

National College of Ireland

BSc (Hons) in Business Information Systems - 2015/2016

David Ryan - 06685129
x06685129@student.ncirl.ie

Instructor Booker

Technical Report



Table of Contents

Executive Summary	4
1 Introduction	5
1.1 Background.....	5
1.2 Aim.....	6
1.3 Technologies	6
2 System.....	8
2.1 Requirements	8
2.2 Functional requirements	9
2.2.1 Requirement 1: Unregistered User.....	9
2.2.2 Description & Priority	9
2.2.3 Use Case	9
2.2.4 Requirement 2: Registered Super User	11
2.2.5 Description & Priority	11
2.2.6 Use Case	11
2.2.7 Requirement 3: Instructor Selection	13
2.2.8 Description & Priority	13
2.2.9 Use Case Diagram	14
2.2.10 Requirement 4: Booking process	15
2.2.11 Description & Priority	15
2.2.12 Use Case	16
2.2.13 Data requirements.....	17
2.2.14 Performance/Response time requirement.....	18
2.2.15 Availability requirement	18
2.2.16 Maintainability requirement	18
2.2.17 Portability requirement	18
2.2.18 Reusability requirement.....	18
2.3 Design and Architecture.....	18
2.3.1 System Architecture Internal	19
2.3.2 System Architecture External	19
2.3.3 Algorithms	20

2.4	Implementation	20
2.4.1	Back-End development	20
2.4.2	Front-End development:	25
2.5	Graphical User Interface (GUI) Layout.....	25
2.5.1	GUI - Mobile	26
2.5.2	GUI- Desktop	34
2.6	Customer Testing:	42
2.7	Testing	42
2.8	Evaluation	42
3	Conclusion	44
4	Further development or research.....	45
5	Bibliography	46
6	Appendix.....	47
6.1	Project Proposal	47
6.2	Background:.....	47
6.3	Technical Approach:	48
6.4	Project Plan:	49

Executive Summary

My project idea is to easily connect learner drivers with driving instructors via a web application.

Currently if a learner is looking to book a driving instructor they must monotonously search through directories in their area. Alternatively a lot of bookings come through word of mouth which is not really sustainable. There are some companies such as Irish School of Motoring (ISM) who can schedule appointments via a website. This does not however facilitate private instructors who may not have the expertise or funding to develop a site themselves.

The application will have instructor profiles, contact details, locations, merits and qualifications of the driving instructors which the user can browse when looking for a lesson. The user can also book a test with an instructor, leave a review, comment and star-rate and take driving theory quizzes among other activities. The main goal of this app is to enable learner drivers to have a wider choice of driving instructors whose experience and ability are transparent.

My aim is to create a web app which is user friendly and provides all the necessary information required for both the driver and instructor. I will implement a payment method using a payment gateway so customers can actually pay for their lessons through the site. This will turn the web app into a more holistic product with end to end functionality. Providing a review process is what I feel will attract the most attention. Currently learner drivers have to take a risk of perhaps paying an instructor who is not at the level they require. The review features will hopefully eliminate this issue.

The entire project is going to be written in PHP and will communicate with a database using SQL commands. The web app will be hosted on an apache web server. JavaScript and html/CSS will be used to provide a user friendly experience.

1 Introduction

This document will provide both a detailed and high level view of the web application Instructor Booker. The application was developed to fill a gap in the diver instructor market. The technologies being used in the development of the application are scalable and robust. I used PHP as the server side programming language in this project which connects to a MySQL database.

1.1 Background

From my research I found that this type of service is few and far between, I therefore feel there could be a large market within reach. I expect learner drivers looking to pass their test to be the main users of this application. All new first time learner permit holders with effect from 6th December 2010 for motorcycles and 4th April 2011 for cars are required to undertake mandatory initial basic training (IBT) with an approved driving instructor (ADI). The course must be 16 hours for motorcycle and 12 hours for car licenses. Since 12 lessons are now mandatory for all new drivers this app will appeal to them. On the other hand, e driving instructors could also be classed as a benefiter of this product as they would also be a user of the service. They will have their own profile on the app which would, in a sense, act as a business card for them. Instructor will also be able to reach a larger number of potential pupils via the app, in the past they may have been limited due to geographical restrictions.

At the end of 2013 there was an estimated 238,000 learner drivers in the Republic of Ireland. Currently 12 lessons cost approx. €300, this totals at approx. €71 million which shows the potential market for a product which assists learners to acquire the necessary skills to pass their driving test.

I held meeting with a number of people looking to book a driving instructor and learnt that most would use Google or ask a friend who had recently taken lessons.

From these meetings I gathered that no one was using apps in order to find instructors. For this reason I decided to develop a web application which user can search via Google, Yahoo etc. The application is mobile dynamic and therefore adjusts for users accessing the site via mobile or tablet.

1.2 Aim

The main aim of this web application is to extend the market reach for both the driver instructor and the learner driver alike by providing an interactive and user friendly experience. Instructor Booker will provide a platform in which the user can obtain driving knowledge and then test the knowledge gained through an interactive quiz section. The web application is built to be simple but effective, providing all the necessary tools one could require in order to feel confident in attempting their full licence test.

1.3 Technologies

The technologies used in the development of the web application are:

- PHP
- SQL
- JavaScript
- HTTP
- HTML
- CSS
- Bootstrap

PHP – PHP was used to create the data logic level of the web application. The main functional aspects of the app used PHP in conjunction with the other technologies mentioned above.

SQL – MySQLi queries were used to insert data and query the database to obtain information. All sensitive information e.g. passwords are hashed using an SHA256 algorithm.

JavaScript – JavaScript provides dynamic front end functionality.

HTTP – A HTTP post request is used to communicate to web servers. The web servers which are being targeted are PayPal and Realex Payment Gateway.

HTML – HTML 5 delivers an interactive user experience.

CSS – CSS allows the web app to become mobile responsive and scalable.

Bootstrap: Bootstrap is a free and open-source front-end framework for creating websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

2 System

In this section, I will outline the structure of this technical report. I will present a detailed description of the requirements which have to be met in order for this application to function. I will then describe the Design and Architecture of the System and how the different components link together in order to provide an interconnected system.

2.1 Requirements

The learner driver will be able to easily navigate the system. No log in is required unless they wish to view the instructors section. They can select from a range of options such as Quiz, Road Theory or Contact. The user interface was designed to provide a simple but enjoyable user experience.

The user can then book a lesson through the application via the booking system contained in the instructors section. Alternatively the user can book directly over the phone with the instructor. The user will be prompted to register if accessing the instructors section. The registration form is simple and quick to complete.

If the app is used to make the booking the instructor and learner will notified via an email once the payment is processed.

If the user is an instructor they will have the option to register/login. In order to register they will complete a registration form which only requires necessary information about the instructor. The instructor will then receive a message to say their application has been submitted to be reviewed.

If the instructor has already registered they will have access to their profile and reviews. The reviews cannot be edited by an instructor only by an administrator.

2.2 Functional requirements

Key requirements:

- Database creation
- Manual database population
- PHP server side communication
- Graphic and Design (HTML, CSS)
- Automated database population
- Database communication
- API connection

The above is required to successfully offer an interactive web app which is easily updated. PHP was used as the server side language along with mySQLi which connects to the MySQL database.

HTML, CSS and JavaScript were be used to create the UI.

2.2.1 Requirement 1: Unregistered User

2.2.2 Description & Priority

This is a functional requirement as instructor's details should not be accessible to unauthorised users.

2.2.3 Use Case

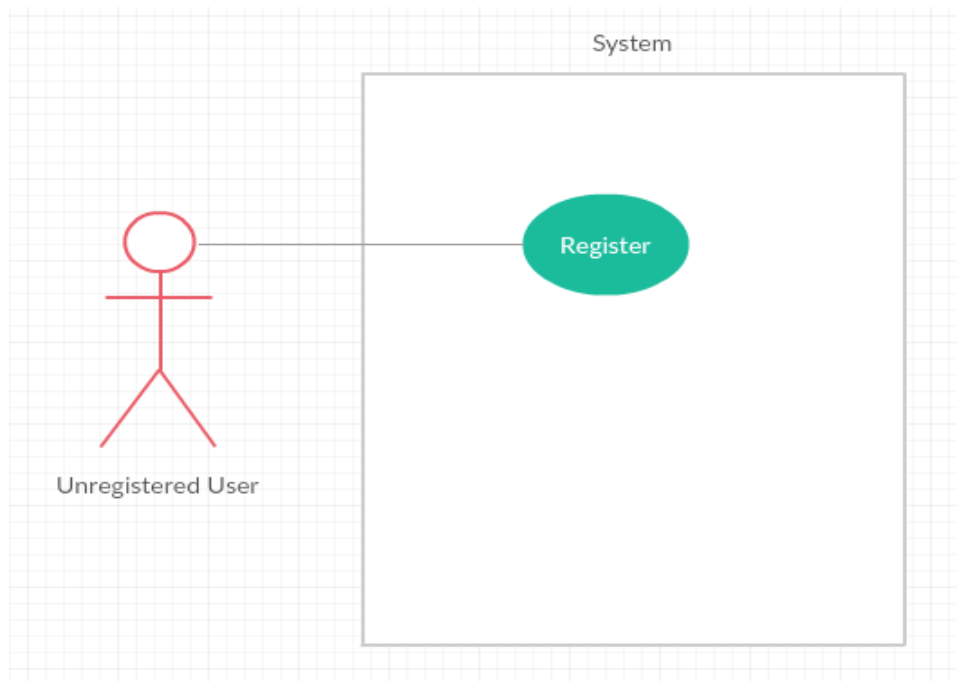
Scope

The scope of this use case is to give an overview of the registration process for users accessing the web app.

Description

This use case describes the new user registration

Use Case Diagram



Flow Description

Precondition

The system is active.

Activation

This use case starts when a user accesses the Instructor Section.

Main flow

1. The user accesses the Instructor Section.
2. The system loads registration
3. The user inputs details
4. System stores user details

Alternate flow

A1: Email address is taken

5. The system indicates that Email address is taken
6. The system notifies the user
7. The use case continues at position 3 of the main flow

Exceptional flow

E1: Password does not meet criteria

8. The system indicates the password does not meet requirements
9. The user re-enters password
10. The use case continues at position 5 of the main flow

Termination

The system presents the home page

Post condition

Account is created and user is logged in.

2.2.4 Requirement 2: Registered Super User

2.2.5 Description & Priority

The requirement for a super user section is highly important as the instructor need to ability to update their profile and have different options than standard users.

2.2.6 Use Case

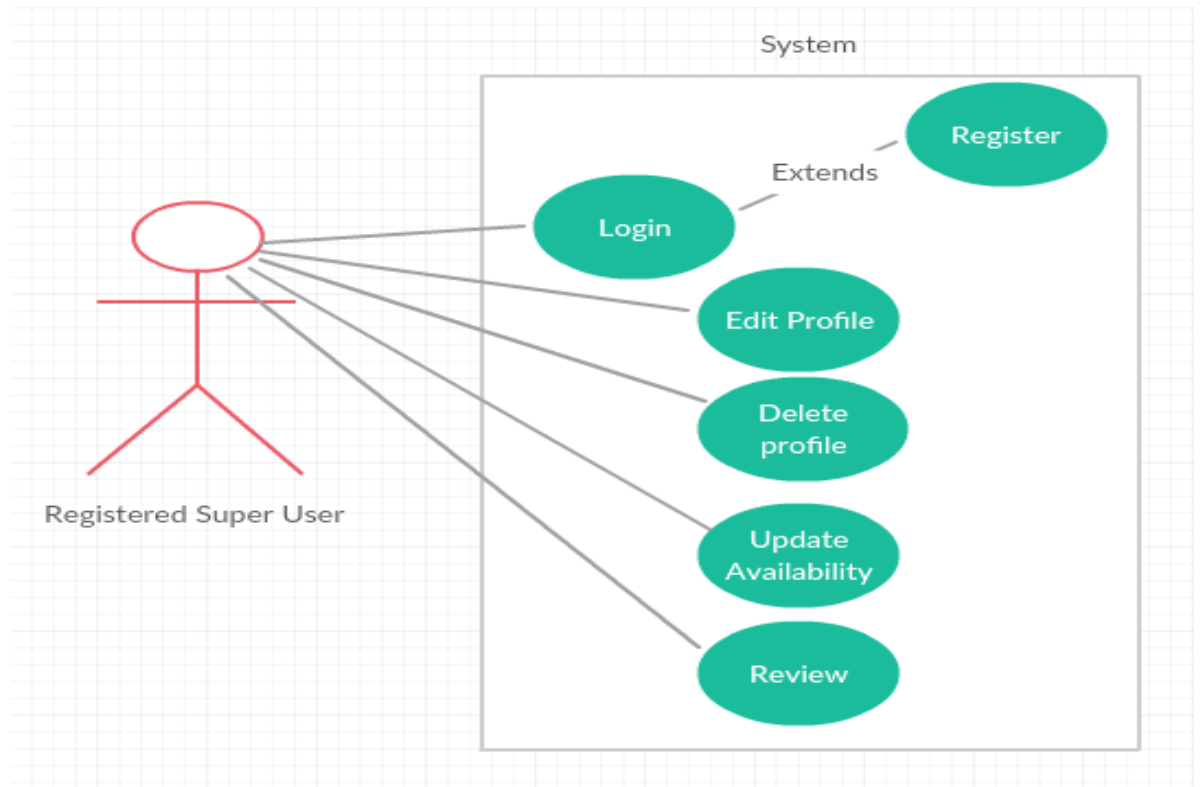
Scope

The scope of this use case is to provide interaction of a registered super user.

Description

This use case describes the access available section for instructors.

Use Case Diagram



Flow Description

Precondition

The system is running.

Activation

This use case starts when an Super User logs into their account

Main flow

11. The system identifies the user
12. The super user edits their profile
13. The system updates the DB
14. The user logs out

Alternate flow

A1: The enter user enters incorrect DOB

- 15. The system notifies the user
- 16. The re-enters correct DOB
- 17. The use case continues at position 3 of the main flow

Exceptional flow

E1: The user deletes profile.

- 18. The system prompts a warning message.
- 19. The user proceeds.
- 20. The use case ends

Termination

The system presents a termination message

Post condition

User logged out.

2.2.7 Requirement 3: Instructor Selection

2.2.8 Description & Priority

This is an essential function of the application as the main concept is to allow users to book an instructor.

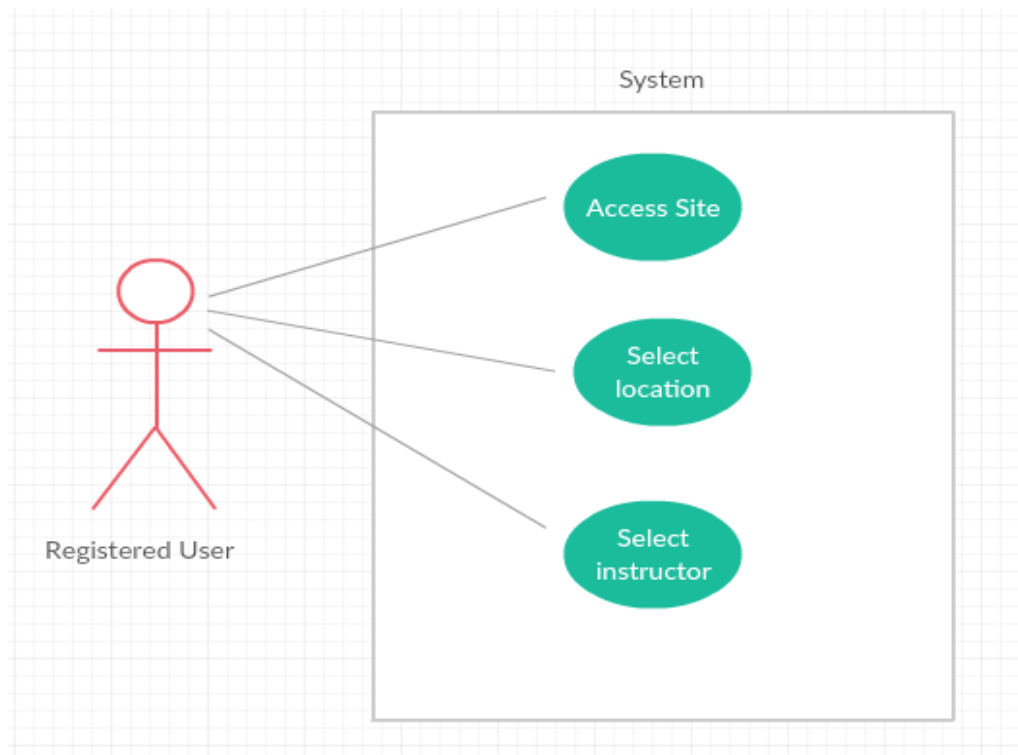
Scope

The scope of this use case is to allow a user to select a driving instructor.

Description

The user will navigate through the app in order to locate a suitable instructor.

2.2.9 Use Case Diagram



Flow Description

Precondition

The system is active.

Activation

This use case starts when a user navigates to the site.

Main flow

21. The user selects “Instructors” tab.
22. The system loads the “Instructors” page.

23. The user selects their location.
24. The system sends a request to the database.
25. The system displays the drivers in the location.
26. The user selects an instructor.

Alternate flow

A1: User selects an area with no instructors.

27. The user selects their location.
28. The system send a request to the database.
29. The system notifies the user.
30. The use case continues at position 2 of the main flow.

Exceptional flow

E1: User cancels search selected

31. The user selects their location
32. The system send a request to the database.
33. The system displays the drivers in the location.
34. The user cancels their search
35. The use case continues at position 2 of the main flow

Termination

The system presents the next step in the booking process.

Post condition

The system waits for further instruction.

2.2.10 Requirement 4: Booking process

2.2.11 Description & Priority

The booking process is a key element in the user experience as it allows the user to finalise an appointment with their instructor.

2.2.12 Use Case

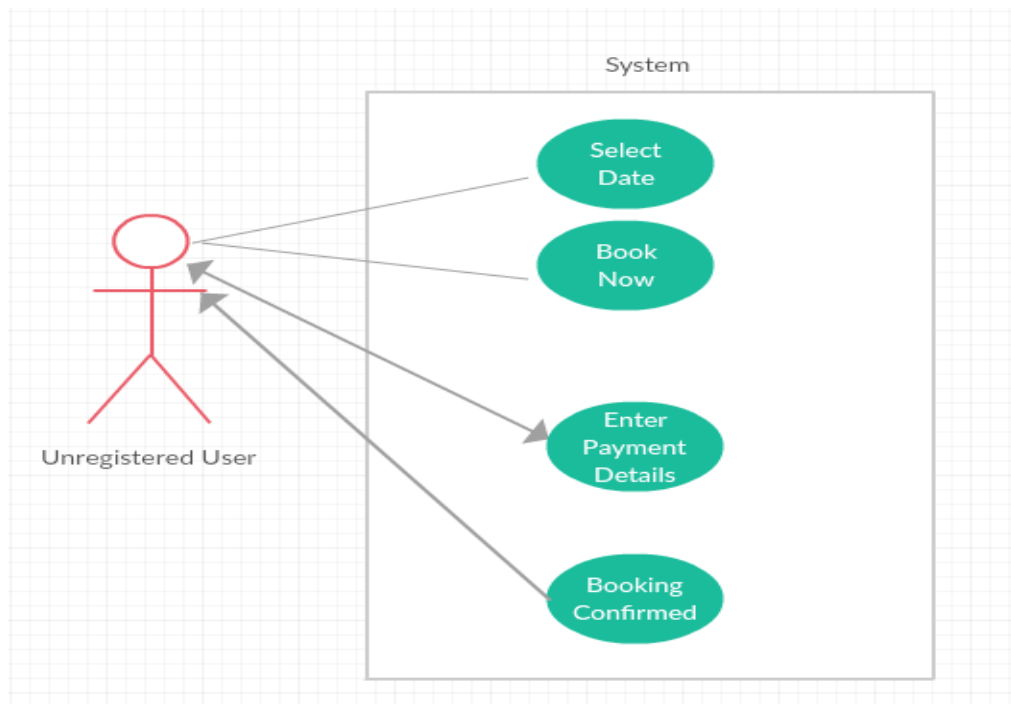
Scope

The scope of this use case is to provide an overview of the booking process.

Description

This use case describes the interaction of the user when navigating through the booking.

Use Case Diagram



Flow Description

Precondition

The system is an active state.

Activation

This use case starts when the users selects a date.

Main flow

- 36. The system identifies the available dates.
- 37. The user clicks “book now”.
- 38. The system send a request to the payment gateway.
- 39. The user enters their payment details.
- 40. The system displays a confirmation message.

Alternate flow

A1: The transaction is declined

- 41. The system notifies the user
- 42. The user resubmits booking
- 43. The use case continues at position 3 of the main flow

Exceptional flow

E1: The customer cancels booking

- 44. The user cancels before booking
- 45. The system updates available dates in DB
- 46. The use case continues at position 1 of the main flow

Termination

The system presents the finalised booking information.

Post condition

The system logs the successful booking.

2.2.13 Data requirements

All sensitive data e.g. Passwords must be securely stored. All passwords are be hashed using a sha256 hashing algorithm.

All data is securely stored in the database with no unauthorised access to the content. Payments are processed through a secure third party gateway using a redirect method. This pushes the PCI compliance requirements to the payment processor. The server will be located in a secure data centre

2.2.14 Performance/Response time requirement

The responses from the database will be as close to real time as possible.

2.2.15 Availability requirement

The Web App will be available 24/7 – preventing hosting issues.

Downtime forecasted to be 1 hour per month for application maintenance/errors.

2.2.16 Maintainability requirement

The database is easily updated through the backend admin control.

The maintenance of the web app will be carried out remotely. Any updates to software code and database design can be deployed remotely without affecting projected uptime figures.

2.2.17 Portability requirement

The web app is accessible through all devices with an internet connection.

The app does not require installation of software and is not device specific.

2.2.18 Reusability requirement

The software can be altered to create similar functioning app without much programming changes.

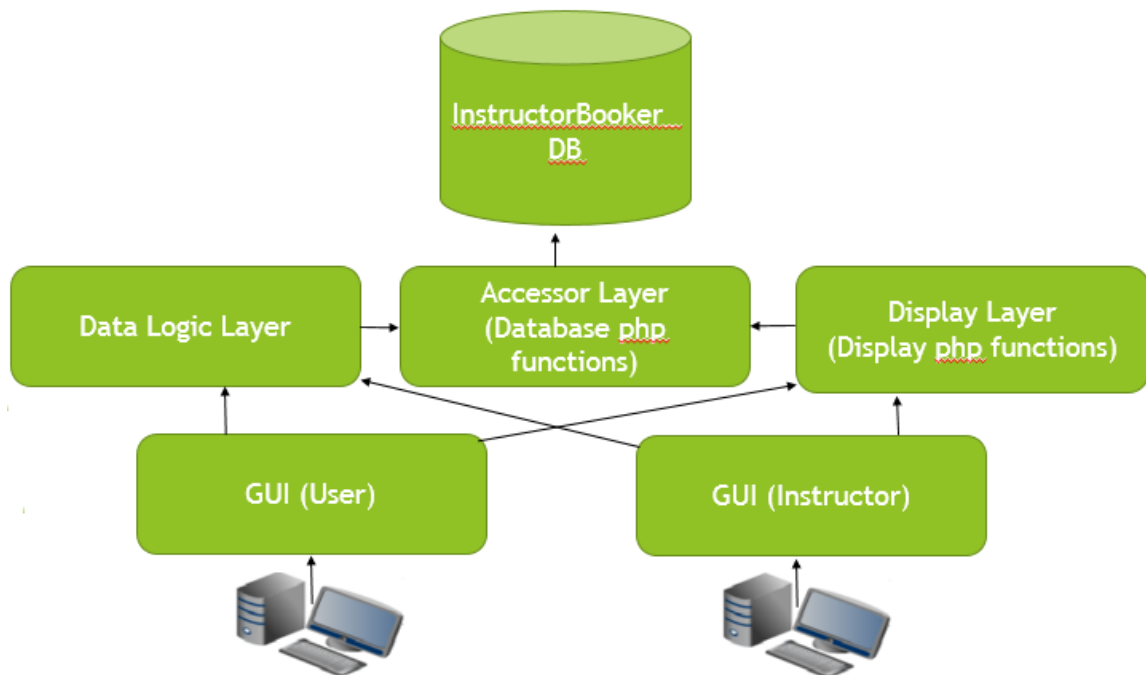
2.3 Design and Architecture

Instructor Booker was built using a Top Down architecture design.

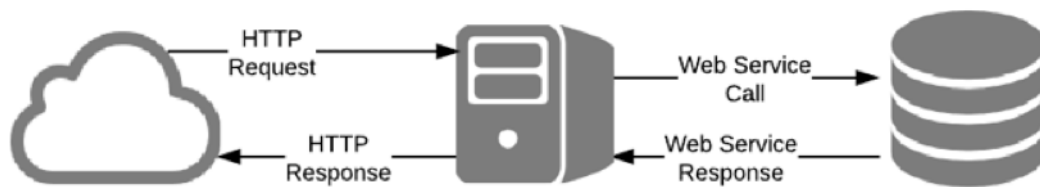
Top-down architecture starts with a broad overview and understanding of the website's strategy and goals, and creates a basic structure first. From there, content relationships are refined as the site architecture grows deeper, but it's all viewed from the overall high-level purpose of the site.

Using this methodology allowed me to keep the user perspective in mind while building the application.

2.3.1 System Architecture Internal



2.3.2 System Architecture External



2.3.3 Algorithms

SHA-1 (Secure Hash Algorithm 1) - is a cryptographic hash function designed by the United States National Security Agency. SHA-1 produces a 160-bit (20-byte) hash value known as a message digest. I used the SHA-1 hashing algorithm to hash the data being posted to the Realex Payments API. Realex Payments API supports two hashing functions MD5 and SHA1, I decided to implement a SHA1 algorithm as my research showed MD5 as the weaker of the two. MD5 generates a 128 bit string whereas SHA-1 produces a 160 bit string.

SHA-256 (Secure Hash Algorithm) is another cryptographic hash function. A cryptographic hash is like a signature for a text or a data file. SHA-256 algorithm generates an almost-unique, fixed size 256-bit (32-byte) hash. Hash is a one way function – it cannot be decrypted back. This hashing algorithm was used to hash user passwords in the database. SHA-256 is a highly robust and secure hashing function.

2.4 Implementation

The web application was developed using PHP, MySQLi, JS, HTML and CSS. The project was broken into two areas of development, Back-End and Front-End.

2.4.1 Back-End development

The first task was to create a database, from there I built db connections.

I created a main database class which communicates with a MySQL database. Each class within the application calls functions which are defined in the dbconnect.php class. This provides a more manageable system and reduces unnecessary database queries.

The database consists of three tables Users, Instructors, Quiz. Each table contains a number of columns which are used to store relative information.

A customer registration and login page were created to allow users to access the instructors section of the site. This feature was developed intentionally to prevent against unauthorised users having access to view the instructor's details.

Below is code snippet from the login.php class:

```
<?php
session_start ();
include_once 'dbconnect.php';
include_once 'sql.php';

$email = "";

// TODO add comments (validate email)
This confirms if the password and email are set before querying the database.
if (isset ( $_POST ['form-username'] ) && isset ( $_POST ['form-password'] ) )
{
```

Below queries the database to confirm login details.

```
    $dbManager = new DBManager ();

    $conn = $dbManager->connect ();

    $stmt = $conn->prepare ( GET_USERS );

    $email = $_POST ['form-username'];

    $stmt->bind_param ( "s", $email );

    $stmt->execute ();

    $result = $stmt->get_result ();

    $userInformation = $result->fetch_assoc ();

    // Below hashes the password in the POST and compares it to hashed the
    password stored in the database.
    if (hash ( "sha256", $_POST ['form-password'] ) === $userInformation
['password']) {

    // Below triggers the redirect to the instructor.php page.
        $_SESSION ['firstname'] = $userInformation ['firstname'];
        header ( "Location: instructors.php" );
    }
}

?>
```

Instructors are added manually through the database control panel. Instructor registration process differs from a regular user's as information is required to confirm they are a qualified instructor.

I created a registration form which instructors can complete which is sent via email and actioned accordingly.

The instructors.php class provides a list of locations which a user can select. Once a location is selected the database is queried to retrieve all the instructors in that area. The rating of the instructor is available and is determined by getting the average of stars which have been allocated to the instructor. The instructor's availability is also provided.

The request.php class sends the transaction request to the Realex Payments API. It also handles the PalPal API request. Using the Realex Payments developer guides I was able to integrate the payment system on the web application. Below is a snippet of code from the request.php class.

<?

```
// This is the account information which is set up with Realex Payments
$merchantid = "instructorbooker";
$secret = "*****";

// The code below is used to create the timestamp format required by Realex
Payments
$timestamp = strftime("%Y%m%d%H%M%S");
mt_srand((double)microtime()*1000000);

// The order id must be unique.
// Below is a combination of the timestamp and a random number which is used to
generate the order ID.
$orderid = $timestamp."-".mt_rand(1, 999);

// In this example these values are hardcoded. This will be set when the amount
is select by the customer.
$curr = "EUR";
$amount = "3500";

// Below is the code for creating the digital signature using the SHA1 algorithm
provided by PHP
$tmp = "$timestamp.$merchantid.$orderid.$amount.$curr";
$sha1hash = sha1($tmp);
```

```

$tmp = "$sha1hash.$secret";
$sha1hash = sha1($tmp);

?>
// Below is the request which is posted to the Realex Payments API

<form action=https://hpp.sandbox.realexpayments.com/pay method=post>

<input type=hidden name="MERCHANT_ID" value="<?=$merchantid?>">
<input type=hidden name="ORDER_ID" value="<?=$orderid?>">
<input type=hidden name="CURRENCY" value="<?=$curr?>">
<input type=hidden name="AMOUNT" value="<?=$amount?>">
<input type=hidden name="TIMESTAMP" value="<?=$timestamp?>">
<input type=hidden name="SHA1HASH" value="<?=$sha1hash?>">
<input type=hidden name="NAME" value="<?=$name?>">
<input type=hidden name="EMAIL" value="<?=$email?>">
<input type=hidden name="INSTRUCTOR" value="<?=$instructor?>">
<input type=hidden name="LOCATION" value="<?=$location?>">
<input type=hidden name="TIME" value="<?=$time?>">
<input type=hidden name="DATE" value="<?=$date?>">
<input type=hidden name="AUTO_SETTLE_FLAG" value="1">
<input type=hidden name="MERCHANT_RESPONSE_URL"
value="http://instructorbooker.com/response.php">

<?

```

The response.php class was created to handle to response back from Realex Payments API. This was also created through the use of the Realex Payments developer guides. Below is a snippet of code from the response.php class:

```

<?php

/*
Note:The below code is used to grab the fields Realex Payments POSTs back
to this script after a transaction has been processed. Realex Payments need
to know the full URL of this script in order to POST the data back to this
script.*/

$timestamp = $_POST['TIMESTAMP'];
$result = $_POST['RESULT'];
$orderid = $_POST['ORDER_ID'];
$message = $_POST['MESSAGE'];
$authcode = $_POST['AUTHCODE'];
$pasref = $_POST['PASREF'];
$realexsha1 = $_POST['SHA1HASH'];
$name = $_POST['NAME'];
$email = $_POST['EMAIL'];
$instructor = $_POST['INSTRUCTOR'];
$location = $_POST['LOCATION'];

```

```
$time = $_POST['TIME'];
$date = $_POST['DATE'];

// This is the account information which is set up with Realex Payments
$merchantid = "instructorbooker";
$secret = "*****";

// Below is the code for creating the digital signature using the SHA1 algorithm
provided by PHP
// This digital signature should correspond to the one Realex Payments POSTs
back to this script and can therefore be used to verify the message Realex sends
back.
$tmp = "$timestamp.$merchantid.$orderid.$result.$message.$pasref.$authcode";
$sha1hash = sha1($tmp);
$tmp = "$sha1hash.$secret";
$sha1hash = sha1($tmp);

//Check to see if hashes match or not
if ($sha1hash != $realexsha1) {
    echo "Hashes don't match - response not authenticated!";
}

?>

<body bgcolor="#FFFFFF">
<font face=verdana,helvetica,arial size=3>

<?php
// The transaction has been successful
if ($result == "00") {

    /*$to = $email;
    $subject = "Instructor Booker confirmation";
    $txt = $name .", \n\n Thank you for your booking, Your booking details
are: Instructor: $instructor, Time: $time, Location: $location, Date: $date";
    $headers = "From: dtk.ryan@gmail.com" . "\r\n".;

    mail($to,$subject,$txt,$headers);*/

?>
<!--Customer response message which appears once the transaction has completed -
->
Thank you, you payment has been processed successfully, you will now receive an
order confirmation email.
<br/><br/>
To return to InstructorBooker please <a
href="http://instructorbooker.com/index.php"><b><u>click here</u></b></a>
<br/><br/>

<?php
The transaction was not successful. You can ask the customer to amend their
details or try a different payment method
```

```
} elseif ($result == "101") {  
?>
```

The quiz.php class was created by inputting a number of quiz questions and answers into the Database. Once the page is loaded a query is sent to the DB to retrieve 10 random questions. A random generator was used to achieve this.

The roadtheory.php class is a content based class and contains no logical functions. This was developed using html and css.

The contact.php class also provides contact information and a contact form which posts to an email address in order to be reviewed.

I used XAMPP to test locally, XAMPP comes with local hosting and a MySQL database which I used before uploading the project to a live web server.

2.4.2 Front-End development:

The entire front end interface was created using Bootstrap. Bootstrap themes were incorporated to provide a responsive and dynamic application. The themes were then altered in order to suit the needs of the web application.

2.5 Graphical User Interface (GUI) Layout

Below is a rendition of the GUI for both the mobile and desktop website applications. Please refer to Appendix 6.4 to see how the GUI has evolved throughout the development.

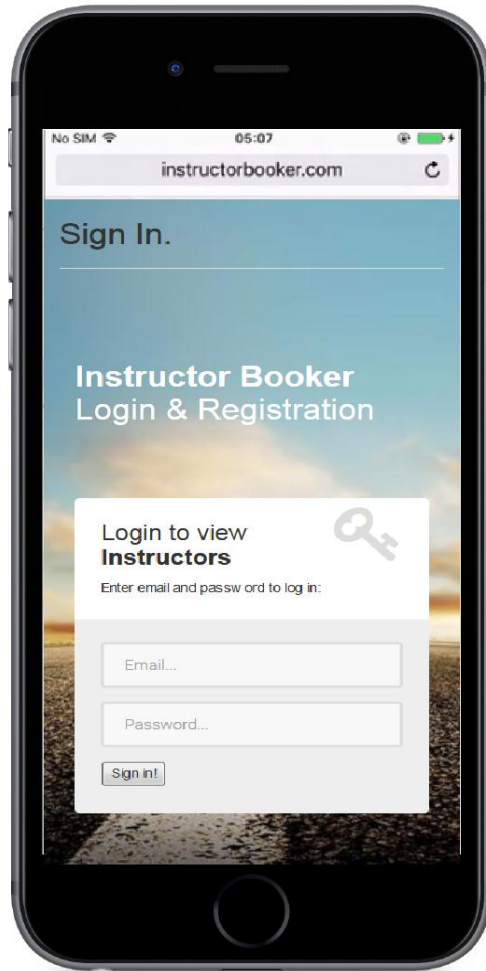
2.5.1 GUI - Mobile

Stage 1:



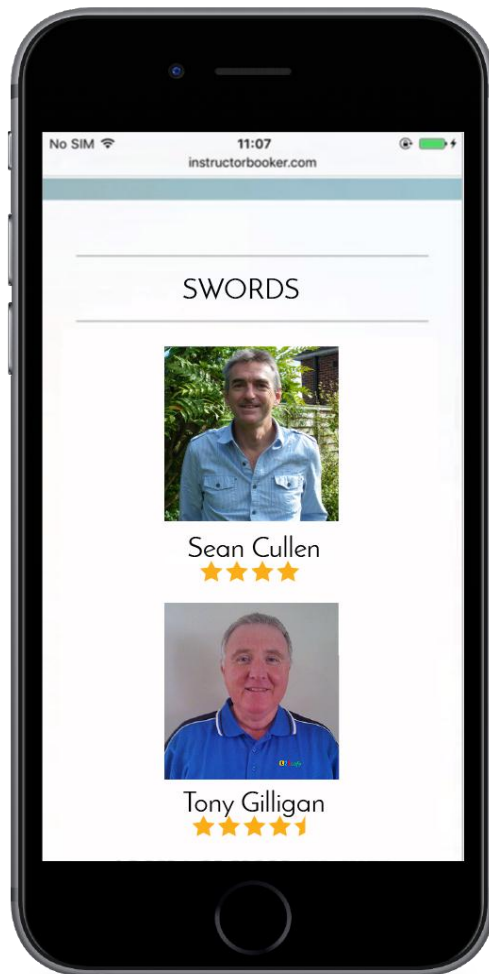
- The home page provides information regarding Instructor Booker

Stage 2:



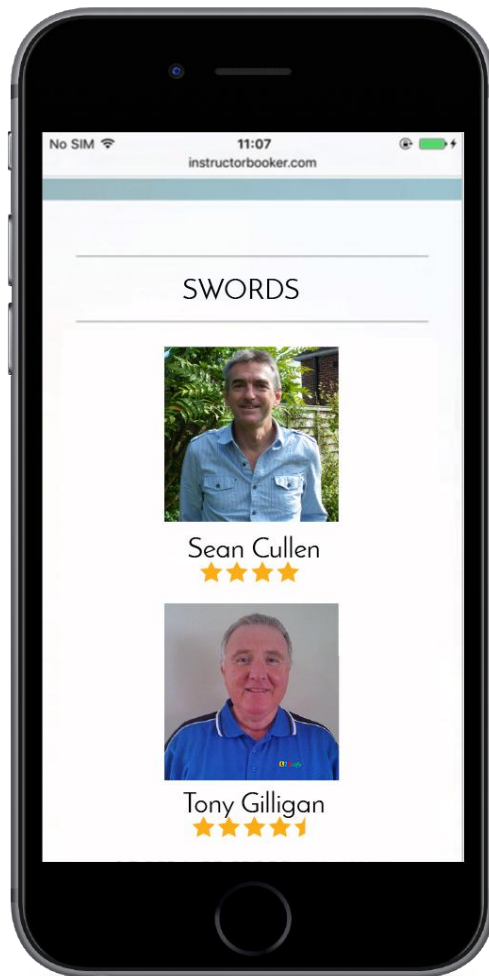
- Once the instructor section is clicked the user is redirected to a login / registration page.

Stage 3:



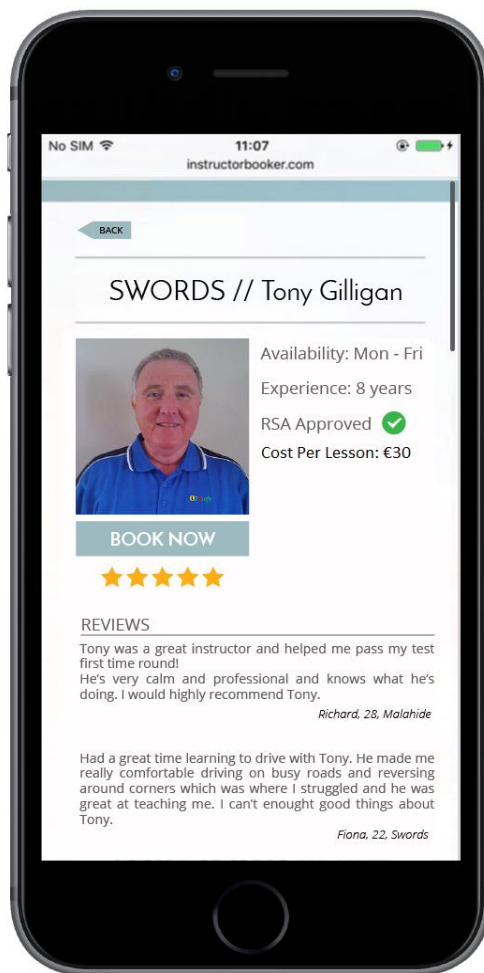
- The user can select the location. The database is queried and displays the drivers in the area.

Stage 4:



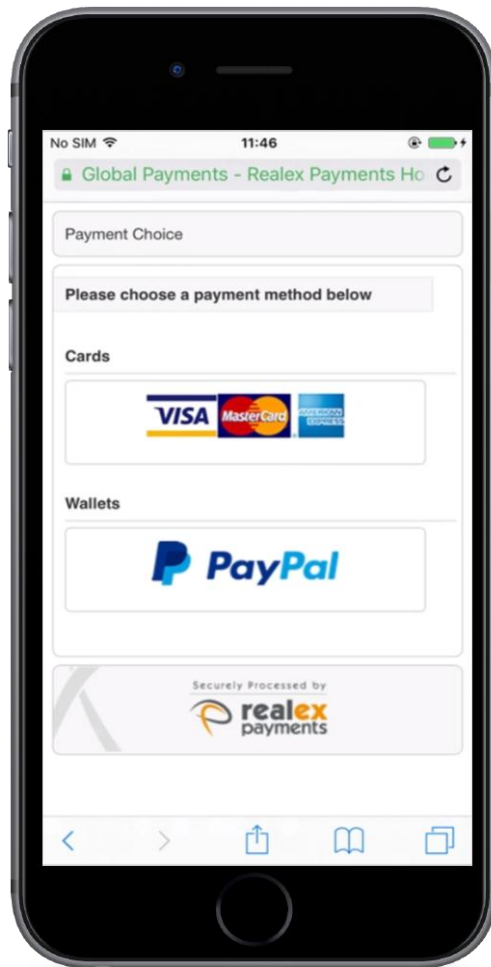
- The user can then select from the list of instructors in selected location. The user can see the review rating before selecting the driver.

Stage 5:



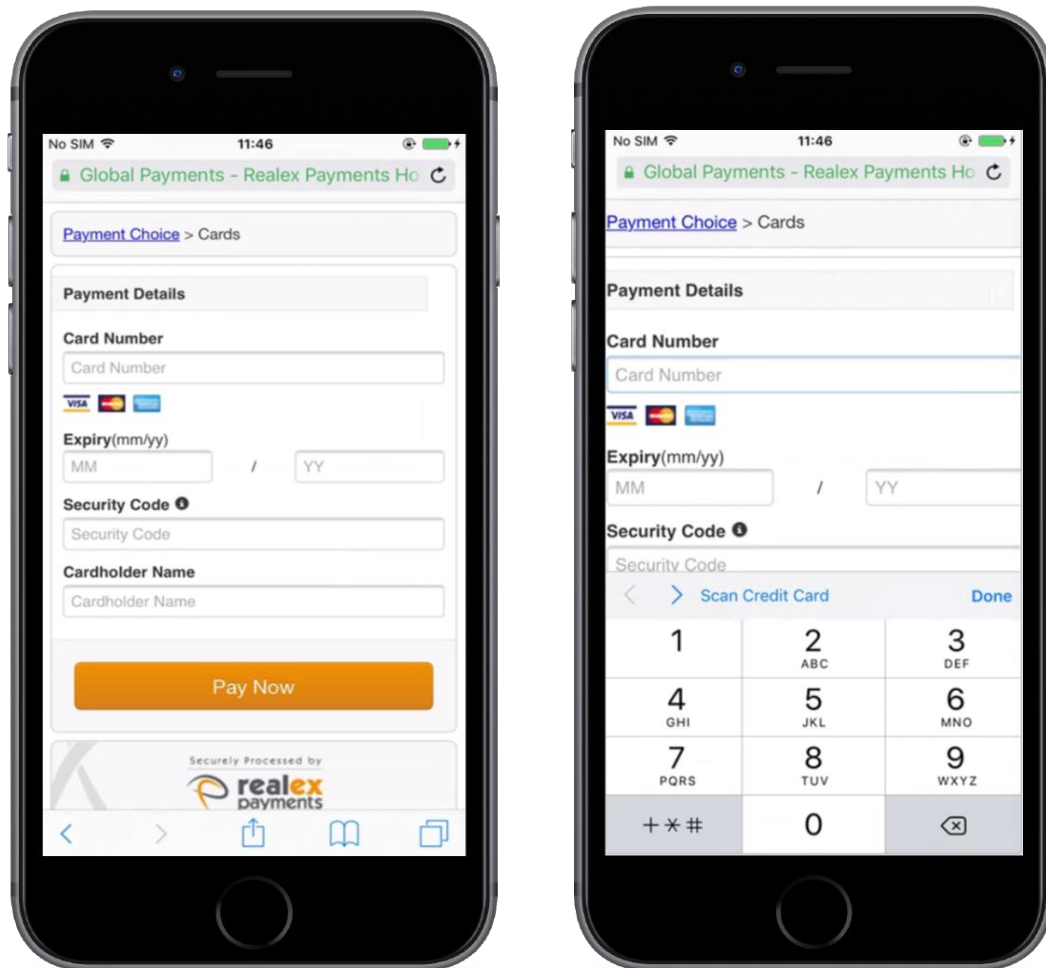
- Once selected the user has visibility of the instructors experience among other details. The user can the book via the “book now” button.

Stage 6:



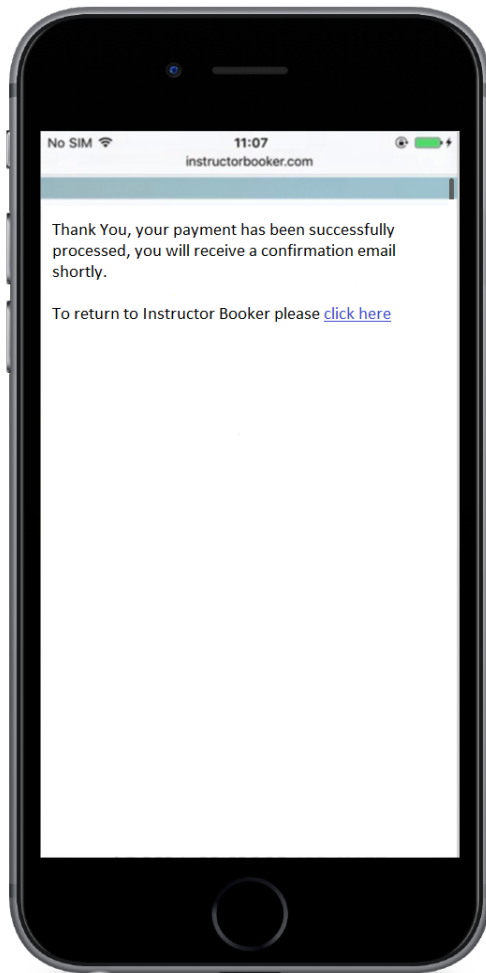
- The user will be redirected to the Realex Hosted Payment Page and can select to pay via PayPal or by Card.

Stage 7:



- If the card option is selected, the user will then enter their card information and the payment will be processed.

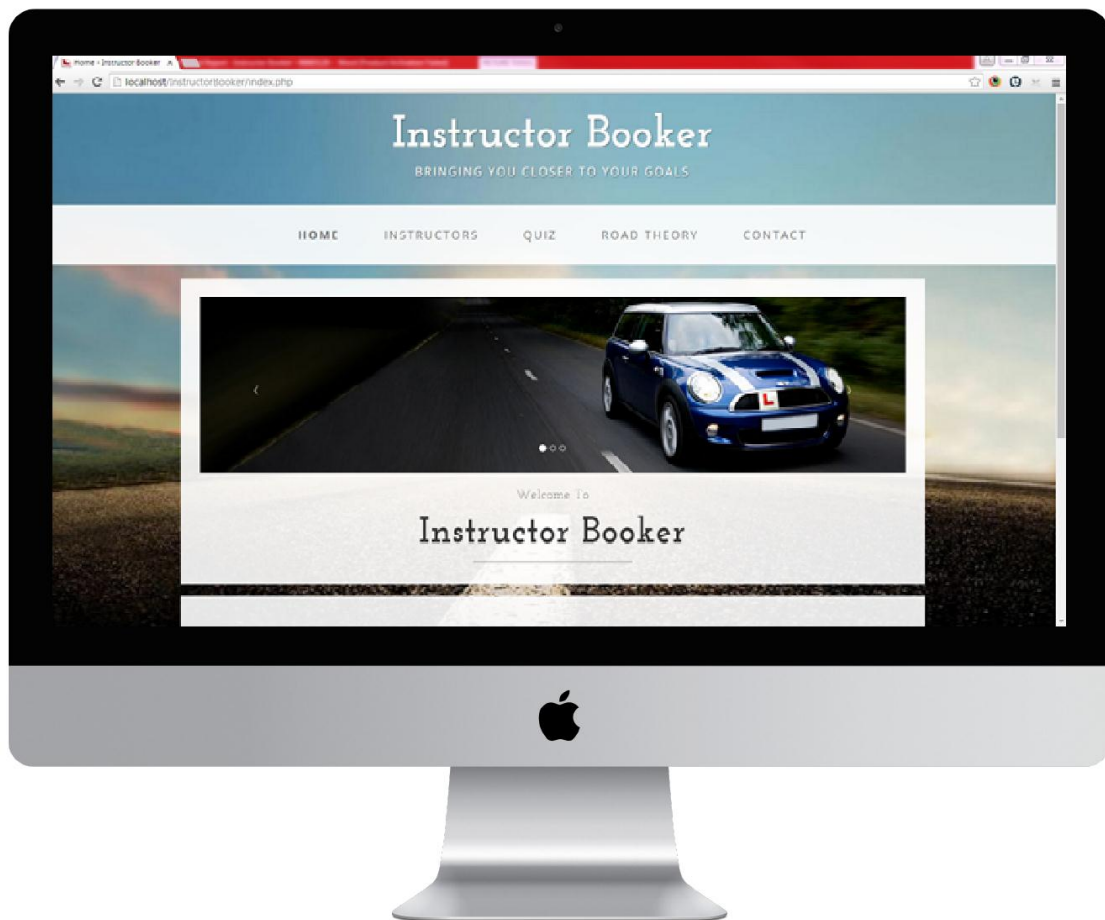
Stage 8:



- The customer will then receive a confirmation message stating if the transaction was successful or declined. An email with lesson details will be send via the response script.

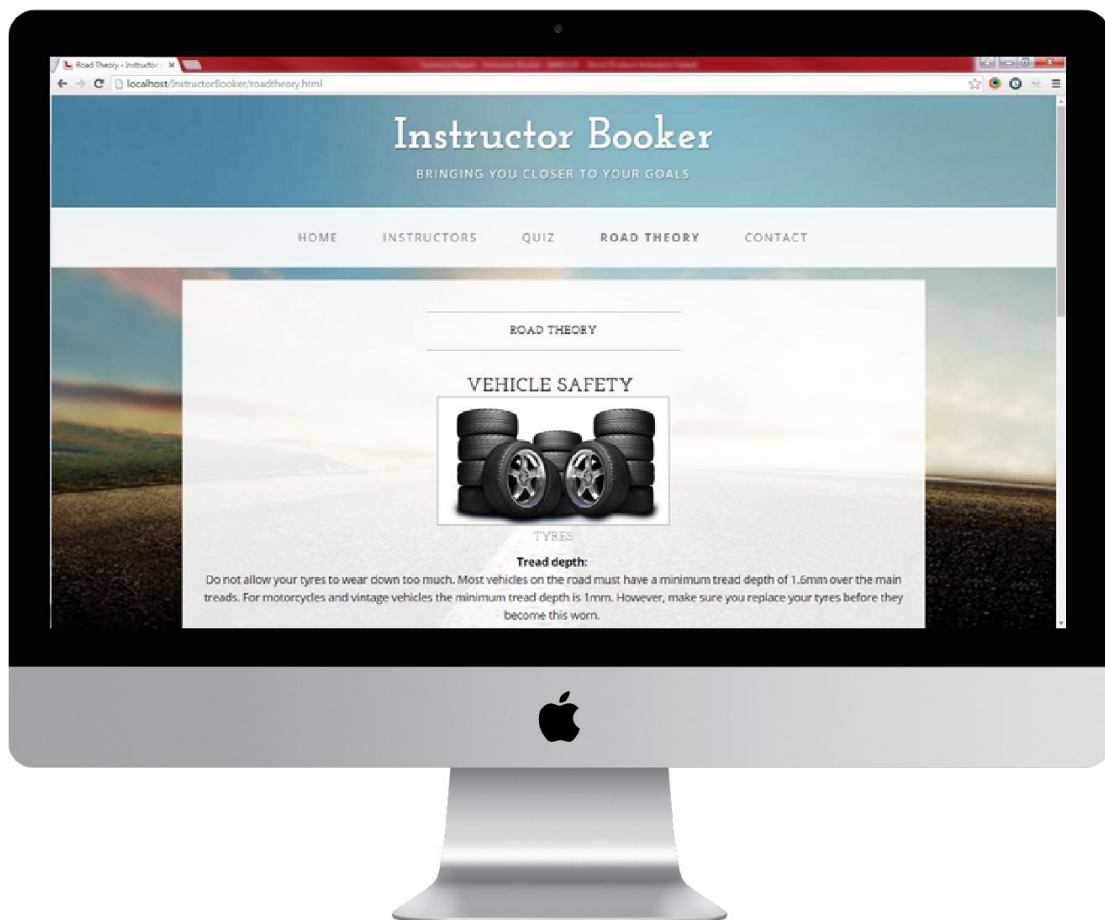
2.5.2 GUI- Desktop

Stage 1:



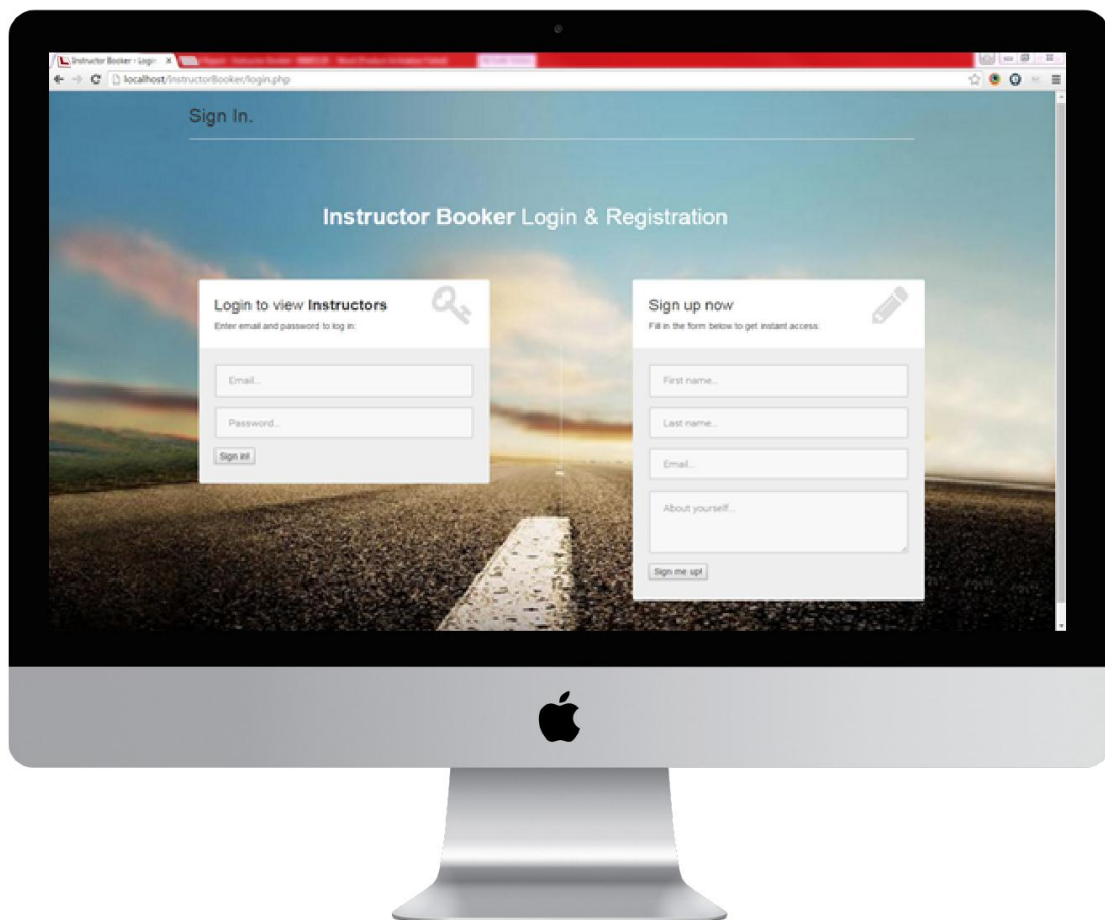
- The home page provides information regarding Instructor Booker

Stage 2:



- Road Theory – Users can improve their knowledge of road theory and then test their knowledge through the use of the quiz section.

Stage 3:



- Login – A user must first login or register in order to view the instructors section. This will stop unauthorised users accessing the instructor's details.

Stage 4:

The screenshot displays the 'INSTRUCTOR BOOKER' website interface. At the top, there's a navigation bar with 'HOME', 'ABOUT US', and 'LEAVE REVIEW'. A user profile 'TimMullins90' is visible in the top right. The main section is titled 'INSTRUCTOR PROFILE // JAMES FITZSIMONS' with a 'BOOK NOW' button. The profile includes a photo of James Fitzsimons, a list of qualifications (IB Approved Driving Instructor, RSA ADI Approved, and Fitzsimons School of Motoring voted as top choice by pupils 2015), and a bio in Latin placeholder text. To the right, there's a star rating of 4.5 / 5 Stars and a section for reviews, showing three identical positive reviews from 'Karen'. The footer contains copyright information for Instructor Booker 2016, contact details, and an association with the Road Safety Authority (RSA).

- Instructor Profile – Once an instructor has been selected the user can view their profile. The profile contains past reviews, qualifications and lesson price. Once happy with the instructor the user can proceed with arranging a lesson by clicking “book now”.

Stage 5:

The screenshot shows the 'INSTRUCTOR BOOKER' website. At the top, there's a logo and navigation links: HOME, ABOUT US, LEAVE REVIEW. A user profile 'TimMullins90' is visible in the top right. The main section is the 'INSTRUCTOR PROFILE // JAMES FITZSIMONS'. It includes a photo of James, his qualifications (IB Approved Driving Instructor, RSA ADI Approved, Fitzsimons School of Motoring voted as top choice by pupils 2015), and a 'BOOK NOW' button. A modal window titled 'SELECT DATE AND TIME OF LESSON' is open, showing a calendar for February 2016. The calendar has columns for SUN, MON, TUES, WEDS, THUR, FRI, and SAT. A legend indicates green for 'Available', red for 'Unavailable', and blue for 'Your Selection'. A 'NEXT' button is at the bottom of the modal. To the right of the modal, there's a star rating of 4.5 / 5 Stars and several reviews for James.

INSTRUCTOR BOOKER

TimMullins90

HOME ABOUT US LEAVE REVIEW

INSTRUCTOR PROFILE // JAMES FITZSIMONS **BOOK NOW**

SELECT DATE AND TIME OF LESSON

<<< FEBRUARY 2016 >>>

| SUN | MON | TUES | WEDS | THUR | FRI | SAT |
|-----|-----|------|------|------|-----|-----|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

Available Unavailable Your Selection

NEXT

4.5 / 5 Stars

Star ratings are based on customer and pupil feedback.

REVIEWS FOR JAMES

"James was an amazing driving instructor! He was really patient and explained everything along th way. His patience and knowledge really helped me feel comfortable behind the wheel and confident enough to take my driving test. And I passed!! Thanks so much James!" - Karen

"James was an amazing driving instructor! He was really patient and explained everything along th way. His patience and knowledge really helped me feel comfortable behind the wheel and confident enough to take my driving test. And I passed!! Thanks so much James!" - Karen

"James was an amazing driving instructor! He was really patient and explained everything along th way. His patience and knowledge really helped me feel comfortable behind the wheel and confident enough to take my driving test. And I passed!! Thanks so much James!" - Karen

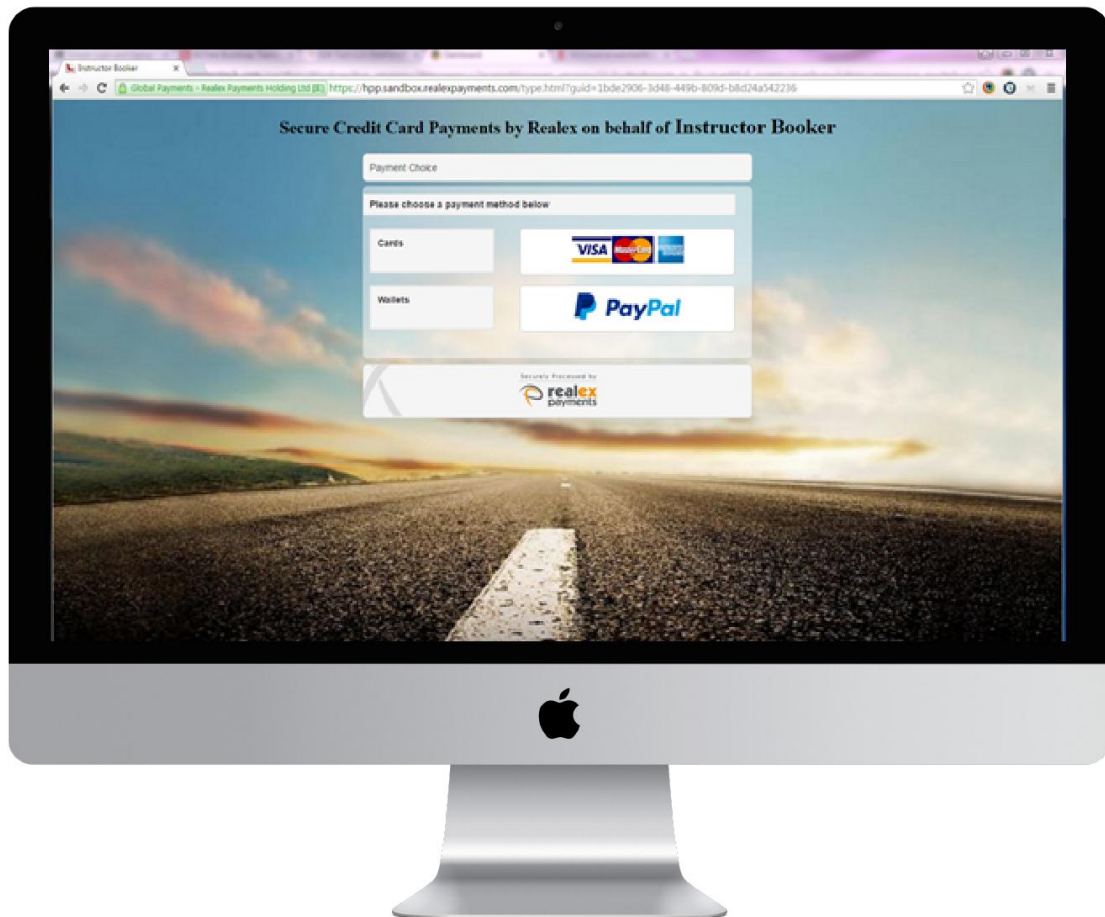
[Write Review](#) [Next](#)

© Instructor Booker 2016 1800 900 800 89 instructorbook@email.ie

In association with the Road Safety Authority RSA

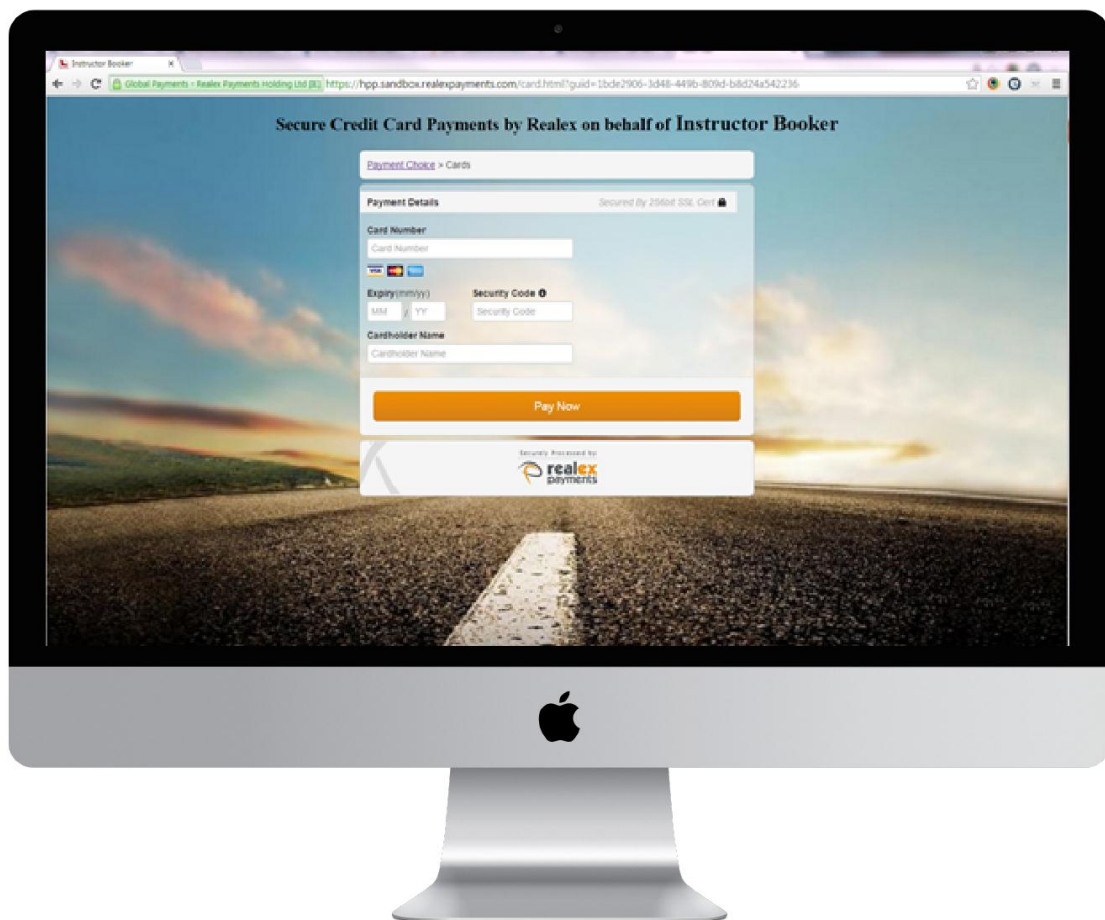
- **Availability** – The dates and times which are available are displayed and the user can select which is most suitable for them. Then book via the book now button.

Stage 6:



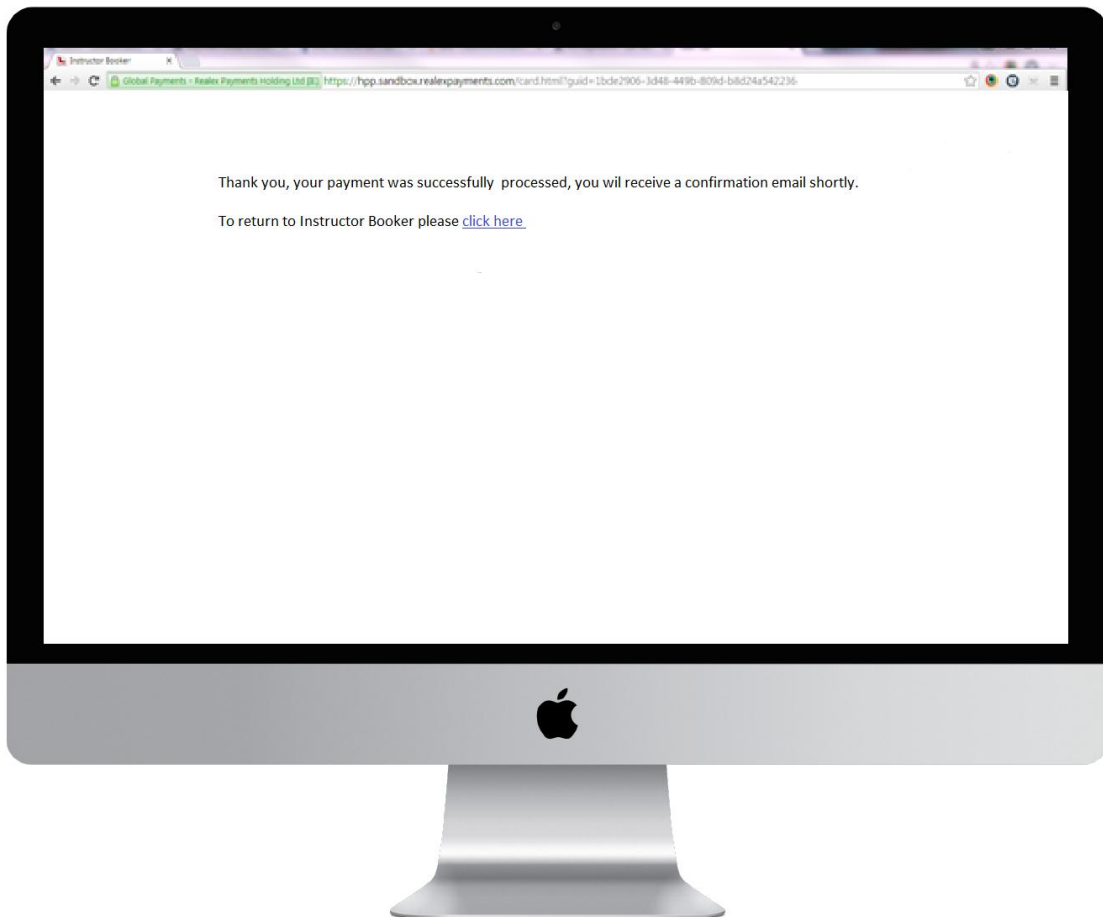
- Request - The user will be asked to confirm their preferred payment method.

Stage 7:



- Payment - The user is redirected to a secure server in order to enter their card details. Once “Pay Now” is clicked the transaction is authorised.

Stage 7:



- Response - Once a payment has been completed the user will receive a confirmation message and email notifying them. They can then return to the home page by clicking the link provided.

2.6 Customer Testing:

I arranged for a number of learner driver to review the application to determine if it suited their needs.

They provided some very useful feedback and suggested a number of improvements which I then built into the application. An example of a suggested improvements was to add a road theory section.

2.7 Testing

I identified the need for thorough testing to eliminate any potential bugs and to ensure the application performed as expected. Some of the testing methods used were as follows:

- **Unit Testing:** I ran manual testing on each function when it was developed to ensure it operated as required.
- **System Testing:** In effect this was white-box testing as I had full knowledge of the internal workings of the application.
- **Performance Testing:** I conducted an overall performance test to ensure the application delivered acceptable response times.
- **UAT Testing:** I arranged for a number of people who have software testing experience to perform UAT testing of the application. Their feedback was positive and constructive, they concluded was fit for purpose.

2.8 Evaluation

Having research the available technologies, I selected what I believe to be the most suitable ones in order to develop a scalable application. In particular using a Bootstrap front end framework allowed me to design a highly dynamic and interactive user interface. The use of PHP enabled me to provide logical functionality in the application and the PHP mySQLi functions used provided the database connections.

Throughout the development I kept in mind the need to have the application as user friendly and intuitive as possible.

Recognising the importance of comprehensive testing, I carried out a range of testing techniques as outlined in the Testing section above. This ensured that the application is robust and stable.

3 Conclusion

I found this project challenging but really enjoyed the process. It helped to enhance my programming and analytical skills. It gave me the opportunity to learn about and use a lot of new technologies which will be of benefit to me in future projects.

4 Further development or research

This system can be used by any business which is looking to avail of a booking system operated through a simple and smart application which provides a review option.

The application is currently based around driving instructors but could be used for any type of instructor booking e.g. Tennis, Personal Trainer etc.

The framework is interchangeable and is not restricted to a particular design.

5 Bibliography

Stack overflow (2016) Available at: <http://stackoverflow.com/> (Accessed: 10 December 2015).

Majid, A. (2015) Creating registration form with PHP and MySQLi. Available at: <http://www.eggslab.net/creating-registration-form-with-php-and-mysqli/> (Accessed: 10 January 2016).

W3Schools online web tutorials (no date) Available at: <http://www.w3schools.com/> (Accessed: 14 January 2016).

Bootstrap, S. (no date) Free Bootstrap themes and Templates. Available at: <http://startbootstrap.com/> (Accessed: 12 February 2016).

Realex Payments (no date) Available at: <https://developer.realexpayments.com/> (Accessed: 20 April 2016).

6 Appendix

6.1 Project Proposal

My project idea is to easily connect learner drivers with driving instructor via a web application. The application will have instructor profiles, company/contact details, locations, merits and qualifications of the driving instructors which the user can browse when looking for a lesson. The user can also book a test with an instructor, leave a review, comment and star-rate and take driving theory quizzes among other activities. The main goals of this app is to enable learner drivers to have a wide choice of driving instructors who's experience and ability are transparent.

My aim is to create a web app which is user friendly and provides all the necessary information required for both the driver and instructor. I hope to implement a payment method using the payment gateway Stripe so customers can actually pay for their lessons. This will turn the web app into a more holistic product with end to end functionality. Using a review process is what I feel will attract the most attention; currently learner drivers have to take a risk of perhaps paying an instructor who is not at the level they require. The review features will hopefully element this issue.

6.2 Background:

From my research that this type of service is few and far between, feel there could be a large market within reach. Currently if a learner is looking to book a driving instructor they must monotonously search through directories in their area. Alternatively a lot of bookings come through word of mouth which is not really sustainable. There are some companies such as Irish School of Motoring (ISM) who can schedule appointments via a website. This does not however facilitate private instructors who may not have the expertise or funding to develop a site themselves.

I expect learner drivers looking to pass their test to use this app, as 12 lessons are now mandatory for all new drivers this app will appeal to them. On the other hand, the driving instructors could also be classed as a benefiter of this product as they would also be a user of the service. They would have their own profile on the app which would, in a sense, act as a business card for their business. They would also be able to reach a larger number of potential pupils they would not have been able to reach beforehand.

6.3 Technical Approach:

The entire project is going to be written in PHP and will communicate with a database using SQL commands. The web app will be hosted on an apache web server. I will start by creating a basic site and connect it to a MySQL database, I will create input fields which write to the database and once the database has been filled I will start on the UI. The IDE being used will be a version of eclipse which is designed for PHP projects. My aim is to be able to post to strips API in order to process the bookings. I will be using sha1/md5 hashing algorithms in order to encrypt sensitive data. Card details will not be taken on the server as I wish to avoid PCI compliance charges. I hope to connect to google maps API in order to dynamically display the available drivers in the area. I will split the tasks up into smaller realistic targets in order to maintain an organized and manageable project.

6.4 Project Plan:

| Project Milestones | | | | |
|--------------------|------------------------|--|--------------------|-------------------|
| Project | Instructor Booker | | | |
| ID | Milestone | Description | Planned Completion | Actual Completion |
| 1 | Basic web Application | Create the basic functions of the web app | 01/11/2015 | 15/01/2016 |
| 2 | Technical Requirements | Technical specification upload | 06/11/2015 | 06/11/2015 |
| 3 | Database | Build a MySQL database | 15/11/2015 | 10/01/2016 |
| 4 | Research | Obtain driving instructors details | 15/11/2015 | TBC |
| 5 | Functionality | Create communication to the database | 15/12/2015 | 18/01/2016 |
| 6 | Algorithms | Implement hashing algorithms for transaction requests. | 10/1/2016 | 28/01/2016 |
| 7 | User interface | Develop the UI using php and html | 01/02/2016 | 15/04/2016 |
| 8 | Prototype | Have working prototype | 05/02/2016 | 03/02/2016 |
| 9 | Database Completion | Required content filled | 25/02/2016 | 26/04/2016 |
| 10 | Functionality | API calls to google maps | 10/03/2016 | Not Completed |
| 11 | Payments | Payment gateway integration | 30/03/2016 | 03/02/2016 |
| 12 | User interface | Finalisation of UI | 10/04/2016 | 03/05/2016 |
| 13 | UAT | User acceptance testing | 15/04/2016 | 05/05/2016 |
| 14 | Evaluation | Ensure all processes are functioning | 20/04/2016 | 08/05/2016 |