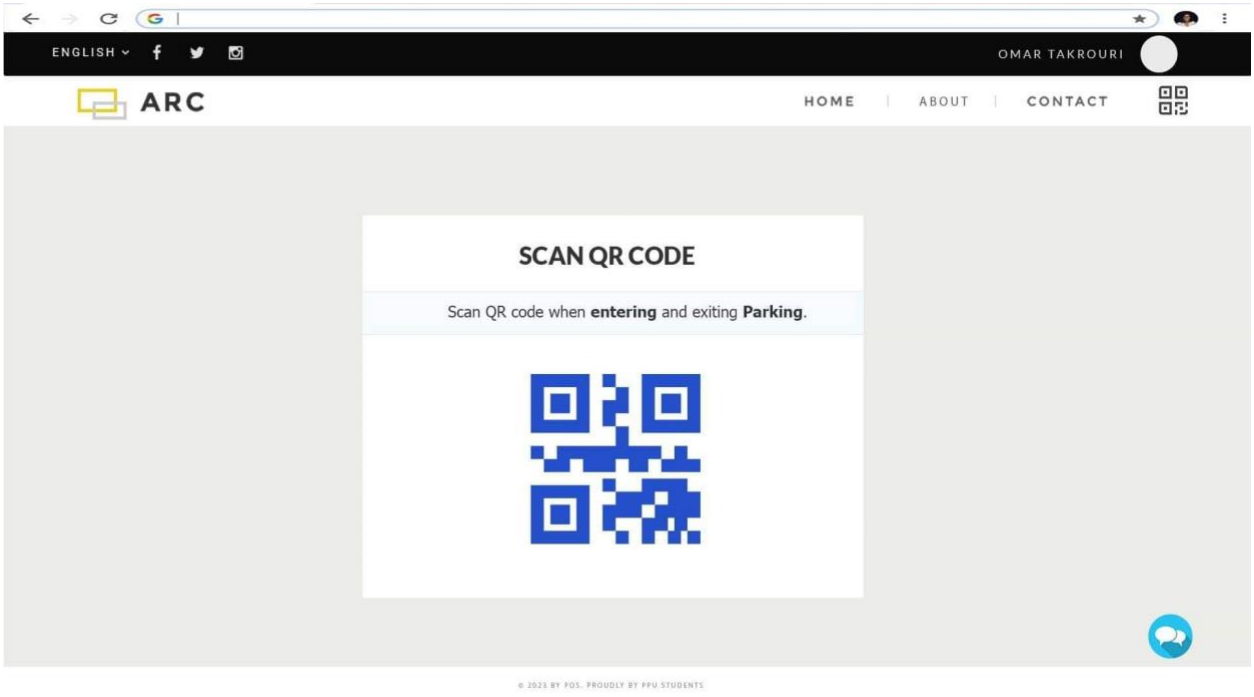

ENGLISH
SIGN IN /REGISTER
OPS
HOME | ABOUT | CONTACT
LOGIN
Email
Password
Remember Me
Forgot Password?
LOGIN
Don't have an account?Sign Up
Log in with Facebook
Log in with google+
© 2023 BY POS. PROUDLY BY PPU STUDENTS

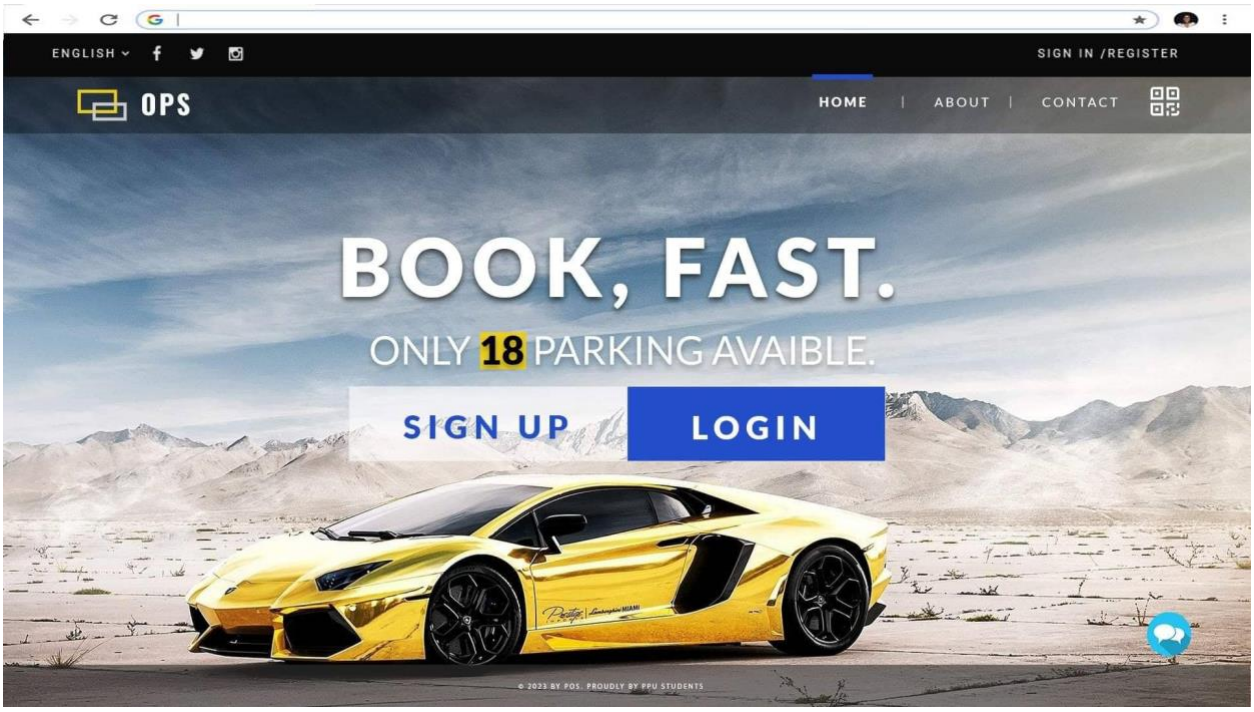

ENGLISH
OMAR TAKROURI
ARC
HOME | ABOUT | CONTACT
Profile
Omar Takrouri
Omar.Takrouri@gmail.com
Add phone number
Omar.Takrouri@gmail.com
Change Password
© 2023 BY POS. PROUDLY BY PPU STUDENTS

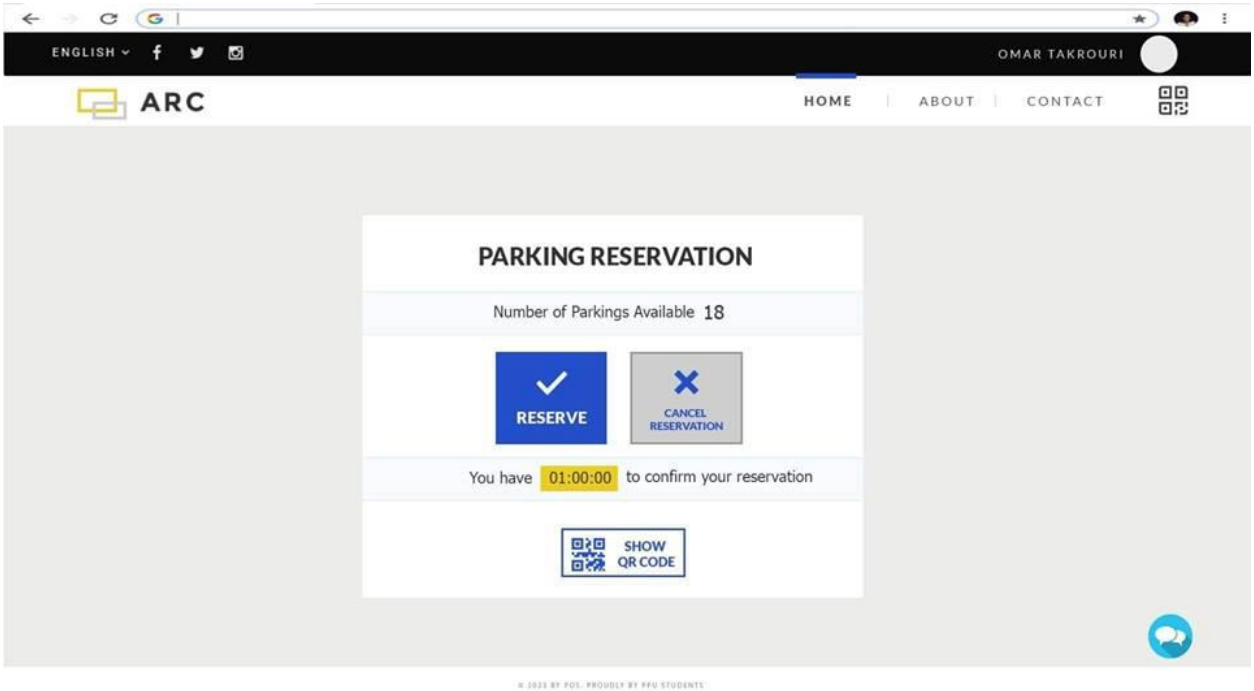

ENGLISH
SIGN IN /REGISTER
ARC
HOME
ABOUT
CONTACT
Send Us Message
If you have anyy conerns, Send us an email and we will reply to you within two buisness day. you can also contact us at 587-555-6666
Name
Email
Phone
Subject
Message
SEND
© 2023 BY POS. PROUDLY BY PPU STUDENTS

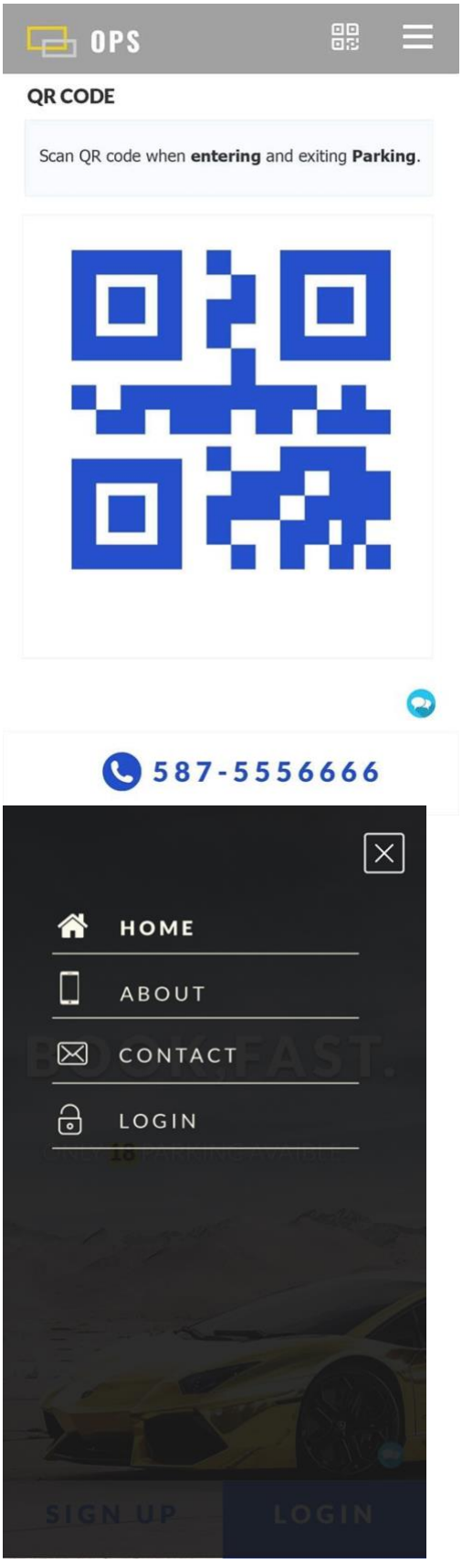

ENGLISH
ADMIN INTERFACE
OPS
HOME
CAMERAS
REVENUES
VIEW C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
VIEW RECORDED FOOTAGE

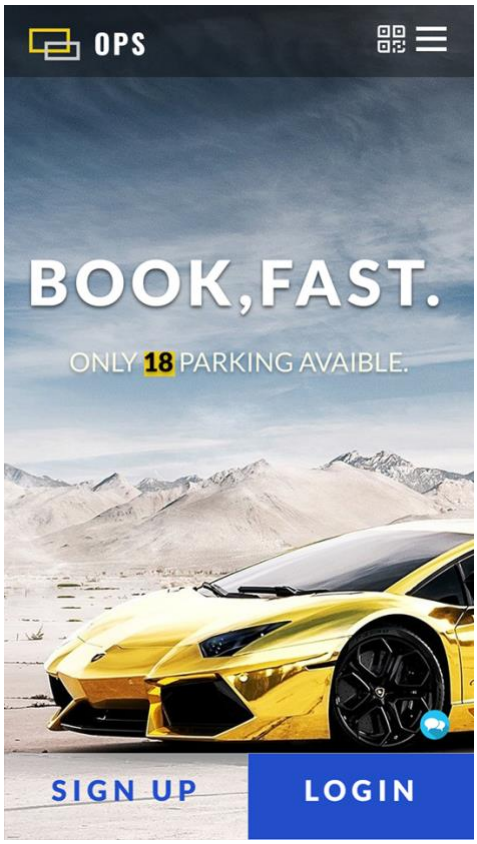

ENGLISH
ADMIN INTERFACE
OPS
HOME
CAMERAS
REVENUES
ONLY 18 PARKING AVAIBLE.
CAMERAS
REVENUES
© 2023 BY OR MOTORS. PROUDLY BY PPU STUDENTS


ENGLISH
OMAR TAKROURI
ARC
HOME
ABOUT
CONTACT
SCAN QR CODE
Scan QR code when entering and exiting Parking.
© 2023 BY POS. PROUDLY BY PPU STUDENTS


ENGLISH
SIGN IN /REGISTER
OPS
HOME
ABOUT
CONTACT
BOOK, FAST.
ONLY 18 PARKING AVAIBLE.
SIGN UP
LOGIN


ENGLISH
OMAR TAKROURI
ARC
HOME
ABOUT
CONTACT
PARKING RESERVATION
Number of Parkings Available 18
RESERVE
CANCEL RESERVATION
You have 01:00:00 to confirm your reservation
SHOW QR CODE


OPS
QR CODE
Scan QR code when entering and exiting Parking.
587-5556666
HOME
ABOUT
CONTACT
LOGIN
SIGN UP
LOGIN


OPS
BOOK,FAST.
ONLY 18 PARKING AVAIBLE.
SIGN UP
LOGIN

# Chapter 5: system testing and implantation.

## Overview:

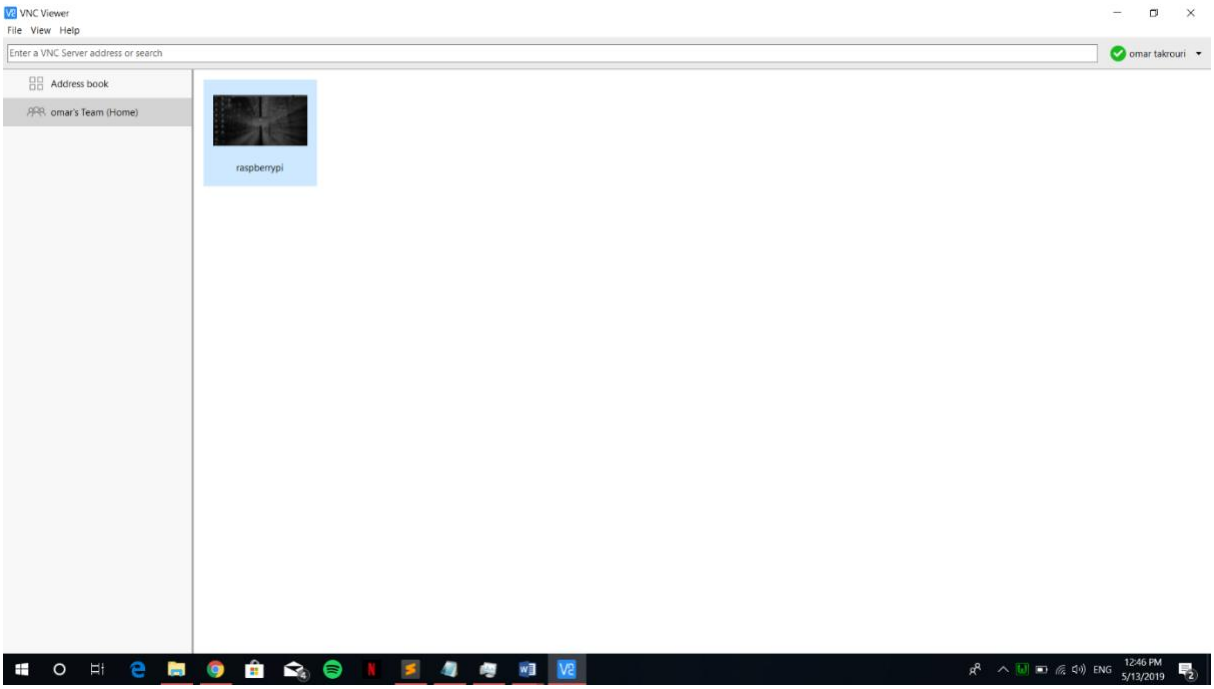
In this chapter we will describe the setup of development environment, and the software that we have used in our project, such as the operating system, the IDEs that have been used to build the project codes either on the master and slave units of the system, and the set of tools that needed to implement the system. In addition, we will describe how we test and debug the system over the milestones of the implementation.

## Implementation requirements

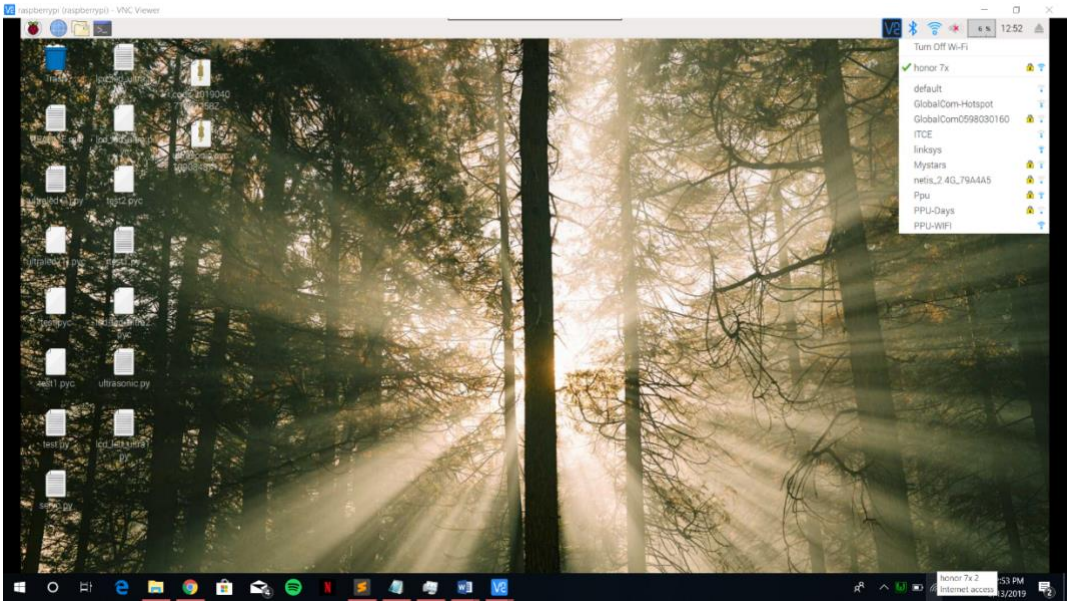
The requirements of system implementation summarized as follows:

1. Raspbian: is the operating system installed on the raspberry pi, it's the most common os used for this micro controller, and it provides a lot of features that could be used in all types of project. It contains a text editor that is usable for most of the programming languages.

2. VNC viewer: is a software installed on both the computer and the raspberry pi, its used to view the raspberry pi interface from the computer. It makes it easier to access the raspberry pi, instead of connecting it to a monitor and having to stay in one place.
The program can work in a local network, or it can be viewed from other networks when logged in with an email account.



In this picture. The image shows the VNC main interface. It shows that I'm logged in and it shows my devices.

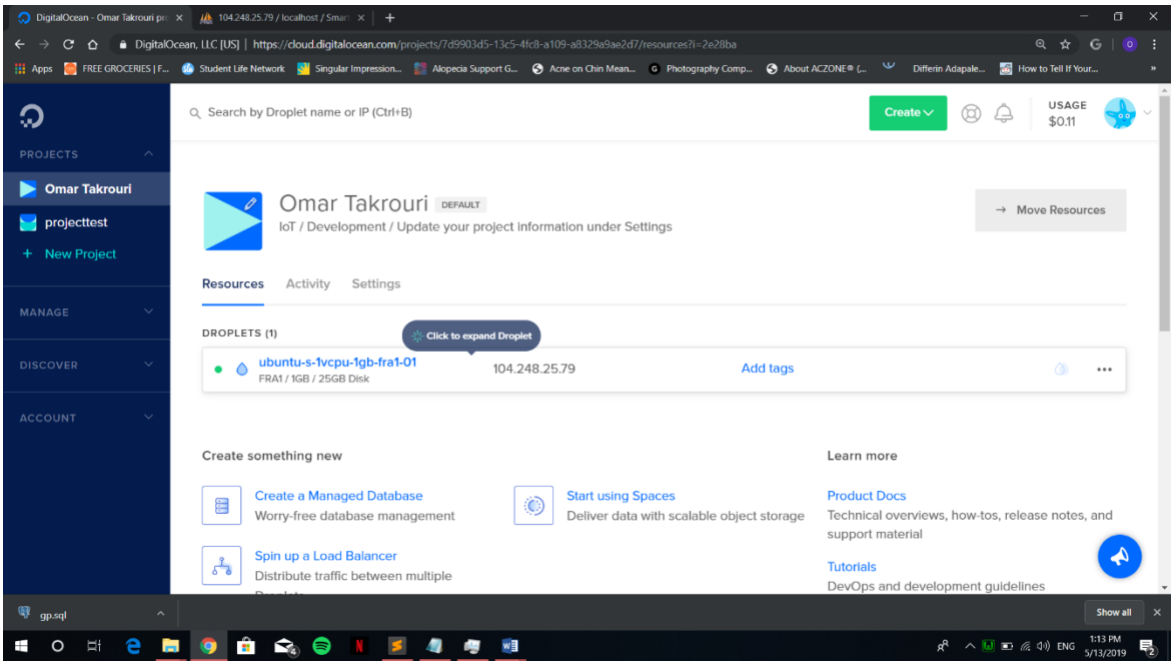


This is the main interface for the raspberry pi shown on the VNC software

3. Sublime: is a tool downloaded on windows to help write the PHP code and HTML, as well as editing the python code for the Raspberry

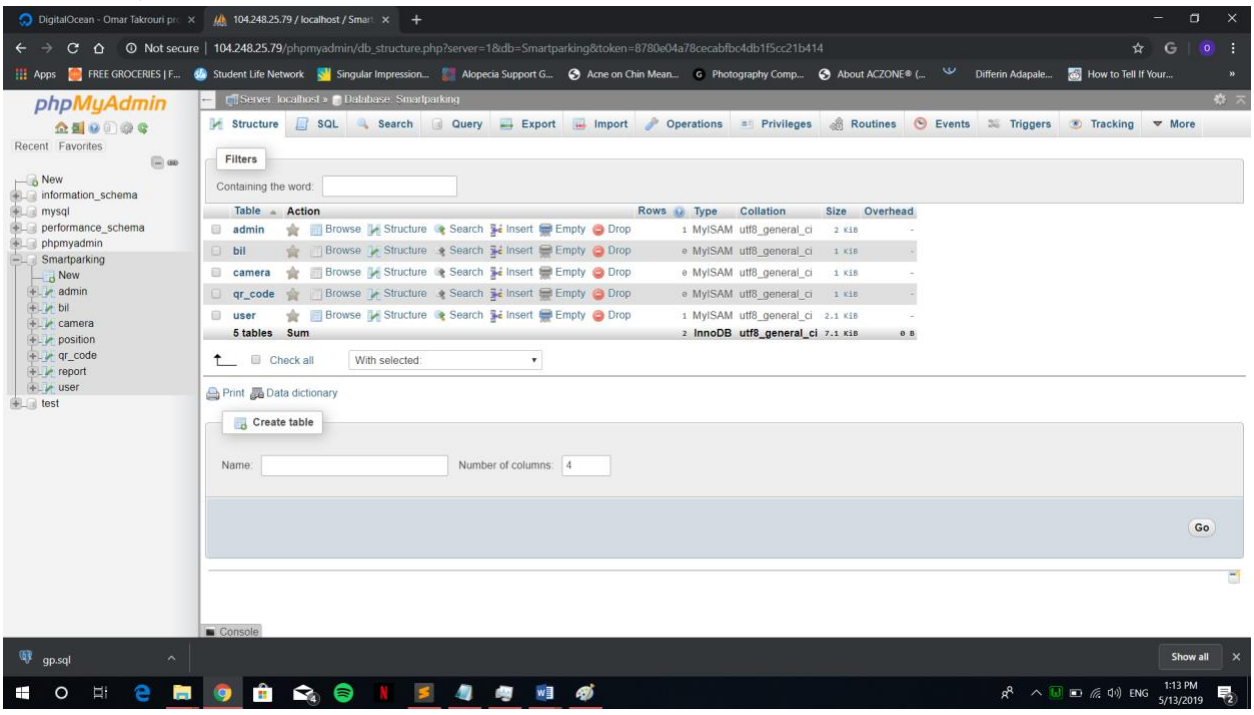
4. Wampserver: is a tool to test out the database before deploying it, it contains PHPmyadmin which is what is most important part for testing our site.

5. Digitalocean.com: is a website we used to upload the database on to. And to connect the website with the hardware side of the project
   First we need to create an account on the website and pay for the service. Which is about $5
   Then, from the dashboard we would create a droplet for the database, which uses Ubuntu to do.




Digitaloceans uses MYSQL, but doesn't use PHPmyadmin, so using Ubuntu we need to install PHPmyadmin.

To access it, we need to enter the IP address in the URL address bar.



# Chapter 6: future work and Conclusion

## Future work:

For future work, we hope to get this project implemented in real life. Some of the changes that would have to be made to the project would be changing the raspberry pi and using a suitable BLC capable of handling real life implantation. But the fundamentals would still be the same as our project and out prototype.

## Overall Evaluation

We are beginners in this field, and we have begun to learn all the necessary technologies from scratch, hoping to gather the maximum collection of information, for use in the implementation of a prototype that capable of meeting the system requirements. Indeed, we succeeded in implementing most of the requirements, but because of the lack of time - not suitable for us as beginners - we were unable to put up an independent and comprehensive prototype efficiently, and also the most critical issue is our inability to provide all the necessary components to implement whole project and test all required features with different possibilities, in addition to the trial and error process is relatively expensive for us as students, because they need more than one model for each component, and we have not been able to get the university support that requires complicated procedures, which we were unable to follow and adhere to, so we have a limited scope of implementation and testing.

## Conclusion

The main goal of our project is to develop a system that will make the experience of parking much easier to the user. This goal will be achieved by gathering the suitable hardware components and develop a friendly software that has a good and interactive graphical user interface (GUI). As well as making it easier in the field