

-----  
My name: Meihui Wu

My student number: 8327464

My email address: mw285@uowmail.edu.au  
-----

## 1 SET UP ENVIRONMENT

- Step 1: open "mysqlConnection.php"
- Step 2: change the below variables based on your database information

```
$serverName = "localhost:3306";  
$uname = "root";  
$password = "ClaireWu0207*";
```

- Step 3: open the "index.php", then the connection is built

**NOTE:** once connected to MySQL, the database will be created and selected, and all the related tables and records will be generated, and they are constrained that will not be generated repeatedly.

## 2 TABLE DESCRIPTION

### 2.1 User Table

- **Attributes**

uid (UserID), name, surname, phone, email, password and type (user/administrator).

- **Explanation**

Each user (user/administrator) has a unique uid, phone, and email. The password is encoded by md5.

### 2.2 ParkingLocation Table

- **Attributes**

pid (ParkingID), location, description, parking\_space (capacity), cost\_per\_hour.

- **Explanation**

Each parking location has a unique pid.

Parking time within 1 hour counts as 1 hour, over 1 hour but less than 2 hours counts as 2 hours, and so on.

### 2.3 Parking Table

- **Attributes**

uid (UserID), pid (ParkingID), start\_time (Parking Start Time), finish\_time (Parking Finish Time).

- **Explanation**

Each user can park in several locations simultaneously, if available.

Users can park multiple times across different locations, but only one active parking record is allowed per location at any given time.

## 3 FILE LIST EXPLANATION

### 3.1 mysqlConnection.php

Connecting MySQL, creating database, selecting database, creating table, inserting initial **Parking Location** records.

### 3.2 createTable.php

Designing the table structure.

### 3.3 common.css

Common CSS style across the application.

### 3.4 index.php

Setting up database environment and displaying login form and sign-up link.

### 3.5 signup.php

Displaying sign-up form and validating entered information to allow user/administrator to register and store their information to the User table.

### 3.6 login.php

Validating entered information by retrieving User table information to allow registered user/administrator to login.

### 3.7 roleClass.php

Constructing classes and functions to meet design requirements.

- **class Role**

Including the common function (searching a parking location) for all roles (user/administrator) in the application.

- **class User**

Extending class Role with user-related functions:

- list all parking locations with available spaces
- list all parkings that has used in the past
- list all parkings that currently using
- check-in for a parking (with notice for the cost per hour of the parking)
- check-out of a parking (with notice for total cost that needs to be paid)

- **class Administrator**

Extending class Role with administrator-related functions:

- insert/edit parking location into the app
- check-in/check-out user for a parking
- list: all, currently with available spaces, currently full parking locations
- list: all users, all users checked in a parking

### 3.8 userFunction.php

Interface of user-related functions and the option to logout.

### 3.9 administratorFunction.php

Interface of administrator-related functions and the option to logout.