

MAGIC PARKING SYSTEM

MIS-620 Analysis and Design of Information Systems / Team: Magic Parking

November 2, 2016

1. Overview of the system

Magic Parking is a software platform to make physical parking spaces available online for reservation. It includes following key functionalities

- Parking space owner can make the space available online with Magic Parking
- Customers can search for parking around them or at particular location and reserve them if available

2. Tools and methods used to implement

We intend to develop above system as a ubiquitous web application. We are also looking forward to integrate our web application with Android and iOS. We considered using following technologies/tools/techniques to achieve the system functionalities.

2.1 Internet Technologies:

- Client Side: HTML5, CSS3, Bootstrap3, JQuery1.11.1
- Server Side: Php5.x
- Database: MySQL
- Test Server: WAMP3.0.6

2.2 Hardware details:

- Programming Language: Embedded C
- Processor: Atmega328 8 bit micro controller
- Clock speed: 8 MHz
- Voltage: 3.3v
- Wi-Fi module: ESP8266
- Sensor: Ultrasonic range finder - SR04

3. Creating database and tables in MySQL

There are 9 Classes in our class diagram and they can be converted into corresponding database tables using the following code

Create and select the database

```
CREATE DATABASE magicParking
```

```
USE magicParking
```

Now create the table for Customer

```
CREATE TABLE CUSTOMER(
    user_id int unsigned auto_increment,
    fname varchar(30) NOT NULL,
    lname varchar(30) NOT NULL,
    email varchar(40) UNIQUE NOT NULL,
    pass varchar(40) NOT NULL,
    is_verified enum('TRUE','FALSE') NOT NULL DEFAULT 'FALSE',
    otp varchar(30) NOT NULL,
    phone varchar(15) DEFAULT NULL,
    car_num varchar(15) DEFAULT NULL,
    car_type enum('sedan','suv','hatchback','minivan/van','truck','convertible') DEFAULT NULL,
    can_reserve enum('TRUE','FALSE') NOT NULL DEFAULT 'FALSE',
    PRIMARY KEY(user_id)
)ENGINE=InnoDB
```

Create table for the parking space owner

```
CREATE TABLE OWNER(
    user_id int unsigned AUTO_INCREMENT,
    fname varchar(30) NOT NULL,
    lname varchar(30) NOT NULL,
    email varchar(40) UNIQUE NOT NULL,
    pass varchar(40) NOT NULL,
    otp varchar(30) NOT NULL,
    is_verified enum('TRUE','FALSE') NOT NULL DEFAULT 'FALSE',
    orgname varchar(20),
    address varchar(20) NOT NULL,
    zipcode varchar(5) NOT NULL,
    state varchar(15) NOT NULL,
    city varchar(15) NOT NULL,
    PRIMARY KEY (user_id)
)ENGINE=InnoDB
```

Create table for the parking lot monitor

```
CREATE TABLE MONITOR(
    user_id int unsigned AUTO_INCREMENT,
    fname varchar(30) NOT NULL,
    lname varchar(30) NOT NULL,
    email varchar(40) UNIQUE NOT NULL,
    pass varchar(40) NOT NULL,
    otp varchar(30) NOT NULL,
    is_verified enum('TRUE','FALSE') NOT NULL DEFAULT 'FALSE',
    owner_id int unsigned NOT NULL,
    phone varchar(15) NOT NULL,
    PRIMARY KEY (user_id),
    FOREIGN KEY (owner_id) REFERENCES OWNER(user_id)
    ON DELETE CASCADE
)ENGINE=InnoDB
```

Create table to store the parking space details

```
CREATE TABLE PARKINGSAPACES(
    space_name varchar(10) NOT NULL,
    owner_id int unsigned NOT NULL,
    lat varchar(10) NOT NULL,
    lon varchar(10) NOT NULL,
    altitude int NOT NULL,
    accuracy int NOT NULL,
    PRIMARY KEY (lat,lon,altitude),
    FOREIGN KEY (owner_id) REFERENCES OWNER(user_id)
    ON DELETE CASCADE
)ENGINE=InnoDB
```

Create table to store the parking space availability

```
CREATE TABLE AVAILABILITY(
    space_id int unsigned auto_increment,
    lat varchar(10) NOT NULL,
    lon varchar(10) NOT NULL,
    altitude int NOT NULL,
    is_occupied enum('YES','NO') DEFAULT 'YES',
    is_reserved enum('YES','NO') DEFAULT 'NO',
    PRIMARY KEY(space_id),
    FOREIGN KEY(lat,lon,altitude) REFERENCES parkingspaces(lat,lon,altitude)
    ON DELETE CASCADE
)ENGINE=InnoDB
```

Create table to store sensor details

```
CREATE TABLE SENSORS(
    sensor_id int unsigned auto_increment,
    space_id int unsigned NOT NULL,
    reg_ip varchar(20) NOT NULL,
    state enum('Occupied','vacant','disabled'),
    PRIMARY KEY(sensor_id),
    FOREIGN KEY(space_id) REFERENCES AVAILABILITY(space_id)
    ON DELETE CASCADE
)ENGINE=InnoDB
```

Create table to store ticket details

```
CREATE TABLE TICKETS(
    ticket_id int unsigned auto_increment,
    user_id int unsigned not null,
    space_id int unsigned not null,
    time_of_booking datetime not null,
    check_in_time datetime not null,
    check_out_time datetime not null,
    state enum('booked','cancelled','suspended','finished') default 'booked',
    extended_check_out datetime default NULL,
    PRIMARY KEY(ticket_id),
    FOREIGN KEY(user_id) references customer(user_id) on delete cascade,
    FOREIGN KEY(space_id) references availability(space_id) on delete cascade
)ENGINE=InnoDB
```

Create table to store complaints from customers

```
CREATE TABLE COMPLAINTS(  
    report_id int unsigned auto_increment,  
    ticket_id int unsigned not null,  
    time_reported datetime not null,  
    complaint_status enum('placed','resolved') default 'placed',  
    PRIMARY KEY(report_id),  
    FOREIGN KEY(ticket_id) references tickets(ticket_id)  
    ON DELETE CASCADE  
)ENGINE=InnoDB
```

Finally, create table to store notifications data

```
CREATE TABLE NOTIFICATIONS(  
    notif_id int unsigned auto_increment,  
    ticket_id int unsigned not null,  
    remind_at datetime not null,  
    notif_status enum('active','expired') default 'active',  
    PRIMARY KEY(notif_id),  
    FOREIGN KEY(ticket_id) references tickets(ticket_id)  
    ON DELETE CASCADE  
)ENGINE=InnoDB
```