

**NAME: Rhys Quilter: - K00241356**

### **Name of your project (0%):**

The name of my project is: Perfect Parking

### **Description of business problem being solved/purpose of system:**

The purpose of perfect parking is to try and solve the widespread problem of traffic congestion and lack of parking especially in Limerick City. Parking is one of the main issues that causes traffic congestion especially in cities, people are selfish when they drive and begin to rush other road users that might be new in the city and looking for parking, when they are rushed and put under pressure, they can end up parking illegally on the side of the road which can cause traffic to be stopped. With my application perfect parking I'm hoping to ease the stress and anxiety that road users face in finding appropriate parking which gives them a price range that they might be looking for and a location straight to the parking lot, this will eliminate traffic congestion, the waste of petrol and diesel from driving around looking for parking and also people's time.

### **Numbered/Ordered List – Detailed Specification**

- **Type** Map API  
Menu  
Drilldown  
Map API
- **Location** Home page  
Login page  
Navigation bar  
Select parking  
View parking

#### **1.0 User Functions**

##### **1.1 Register**

**Type:** menu form item

**Location:** Home page (not logged in/registered users)

**Action:** once a Guest user opens the application they will be greeted with a log-in page, there will also be a **button** underneath the form that states, "don't have a log in register now". If the guest attempts to login and doesn't have a log-in, they will be prompted with an error message stating them to register. Once the guest selects the registration button, they will be brought to the register form, here they will be asked to enter e-mail, password, confirm password. Once registered their details will be sent and saved to the UserDetails table in the DB. They will then be redirected to the Log-in page.

## 1.2 Login

**Type:** menu form item

**Location:** Login page (not logged in users)

**Action:** the user will provide an e-mail and password to login. These details will be verified by the DB by a lookup. If the details entered is correct the user will be logged in and be brought to the select parking/parking lookup page. If their information is incorrect, they will be prompted to reenter e-mail and password.

## 1.3 Logout:

**Type:** menu item

**Location:** Navigation bar item (for logged in users)

**Action:** when selected in the navigation bar of a logged in user the user will be logged out and sent back to the login page.

## 1.4 Select Parking

**Type:** Menu item

**Location:** Select parking location (Accessed from navigation bar)

**Action:** once a user is logged in they can access the **select parking** page, from here users can select parking in the city that they would like to view. There will be a list of all the different parking lots in the city.

## 1.5 View Parking

**Type:** Parking Drilldown

**Location:** View parking page (Accessed through select parking)

**Action:** this page will show all the information about the selected parking, it will show the parking price per hour, and per day. There will also be a map that can be viewed by the user and shows the location of the parking.

## 1.6 Get direction

**Type:** Map API

**Location:** View parking page (Accessed from the map)

**Action:** when the user is viewing the map, if the user clicks the map, it will show information such as the address and the name of the parking, it will also prompt the user to get the directions from their location to the parking lot, this is done through google maps.

## 1.7 View prices

**Type:** button

**Location:** View parking page

**Action:** when the user is viewing the parking there will be a button stating view prices, when the user presses this button, it will show the parking rates for that parking lot.

## 2.0 Administration functions

### 1. Login:

**Type:** menu form item

**Location:** Login page (not logged in users)

**Action:** the user will provide an e-mail and password to login. These details will be verified by the DB by a lookup. If the details entered is correct the user will be logged in and be brought to the select parking/parking lookup page. If their information is incorrect, they will be prompted to reenter e-mail and password.

#### 1.1. Logout:

**Type:** menu item

**Location:** Navigation bar item (for logged in users)

**Action:** when selected in the navigation bar of a logged in user the user will be logged out and sent back to the login page.

#### 1.2. View Registered Users:

**Type:** Menu Item

**Location:** view all users

**Action:** When selected this menu item will present a table of all user's IDs and usernames along with status (active).

#### 1.3. Edit Price:

**Type:** Button

**Location:** Edit parking

**Action:** when the admin presses the button it will allow him to update the price of parking per hour/day/week, when this is done the users will be greeted with a new price once it has been successfully updated.

#### 1.4. view User

**Type:** Menu item/button

**Location:** view user

**Action:** when the admin is viewing all users, they can then select a user which they wish to view the details for. When they press on the user it will view that one user's details.

### 1.5. Delete user

**Type:** button

**Location:** view user

**Action:** when the admin is viewing the user details, they will also have a button which they can use to delete user, this might have to be done in certain circumstances such as user has violated the terms and conditions, when the admin presses the delete user button it will delete all the user details from the database.

## Preliminary Database design

**Haven't designed my database yet!!!**

**I have drawn up some database prototypes hoping to get this done for the end of the week, if I do ill email you the updated word document**