National College of Ireland
BSc (HONS) in Computing
2016/2017

Conor Prunty
13102311
conor.prunty@student.ncirl.ie

Online Booking System

Technical Report

Table of Contents

Executive Summary 4

1 Introduction 5

1.1 Background 5

1.2 Aims 6

1.3 Technologies 6

2 System 9

2.1 Requirements 9

2.2 Requirement Gathering 9

2.3 Functional Requirements 13

2.3.1 Use Case Diagram 13

2.3.2 Requirement 1: Admin user registration 14

2.3.3 Requirement 2: Admin to log in 15

2.3.4 Requirement 3: Setup of admin's specific features 17

2.3.5 Requirement 4: Customer to submit a booking request 18

2.3.6 Requirement 5: Admins access rights to the bookings 20

2.3.7 Requirement 6: Admin edits one of their setup areas 21

2.3.8 Requirement 7: Admin user customises the page 23

2.3.9 Requirement 8: Customer cancels their own booking 24

2.3.10 Requirement 9: Admin user edits their own details 26

2.3.11 Requirement 10: Customer shares booking 27

2.4 Non-Functional Requirements 28

2.4.1 Usability requirement 28

2.4.2 Reliability requirement 29

2.4.3 Performance requirement 29

2.4.4 Supportability requirement 29

2.4.5 Security requirement 29

2.4.6 Interface requirement 29

2.4.7 Operation requirement 29

2.5 Design and Architecture 30

2.6 Implementation 31

2.7 Graphical User Interface (GUI) Layout 33

2.7.1 Admin Registration 33

2.7.2 Setup of specific features 34

2.7.3 Booking Request 35
2.7.4 Admin Page 36
2.8 Admin Testing 37
2.8.1 Registration 37
2.8.2 Login 43
2.8.3 Navigation Bar 47
2.8.4 Setup 53
2.8.5 Update Bookings 62
2.8.6 Delete Bookings 71
2.8.7 Update Information 72
2.8.8 Add Banner 75
2.8.9 View All Bookings 77
2.9 Customer testing 81
2.9.1 Homepage 81
2.9.2 Navigation Bar 83
2.9.3 New Booking 85
2.9.4 Cancel Booking 98
2.9.5 Contact Form 100
2.10 Evaluation 104
3 Conclusions 108
4 References 109
5 Appendix 110
5.1 Declaration 110
5.2 Project Proposal 111
5.3 Project Plan 112
5.4 Monthly Journals 113
5.4.1 September 113
5.4.2 October 117
5.4.3 November 120
5.4.4 December 123
5.4.5 January 126
5.4.6 February 127
5.4.7 March 129

Executive Summary

The project is an online booking system, designed for use with any sports clubs who have facilities for hire within their own club. This is an area that is not currently widely available, in Ireland at least, and I feel that this project could prove very beneficial for any future customers. Most clubs currently use forms online, or have staff answering telephone calls, and I feel this application could save the club time and money by all of it being done online, and quicker than with human intervention. The online system has a simple form UI with the available options for the user to book their specified time and area, with immediate confirmation being sent once booked.

The design is in the form of a website, with all the necessary features as explained in this document. The main features are the admin section which can only be accessed by an admin, and the booking page, which can be accessed by the general public, once they have knowledge of the correct URL. The information is stored in a database, and gets updated and retrieved to be displayed to the customer, upon using some of the different functions available. Validation is in place in the backend to stop timeslots being double booked, and also on the initial setup to ensure at least a minimum of one area and time are selected.

Overall, the application works well and as intended. The functionality is generally seamless, and provides a nice easy-to-use interface, for both the admin user and the external customer.

1 Introduction

This project is an online booking system available to all sports clubs with facilities to hire. Once set up, it can be used by members of the public to book areas made available by the club's admin.

1.1 Background

This project was chosen as the number of online booking systems for sports clubs around Dublin is fairly minimal. The majority of them just have a landline number to ring, an email address to be used or a form to fill in online, but that doesn't give an immediate response.

From past experience of working in a GAA club with a function room to rent, I noticed that the only option was a landline to ring in order to book it. This phone was located in a small room at the end of the bar. However, the bar is closed during the day and only opens in the evening time on weekdays, but on weekends it can open earlier. If the bar was busy, the staff would not get a chance to answer the phone, or if it was a quiet day, there might only be one member of staff working and he or she may not have the chance to answer. Also, if they answer, they may not know immediately if the room is free on a particular date, or the cost, and have to ask you to either phone back, or call in to the bar in person.

I researched several other GAA clubs online and found that none of them have an online booking system. I had a look at several different sites, including Naomh Barrog, Naomh Mearnog, St Sylvesters, St Vincents, Raheny, St Peregrines, St Brigids, Erins Isle, O Tooles, Clontarf and UCD. These clubs have a variety of potential rooms or pitches to book, but all without an online system for doing so.

I also researched sports & leisure clubs, where some are specifically designed to just have pitches or function rooms to rent. The majority did not have an online booking system, such as Astro Park, Trinity Sports & Leisure, SportsCo, Clontarf Road Astro Pitches and Belvedere College.

However, I did come across some other clubs with a different outcome. ALSAA had a link for an online booking system, which originally said it was currently unavailable. This site has since been updated, and they now have an online booking form, with the option to pay also. The other site with a difference was the AUL Complex. It does have a fully functioning online booking system for their pitches, but you currently must be a member. Although this site is already online with the booking functionality, I will use it to my advantage to see what is out there and get an idea of how my system should work.

In this day and age, people want things instantly, and with as little effort as possible. As everything is basically online, people can be impatient and I feel would book with a particular club if they knew they'd be able to see what is available and book there and then and receive an immediate response.

1.2 Aims

This project aims to address a gap in the current market, as described in the background section above. The idea is to make a user-friendly, easy-to-use booking system, so it can be purchased and used by many people, at home and abroad if necessary.

Demonstrations and test runs can be given to future customers to see if it is a success before they have to pay for it. This is a good way to get people to use it, as it's initially free, and if it is a success, I estimate that a high percentage would be willing to pay for it permanently, if the price was right.

1.3 Technologies

HTML - Hypertext Mark-up Language: a standardized system for creating the structure of web pages. HTML is used here, along with Bootstrap, for the general structure of each web page and how the design of the page is implemented.

CSS - Cascading Style Sheets: used to style HTML elements by applying custom presentation functions. CSS is used to 'beautify' the pages, and improve the look-and-feel of the site.

Bootstrap - Bootstrap is a HTML, CSS, and JS (JavaScript) framework for creating the layout of web pages. Bootstrap contributes massively, along with HTML, in the overall design of the pages.

PHP - Hypertext Pre-processor is a widely used scripting language. PHP is used everywhere throughout the site, largely for combining the pages to the database.

MySQL – This is a relational database system. MySQL is used to gather the required information from the database and pass it back to the desired pages.

JavaScript – JavaScript is a programming language; one of three core technologies of the World Wide Web. JavaScript is used for creating functions on the site, such as the randomly generated booking reference number.

jQuery – a JavaScript library used to simplify the use of JavaScript. This is used on a number of occasions across the site, such as the setup page where the user can quickly select/de-select all the timeslots presented.

The system itself will have its general layout using the bootstrap framework, tied in with HTML & CSS to display the content required. Proper layout makes the look of the objectives more appealing to the customers (Jetpeppers.com, 2017). This will aid the customer's experience from both the club admin, and the outside customer who is using the booking system. The code behind the booking system will be written using languages such as JavaScript and PHP. These are well-known languages and will allow the system to be designed with all the necessary functionality, such as sending a notification email to the customer to advise their booking is complete.

The information will need to be stored somewhere in order to display to the user. For this, I will have this saved in a very common database, MySQL. In order to submit and extract information from the database, SQL statements will be used.

For example, setting up the details of the admin user, or sending a request to the database to see if a timeslot has been previously booked. It will also display functions such as: a particular slot is already booked, or to call some of the user's information to notify them when a booking has been made.

The initial set up will be written in a server-side language; PHP. This will allow the admin of the club to log in to allow them to set up their customary options for the system. This will be written in a secure manner to prevent potential hacks, by using functionality such as hashing the password so if a malicious user gained access to the database, they still wouldn't be able to get any of the passwords.

2 System

2.1 Requirements

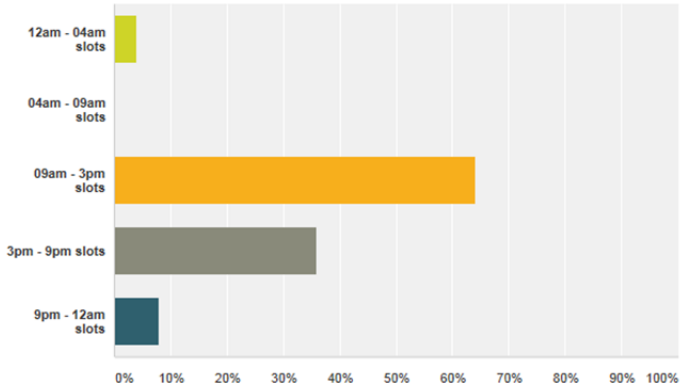
The system will be hosted online, and as such will require all users to be using a device with an active network connection and the ability to connect and browse the Internet. The admin user has the access rights to view, create or delete bookings, should they choose. The end-user will only have the access rights to make a booking or cancel their own booking.

2.2 Requirement Gathering

I used an online survey to get further information on how the external customer would like certain parts of the application. I just posed five questions, and left the survey open until I had 25 responses. The return on investment for good requirements gathering is virtually always higher than the cost (Perrigon, 2017). This helps ensure the customer will get what they actually want.

Q1: When an admin user is setting up their area, which timeslots should be selected by default?

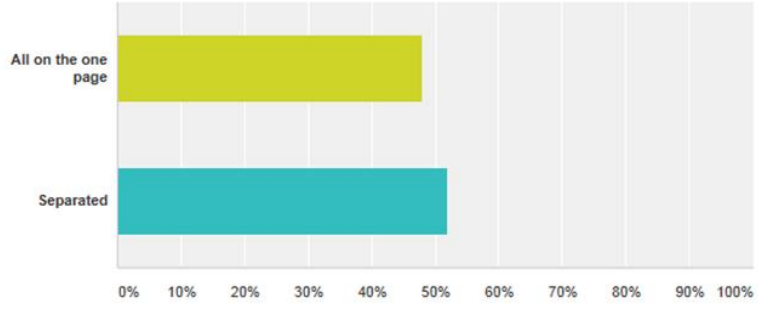
Result: Two timeslots were clear winners – 09am – 03pm and 03pm – 09pm. I have set this up within the application, and also added a select all button for both the early timeslots, and the late ones. This allows quick selection for the admin user.



Answer Choices	Responses	
12am - 04am slots	4.00%	1
04am - 09am slots	0.00%	0
09am - 3pm slots	64.00%	16
3pm - 9pm slots	36.00%	9
9pm - 12am slots	8.00%	2
Total Respondents: 25		

Q2: As a customer, would you like all the information on the one page when making a booking, or separate it out into different sections?

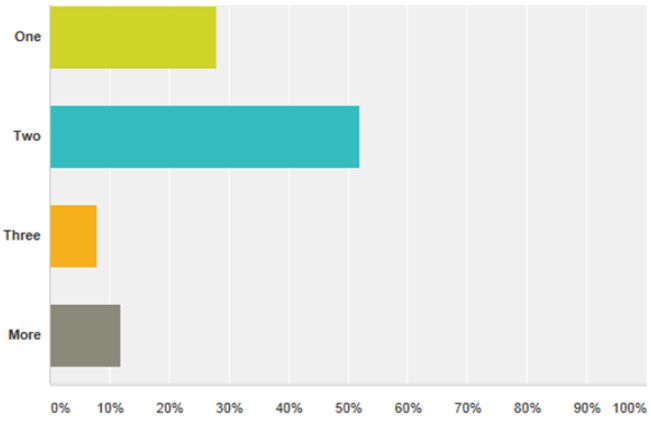
Result: This was a close call, with a 52% - 48% swing in favour of having them separated. It is probably easier as the developer to put them all on the one page; however, this way makes the code a bit cleaner and easier to read for other people.



Answer Choices	Responses	
All on the one page	48.00%	12
Separated	52.00%	13
Total		25

Q3: How many admin users should be allowed per club?

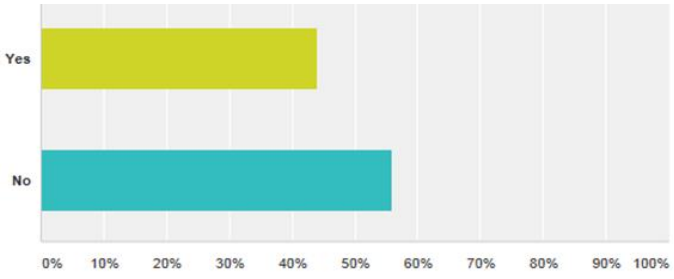
Result: This also had a clear winner; two. I have coded the application so that a maximum of two admin users can be registered per club at any one time.



Answer Choices	Responses	
One	28.00%	7
Two	52.00%	13
Three	8.00%	2
More	12.00%	3
Total		25

Q4: Should the admin be able to change how the site looks, i.e. change the colour scheme?

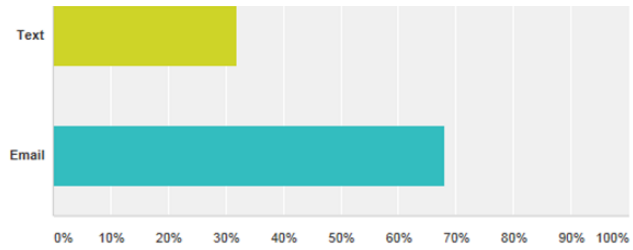
Result: Whilst the 'No' vote won this question, I have allowed the admin user to add their own banner to the site, due to further feedback I received.



Answer Choices	Responses	
Yes	44.00%	11
No	56.00%	14
Total		25

Q5: As a customer, would you prefer an email or text message when receiving confirmation of a booking?

Result: I was surprised by this result; email won by more than double over text message. I asked some of the participants about this question and was told they would much rather give an application an email address instead of their mobile phone number, which does make sense.

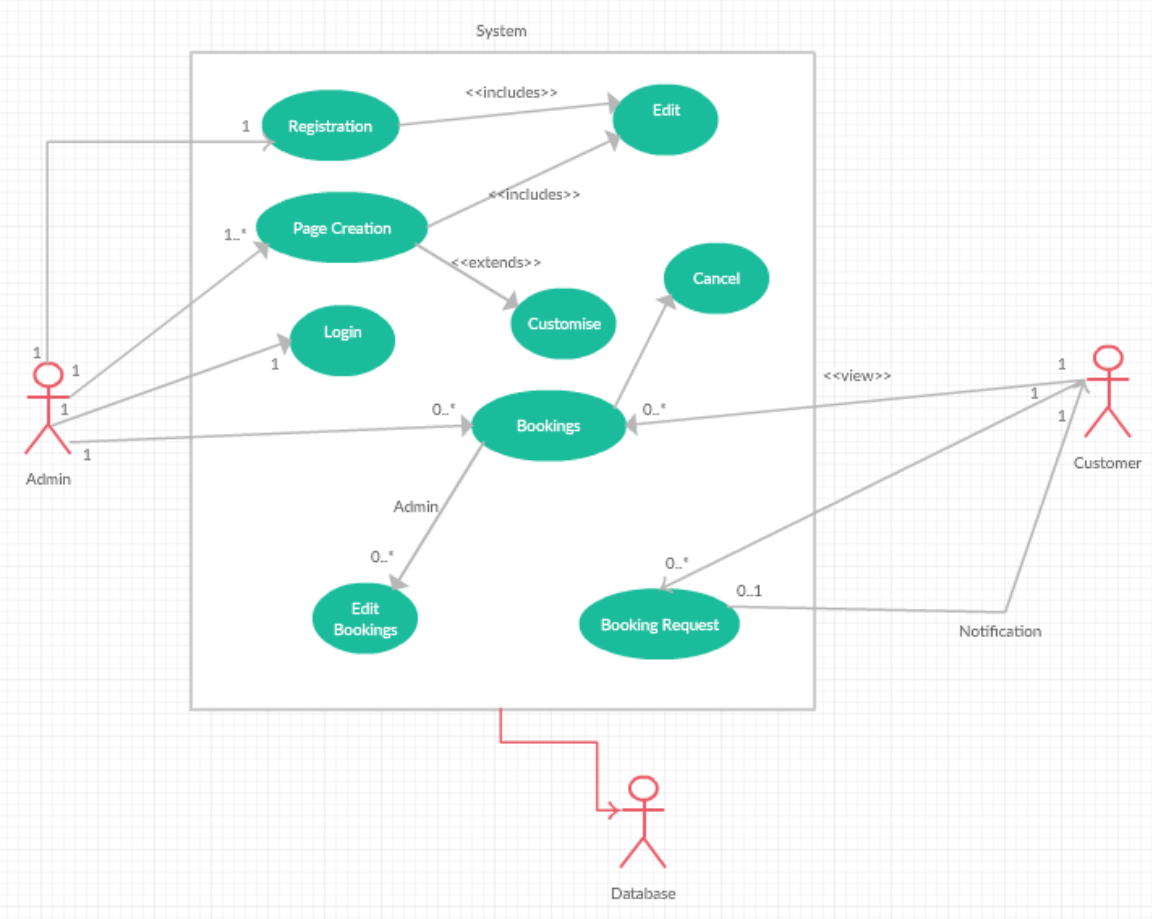


Answer Choices	Responses	
Text	32.00%	8
Email	68.00%	17
Total		25

2.3 Functional Requirements

- Admin user registration
- Admin to log in
- Setup of admin's specific features
- Customer to submit a booking request
- Admins access rights to the bookings
- Admin edits one of their setup areas
- Admin user customises the page
- Customer cancels their own booking
- Admin user edits their own details
- Customer shares booking

2.3.1 Use Case Diagram



2.3.2 Requirement 1: Admin user registration

2.3.2.1 Description & Priority

The initial set up of the club's admin user. This is the highest priority as without admins, there wouldn't be a system. The admin user's registration is the first step in a club getting the system up and running.

2.3.2.2 Use Case

Scope

The scope of this use case is to set up a new admin user from scratch for a specified club.

Description

This use case describes the process how an admin user creates their account.

Use Case Diagram

Admin → Registration

Flow Description

Precondition

Registration page presented to the club admin.

Activation

This use case starts when the admin navigates to the registration page within a browser.

Main flow

1. The admin user creates a username and password, and provides a valid email address in the online form.
2. The system adds the user to the database, with an encrypted password.
3. The user gets a notification email to notify of successful registration.

Alternate flow

1. The admin user enters an email address that's already registered.
2. The system responds by notifying the user on the web page that the email address is already in use.
3. The admin user must continue at the start of the main flow.

Exceptional flow

1. The admin user enters an email address in an invalid format.
2. The system responds asking them to re-enter a valid email address.
3. The user must continue at the start of the main flow.

Termination

The user has completed registration.

Post condition

The system returns to the homepage.

2.3.3 Requirement 2: Admin to log in

2.3.3.1 Description & Priority

Logging in is a simple function, yet vital. This is required for the admin user to access the secure part of the system in order to allow them to do all the setup and workings of the system, from an admin perspective.

2.3.3.2 Use Case

Scope

The scope of this use case is to allow the admin user gain access to the system.

Description

This use case describes the process how an admin user can log in with their credentials.

Use Case Diagram

Admin → Login

Flow Description

Precondition

An admin user has been successfully registered.

Activation

This use case starts when the admin navigates to the login page within a browser.

Main flow

1. The admin user enters their valid credentials from when they registered.
2. The system checks the database and returns a successful response.
3. The user is presented with the admin page.

Alternate flow

1. The admin user enters invalid credentials.
2. The system checks the database.
3. An error message is presented to the user and they must begin at the start of the main flow.

Termination

The user has successfully logged in.

Post condition

The system displays the admin page.

2.3.4 Requirement 3: Setup of admin's specific features

2.3.4.1 Description & Priority

This is where the admin user sets their specific features. They must choose what areas they have available for rent, the timeslots available, the days of the week required and the cost of the rent per hour. This is also a high priority as the club need to define their own available features and times.

2.3.4.2 Use Case

Scope

The scope of this use case is to set up the features required by a club admin.

Description

This use case describes the process of setting up the specific features required by a particular club.

Use Case Diagram

Admin → Page Creation

Flow Description

Precondition

The admin user has successfully registered.

Activation

This use case starts when the admin user logs in to the system.

Main flow

1. The system presents the user with a dropdown menu of all the available areas.
2. The user selects the area they require and clicks submit.
3. The user then needs to select the days, times and cost of the area.
4. The system updates the database with the user's selections.
5. The system returns the user's selections in the form of a table.

Alternate flow

1. The user does not see their area of choice in the full dropdown menu.
2. The user adds in the area in the text box and clicks submit.
3. The user chooses the new area from the dropdown menu.
4. The system continues at point 3 of the main flow.

Termination

The user completes the set up and the system has created the booking page.

Post condition

The system displays the booking page to the user.

2.3.5 Requirement 4: Customer to submit a booking request

2.3.5.1 Description & Priority

The main use of the system; a customer uses the functionality to book an available section online and gets an immediate response. This isn't absolutely crucial, as the system is up and running at this stage, but this use case is the purpose of the system.

2.3.5.2 Use Case

Scope

The scope of this use case is how a customer submits a booking request.

Description

This use case describes the process for a customer to request a booking of an available section.

Use Case Diagram

Customer → Booking Request

Flow Description

Precondition

The admin user has set up their specific page.

Activation

This use case starts when the customer navigates to the booking page.

Main flow

1. The system presents the available areas for rent.
2. The customer chooses their area of choice.
3. The system displays the available timeslots.
4. The customer clicks to make a booking.
5. The customer chooses the day required.
6. The customer chooses the time required.
7. The system confirms in a form the area, day and time entered.
8. The customer enters their name and email address and clicks submit.
9. The system sends an email with the details and a booking reference number to the customer.

Alternate flow

1. (At point 5 of the main flow) The customer changes their mind about the area they chose.
2. The customer goes back to the last page.
3. The customer chooses the option to select a different area.
4. The system continues at point 2 of the main flow.

Exceptional flow

1. (At point 8 of the main flow) The customer does not enter an email address.
2. The system responds with a message to the user to enter a valid email address.
3. The customer continues at point 8 of the main flow.

Termination

The system has accepted the booking and informed the user via a pop-up message, and an email has been sent.

Post condition

The system displays the booking complete page.

2.3.6 Requirement 5: Admins access rights to the bookings

2.3.6.1 Description & Priority

The admin user has access rights to view all bookings and delete bookings if required. This is a medium priority as the admin user may not need to use their access rights too often.

2.3.6.2 Use Case

Scope

The scope of this use case is to ensure the admin user has the ability to view all bookings or delete them.

Description

This use case describes the process of how an admin user would view the bookings or delete them.

Use Case Diagram

Admin → Bookings → Edit Bookings

Flow Description

Precondition

A customer has submitted a booking request.

Activation

This use case starts when the admin user logs in to the system.

Main flow

1. The admin user navigates to the ‘all bookings’ page.
2. The system displays all current bookings.
3. The admin user enters any information in the search bar.
4. The system presents only those options in the table with what was searched.
5. The admin user chooses a particular booking and clicks the trash symbol.
6. The system confirms whether they wish to cancel or not.
7. The admin user selects ‘yes’.
8. The system refreshes the list of bookings and the deleted one is no longer there.

Alternate flow

1. (At point 7 of the main flow) The admin user selects ‘no’.
2. The system returns to the booking page and no change has occurred.

Termination

The admin user navigates away from the booking requests page.

Post condition

The system displays the page that the user chose.

2.3.7 Requirement 6: Admin edits one of their setup areas

2.3.7.1 Description & Priority

The admin user can amend their initial setup, by adding or removing times or days. This is a medium priority as it’s vital to have, but may not always be used.

2.3.7.2 Use Case

Scope

The scope of this use case is to ensure the admin user has the ability to edit their areas as required.

Description

This use case describes the process of how an admin user would update one of their areas.

Use Case Diagram

Admin → Page Creation → Edit

Flow Description

Precondition

An area has already been set up.

Activation

The use case starts when the admin user navigates to the update bookings page.

Main flow

1. The user chooses the area to be updated.
2. The system displays the full table for that area.
3. The user chooses to add/remove a day/time from the dropdown menu.
4. The system presents the days/times available to be added/removed.
5. The user chooses the option required.
6. The system re-displays the table with the change included.

Alternate flow

1. (At point 5 of the main flow) The user decides not to choose an option and returns to the homepage.
2. The user then re-navigates to the update page.
3. The system displays no changes have been made.

Termination

The admin user successfully updates their area of choice.

Post condition

The system displays the booking page.

2.3.8 Requirement 7: Admin user customises the page

2.3.8.1 Description & Priority

The admin users can customise the page by adding their own banner to the top of each page. This is a medium priority as it's more a nice-to-have than a necessity.

2.3.8.2 Use Case

Scope

The scope of this use case is to ensure the admin user can customise the page by adding a banner.

Description

This use case describes the process of how an admin user would add or remove a banner.

Use Case Diagram

Admin → Page Creation → Customise

Flow Description

Precondition

An admin user has logged in.

Activation

The use case starts when the admin user navigates to the add banner page.

Main flow

1. The user selects an option from the dropdown menu of available banners.

2. The system displays what the banner will look like.
3. The user selects the Choose button.
4. The system returns to the Admin page with the banner displayed as it would on the customer pages.

Alternate flow

1. (At point 3 of the main flow) The user decides to choose a different banner from the original one chosen.
2. The system changes the view to that of the new one.
3. The customer continues at point 3 of the main flow.

Termination

The admin user has chosen their banner.

Post condition

The system displays the admin page.

2.3.9 Requirement 8: Customer cancels their own booking

2.3.9.1 Description & Priority

The customer can cancel their booking using the unique code issued to them when the notification has been sent. This is a medium priority as it is a situation that undoubtedly will arise.

2.3.9.2 Use Case

Scope

The scope of this use case is how a customer cancels one of their own bookings.

Description

This use case describes the process for a customer to cancel a booking that they have already made, using the unique code issued to them after their initial successful booking.

Use Case Diagram

Customer → Bookings → Cancel

Flow Description

Precondition

The customer has made a successful booking.

Activation

This use case starts when the customer navigates to the cancel booking page.

Main flow

1. The customer enters the unique code issued upon booking.
2. The system looks up the database to ensure is valid.
3. The system confirms the booking is deleted.
4. The system returns to the customer homepage.

Alternate flow

1. (At point 1 of the main flow) The customer enters the wrong code.
2. The system looks up the database to check if valid.
3. The system presents an error message to the customer.
4. The customer continues at point 1 of the main flow.

Termination

The system has accepted the cancellation and informed the user.

Post condition

The system returns to the customer homepage.

2.3.10 Requirement 9: Admin user edits their own details

2.3.10.1 Description & Priority

The admin users can edit their own details – email address and password. This is a low priority but still good to have with any application.

2.3.10.2 Use Case

Scope

The scope of this use case is how admin users edit their own details.

Description

This use case describes the process for admin users to edit their account details.

Use Case Diagram

Admin → Registration → Edit

Flow Description

Precondition

An admin user is logged in to the system.

Activation

This use case starts when the admin user navigates to the edit account page.

Main flow

1. The admin user enters a new email address and password.
2. The system updates the database with the new details.
3. The system returns the admin user to the homepage.

Alternate flow

1. The admin user enters an invalid email address.
2. The system presents an error to the admin user.
3. The admin user continues at point 1 of the main flow.

Termination

The system has successfully updated the account details.

Post condition

The system returns to the admin homepage.

2.3.11 Requirement 10: Customer shares booking

2.3.11.1 Description & Priority

Once the customer has made a booking, they are given the option to share on Facebook or Twitter. This is a low priority as it is simply a nice-to-have.

2.3.11.2 Use Case

Scope

The scope of this use case is that a user can share their booking with social media.

Description

This use case describes the process by where the customer shares their booking online using social media.

Use Case Diagram

Customer → Bookings

Flow Description

Precondition

A successful booking request has just been made.

Activation

This use case starts when a customer completes a successful booking request.

Main flow

1. The customer chooses one of the two share options.
2. The system opens up a box to allow the user to share their booking.
3. The customer enters a comment to go with the pre-entered text made by the system.
4. The customer shares the comment.
5. The customer closes the box.
6. The system returns to the booking page.

Alternate flow

1. (At point 2 of the main flow) The system asks the user to log in to the social media.
2. The customer enters their credentials.
3. The customer continues at point 2 of the main flow.

Termination

The customer has shared their comment on social media.

Post condition

The system displays the booking page to the user.

2.4 Non-Functional Requirements

2.4.1 Usability requirement

The user will not need any specific technical knowledge, just a general know-how of computers in general, i.e. email. Someone familiar with how the system works will run through it with the new admin user to ensure set up has worked.

2.4.2 Reliability requirement

The system should be available at all times. The customer must provide a valid email address as the system will reject invalid formats, i.e. missing the @ symbol.

2.4.3 Performance requirement

The system will update the database and UI almost instantly once the requests are submitted. User tasks are not time critical but it is foreseen that a user will not need more than a few minutes at a time using the system.

2.4.4 Supportability requirement

A system owner will support the system. The system can be used by any platform that can access the internet.

2.4.5 Security requirement

The booking requests are stored in a secure database, with preventative measures in place to prevent from SQL injections. The admin users' password will be encrypted when stored in the database for further security.

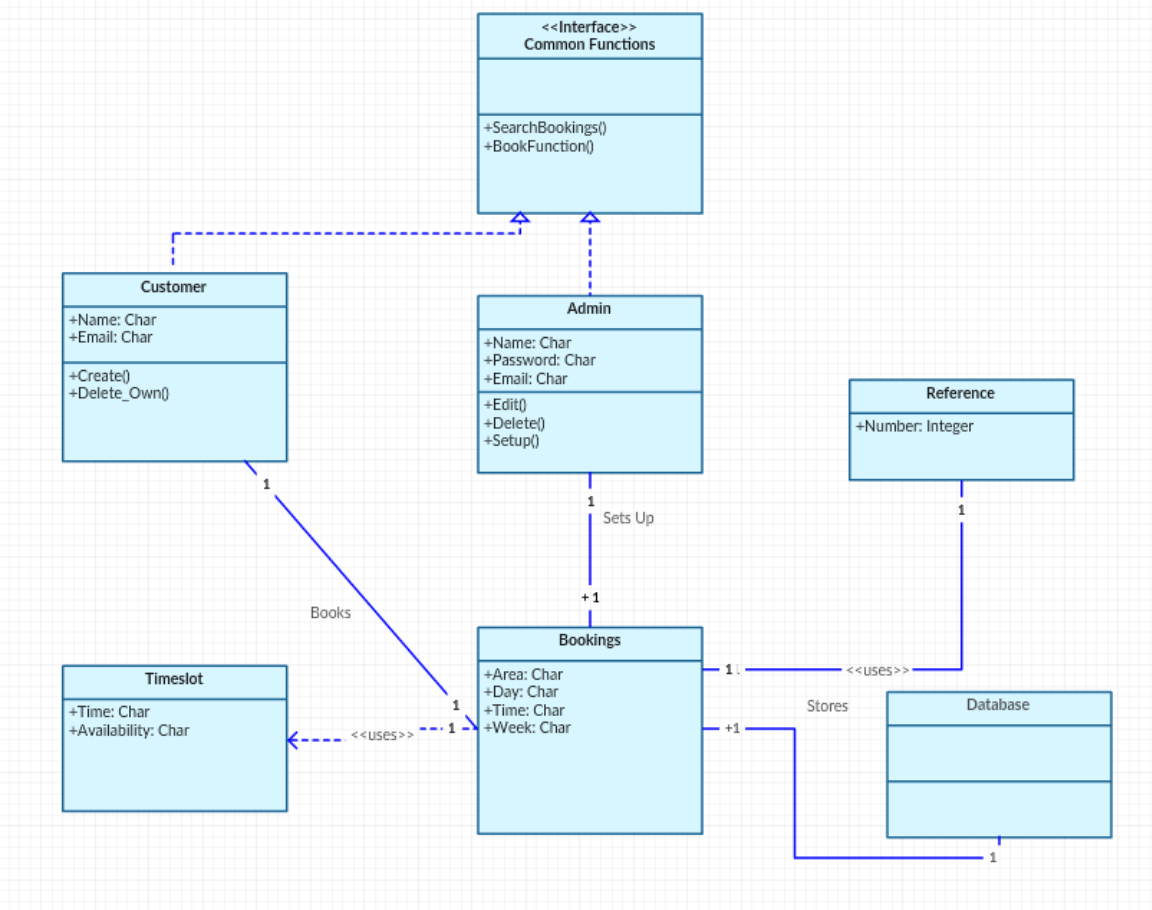
2.4.6 Interface requirement

The system can be integrated with any online site as it only requires a hyperlink to the system's site. A specific URL will be created per club, so that it will navigate to a unique URL. The club can request their page to be personalized if they desire. Data is created on the site itself via booking submissions.

2.4.7 Operation requirement

All operations will be managed and maintained by the system owner.

2.5 Design and Architecture



The system was created with simplicity in mind. The user interface interacts with the database in the background, in order to display the required content to the user, depending on what the user has selected on the various pages.

This architectural diagram represents the system as a whole. All the areas have been defined, each with their specific implementations, and interactions with other parts of the system. This design was created to capture all the sections of the system.

The admin has the characteristics of a name, password and email address. They have the options of edit, delete and setup bookings which are vital to the system. The bookings are all stored in a database for easy access. Each booking created uses a booking reference number in order for to distinguish between bookings.

The customer has a name and email address for when they are making a booking. They also have the option of deleting their own booking, via the reference number created when the booking was made.

2.6 Implementation

For the majority, the application uses PHP to combine with MySQL in order to post updates to the database, which in turn then displays what the user sees. PHP is fast, efficient, and works great in almost any web browser as well as integrating into almost any server (Tech-faq.com, 2017).

Here are some examples of the key areas of the application, along with some code snippets and explanations of how the functionality works.

User adding a new day to the one of their areas:

```
$sql = "UPDATE `daysUsed` SET `$name`='Yes' WHERE `days` = '".$_POST['userOption']."';";
$sql .= "UPDATE `$name` SET `".$_POST['userOption']."`='Free'";
```

As you can see it sets the value of the posted option to ‘Yes’ – which are the ones that are to be displayed – and then sets the values to be ‘Free’ to let the user know they are available.

The system then calls the database like below to display the table:

```
$query = mysqli_query($db,"
SELECT *
FROM ".$_POST['userOption']."
WHERE Used = 'Yes'
ORDER BY id asc;
```

This will store all the values required and then the system displays them in a table by echoing each row.

Another interesting feature is the ability for the system to automatically display the banner options when the user clicks on one of the items in the dropdown menu. This is achieved using a simple function and it saves the user having to click a button, and instead can just select the option.

It uses a simple if statement and to choose the option from the user's selection:

```
if(selectDd.value === "1")
      document.getElementById("icon").innerHTML =
           "<img id='iconSize' src='images/football.jpg'/>";
```

To display the dropdown menu, for both customers and admin users, of all the areas available, this is achieved with a nice while loop in PHP to gather all the rows and output them into a HTML select box:

```
do{
    unset($id, $name);
    $id = $row['allAreas'];
    $allAreas = $row['allAreas'];
    echo '<option value="'.$allAreas.'">'.$allAreas.'</option>';
}
while ($row = $stmt->fetch()) ;
```

This also puts the area as the value of each option to make the code easier to read, especially when calling the area chosen from other classes.

This code is used across the application in various sections, and works well to display the required output to the user. There are different variations of it in different places, depending on what the user has selected.

Another code snippet here shows how the system will reject invalid email address formats, on any of the forms.

```
    if (!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
?>
          <script type="text/javascript">
               alert("Please enter a valid email address.");
               window.location.href = "contact.php";
           </script>
    <?php
        die();
    }
```

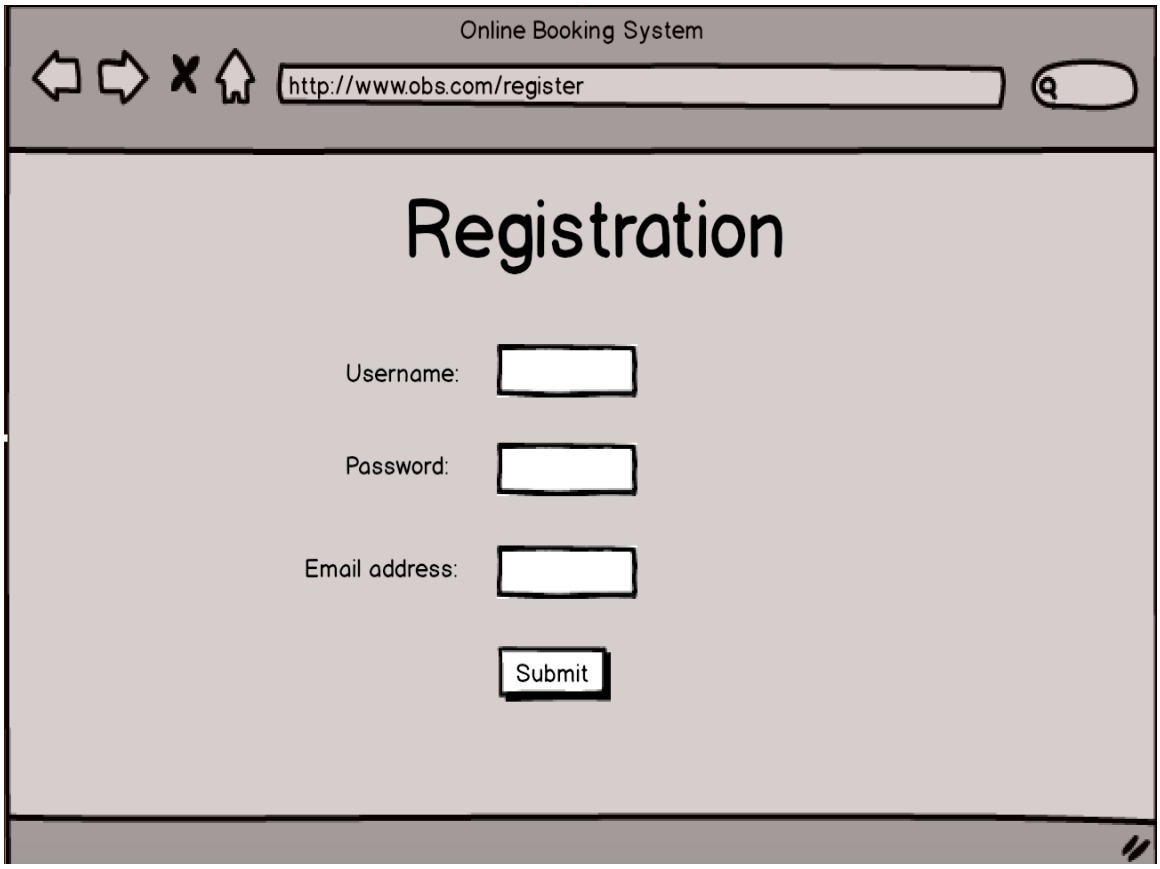
This checks the email variable entered in the form, and uses a PHP function to validate it to ensure it contains '@' and '.', in that order. A small piece of JavaScript is included to display a message to the user, and then to re-load the page.

2.7 Graphical User Interface (GUI) Layout

This section shows an example of how each page will look when fully designed. While there may be a few tweaks along the way, this will give the customer a very good idea of what to expect. Prototyping is very important for any project design. While skipping prototyping might save some time during design, that surplus can be lost many times over in development (Designshack.net, 2017).

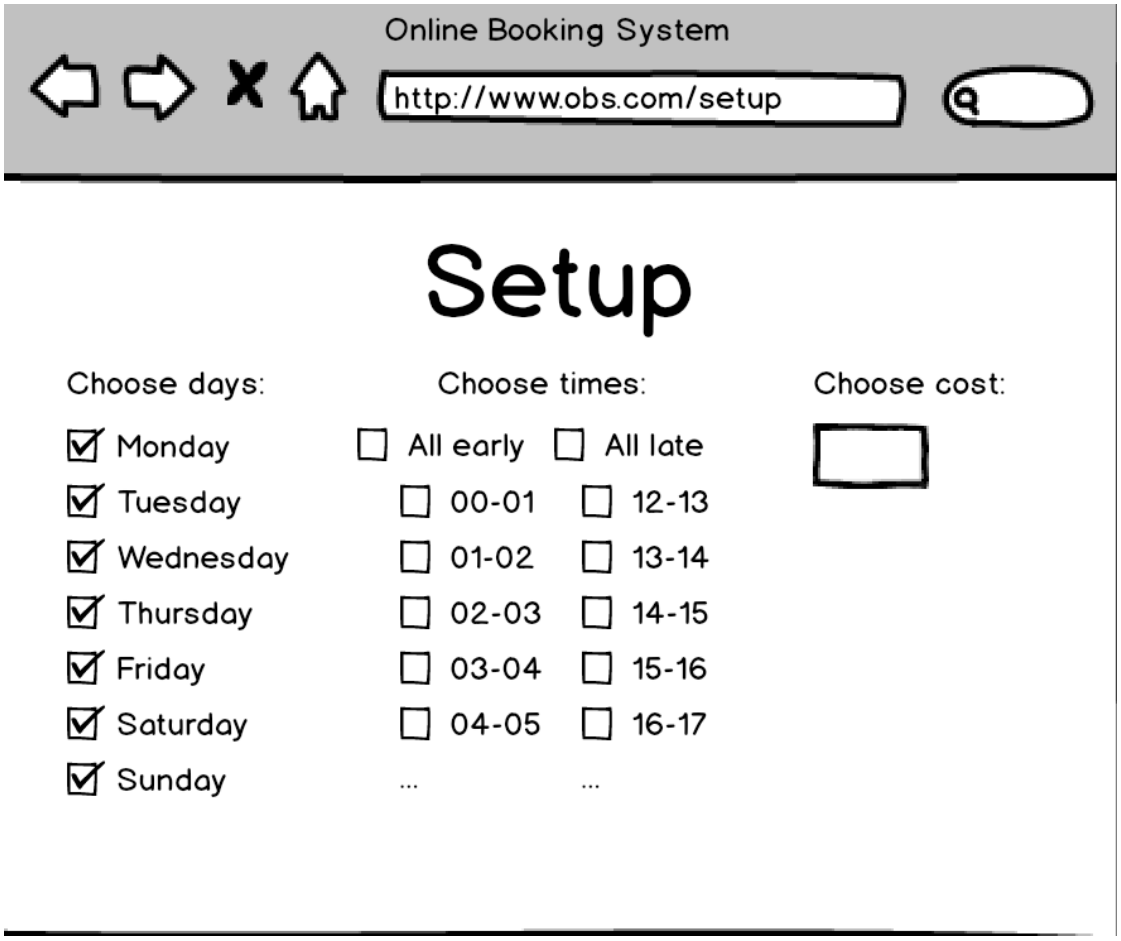
2.7.1 Admin Registration

This displays how the registration page will look to the admin before they register with their username, password and email address. Each field is mandatory, and the email address must be unique, and in a valid format.



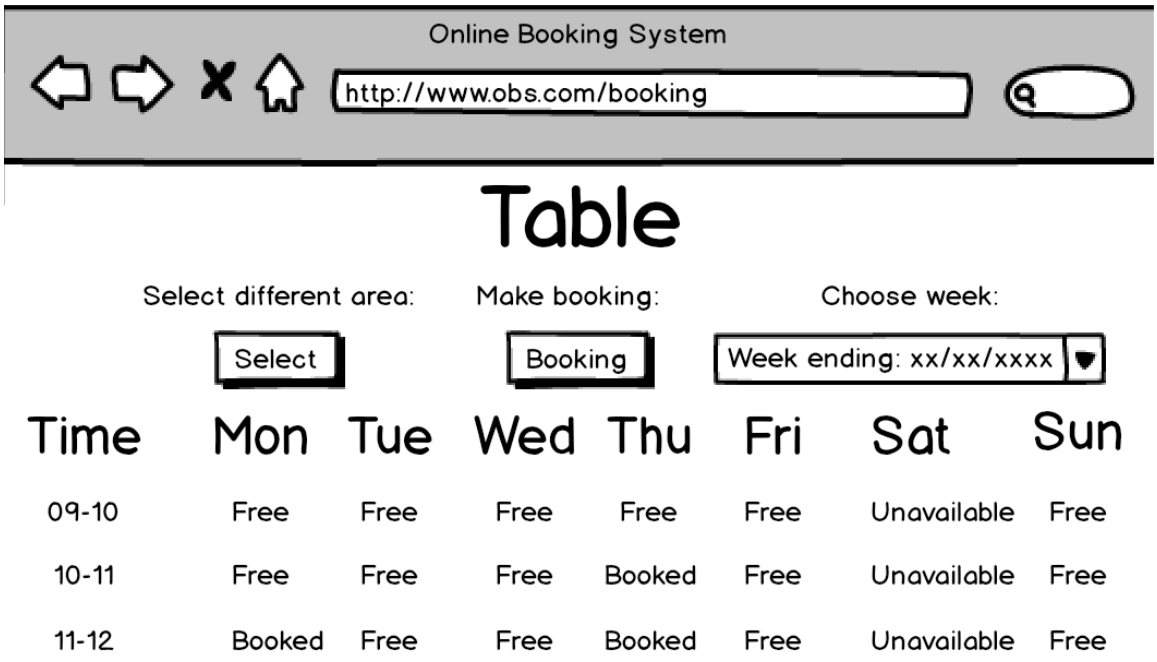
2.7.2 Setup of specific features

Once an area has been chosen, the user is presented with the options available for their area. This includes days of the week, hours of the day and the cost per hour.



2.7.3 Booking Request

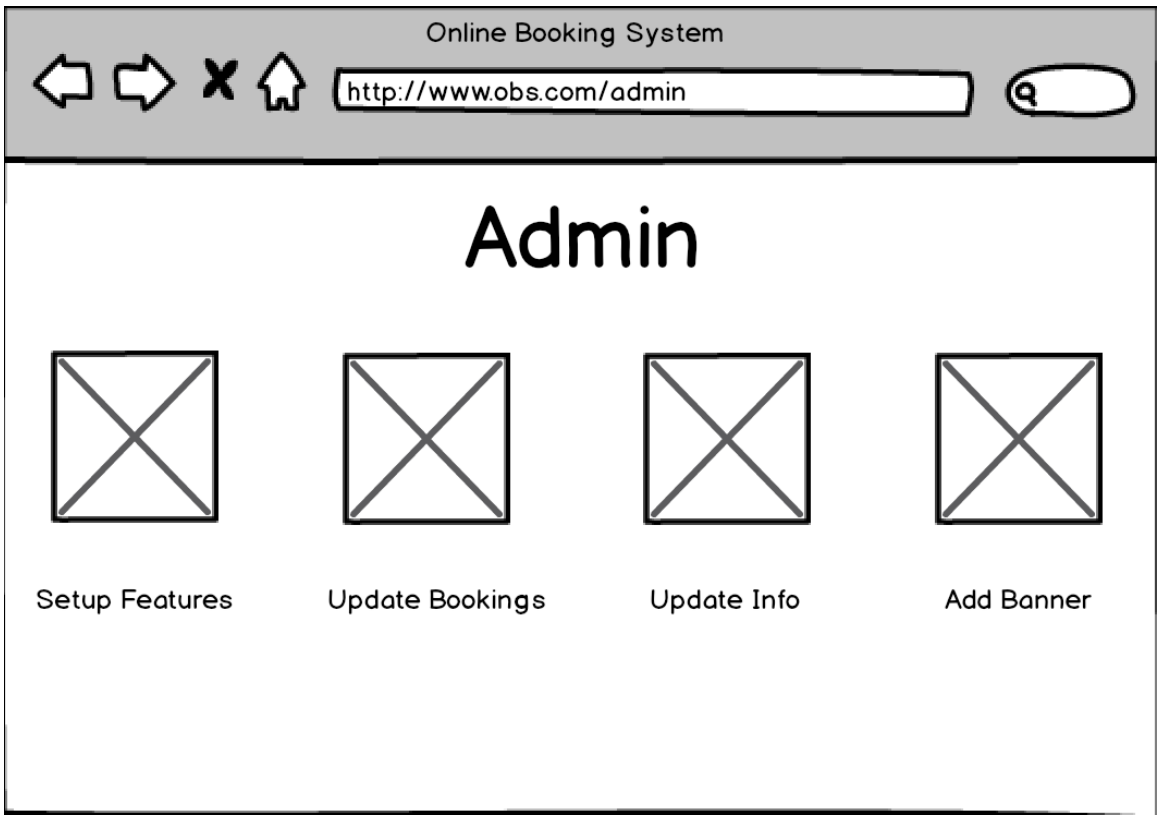
This shows what the page will look like when a customer has selected their area of interest, by displaying a table of available days and times. Each time will have either ‘Free’, ‘Booked’ or ‘Unavailable’ displayed. There is also an option to choose a different week, or to go back and choose a different area.



Time	Mon	Tue	Wed	Thu	Fri	Sat	Sun
09-10	Free	Free	Free	Free	Free	Unavailable	Free
10-11	Free	Free	Free	Booked	Free	Unavailable	Free
11-12	Booked	Free	Free	Booked	Free	Unavailable	Free

2.7.4 Admin Page

This display is for the features the admin user has. From here they can: change the features and timeslots available, update or delete existing bookings, change their own information, e.g. email address or password, or add a personalised banner to the application.



2.8 Admin Testing

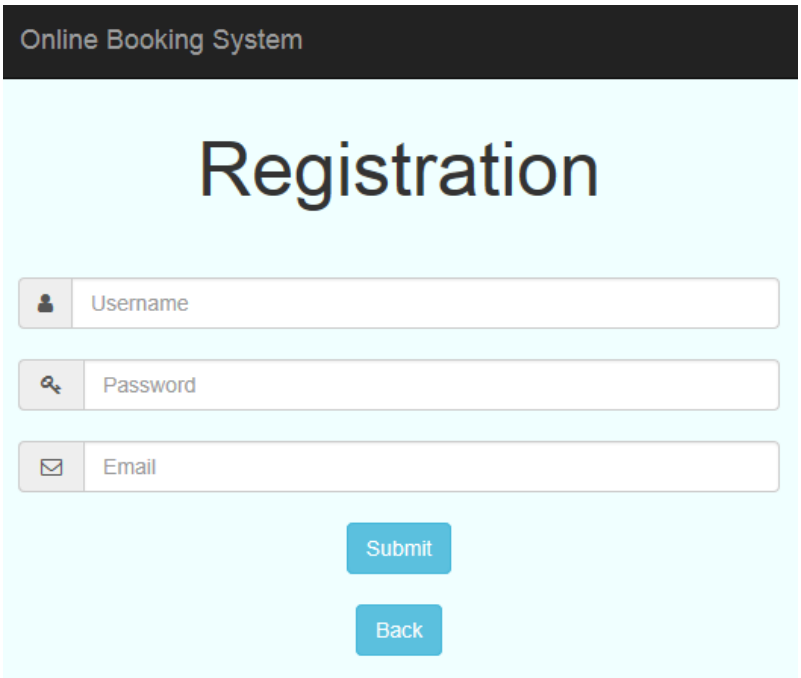
Testing is very important as it is required to point out the defects and errors that were made during the development phase. It is also very important to ensure the quality of the product (Level et al., 2017).

The following sections were tested as an admin user:

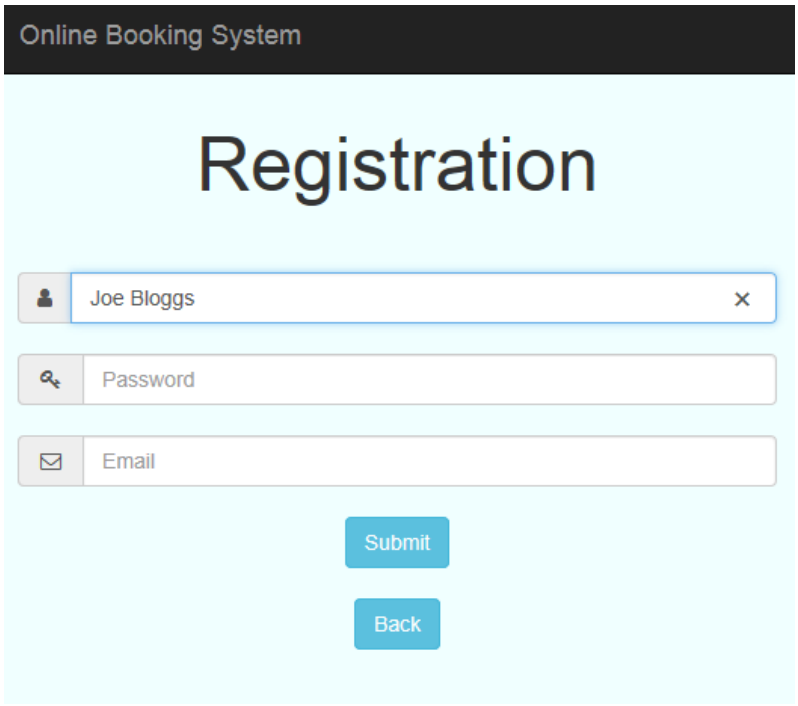
- Registration
- Login
- Navigation Bar
- Setup
- Update Bookings
- Delete Bookings
- Update Information
- Add Banner
- View All Bookings

2.8.1 Registration

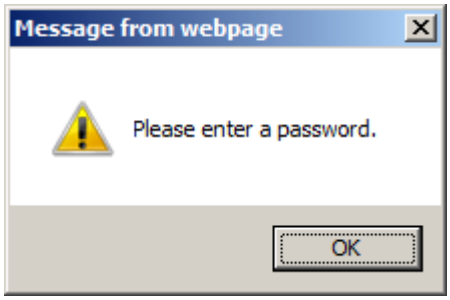
The registration page.



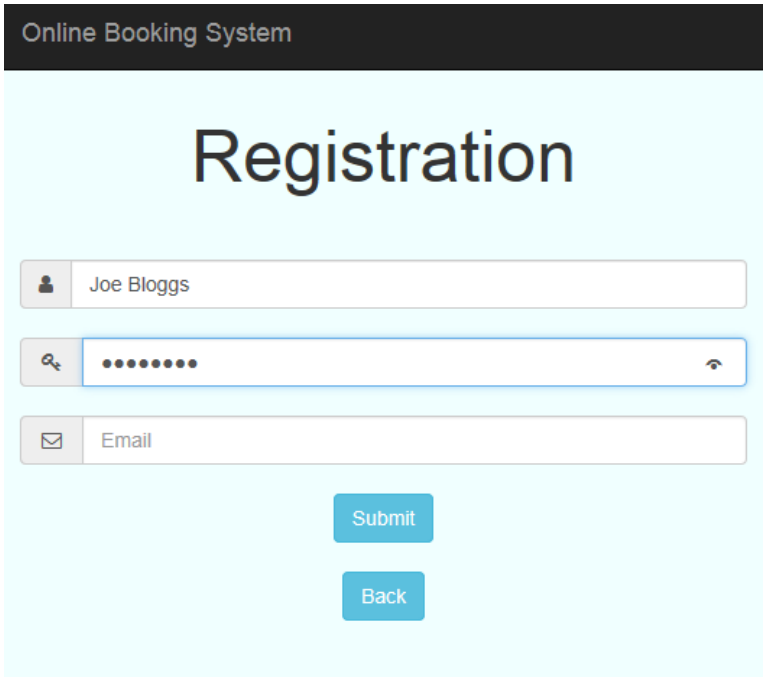
Test RP1: Register with just a username



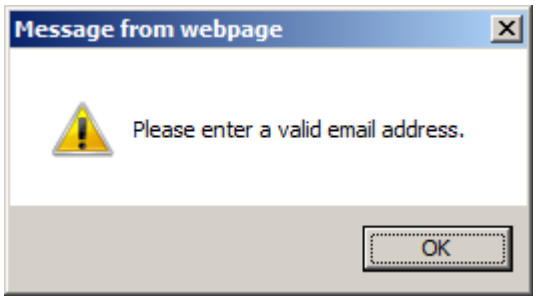
Result: The system asks for a password.



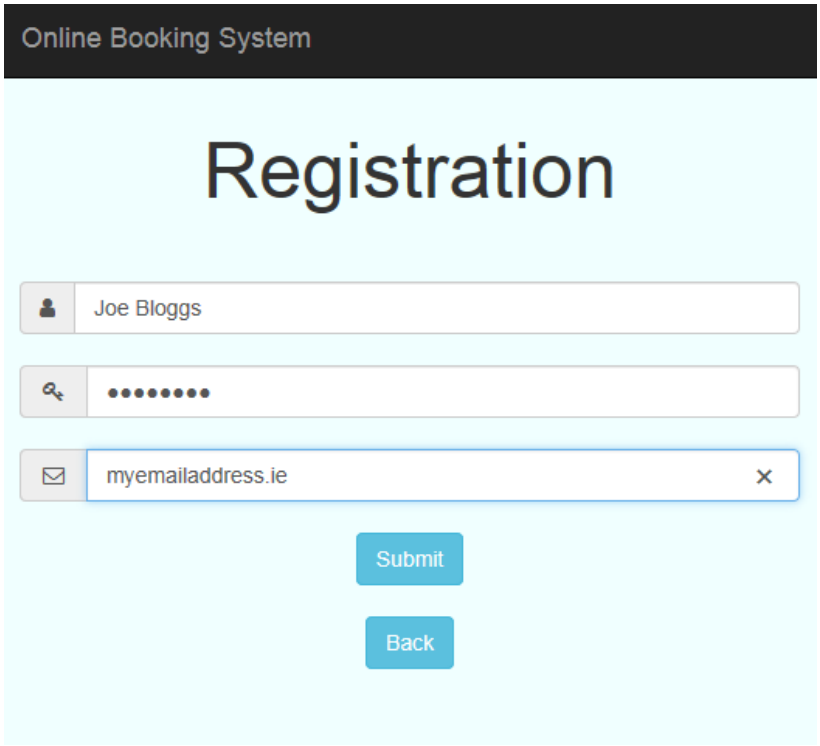
Test RP2: Register with a username and password, but no email address.



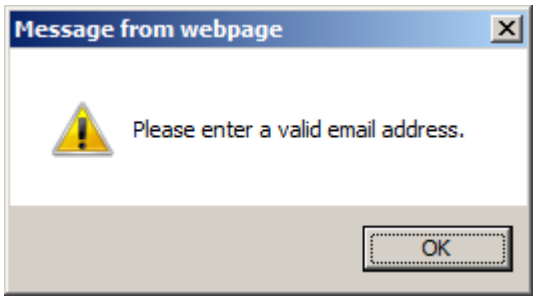
Result: The system asks for an email address.



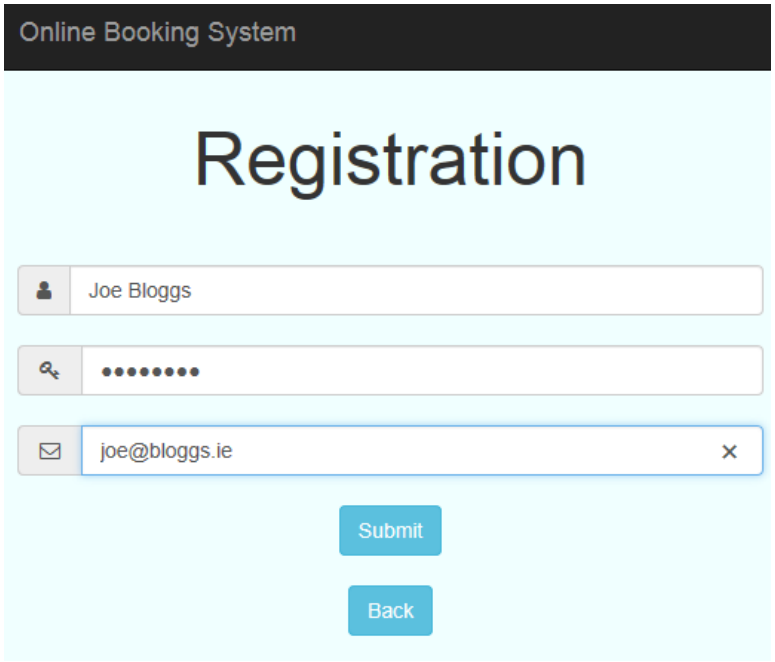
Test RP3: Register with a username, password and an invalid email address.



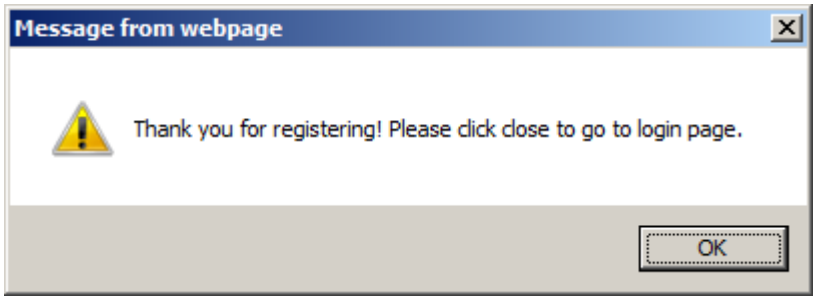
Result: The system asks for a valid email address.



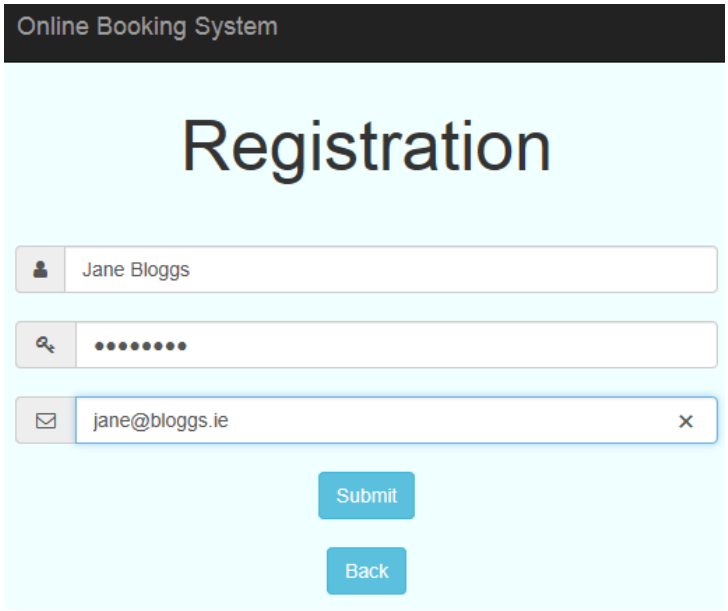
Test RP4: Register with a username, password and a valid email address.



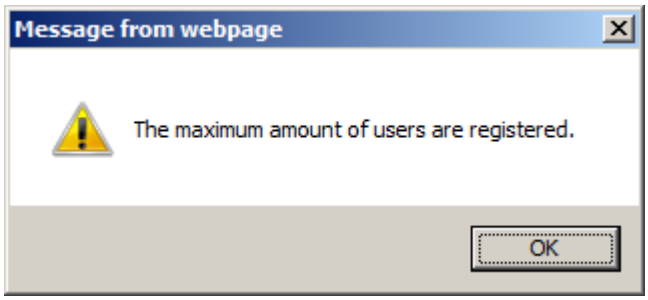
Result: The system has successfully registered the user.



Test RP5: Attempt to register when the maximum number of users has already been registered (two).

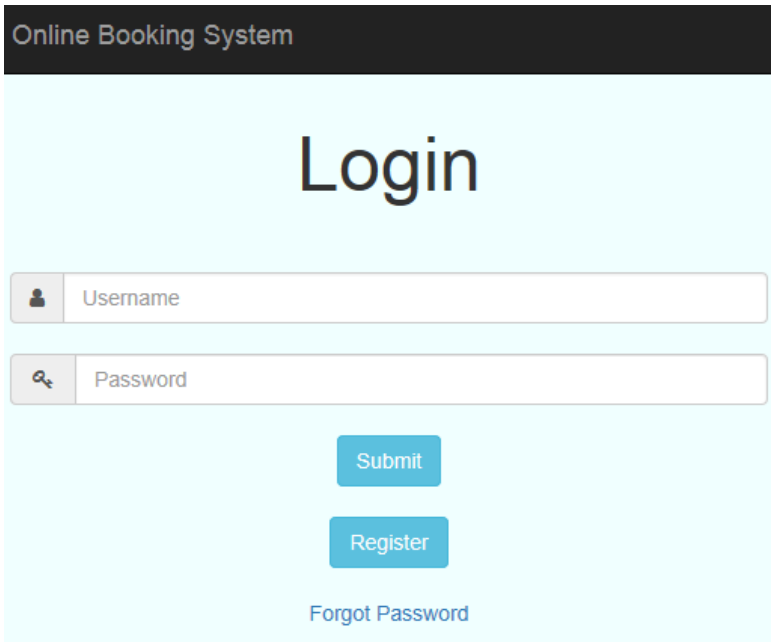


Result: The system displays that the maximum number of users are registered.

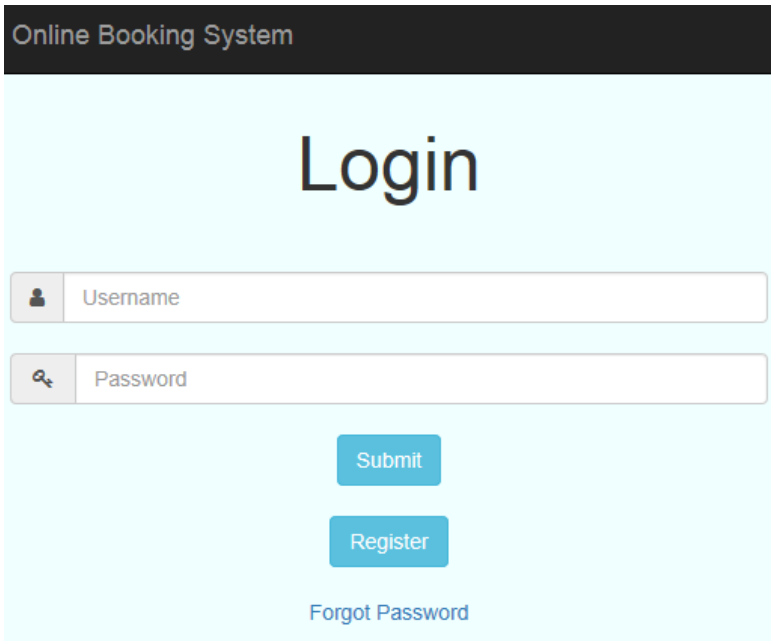


2.8.2 Login

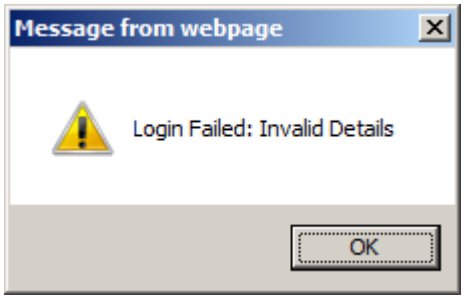
The login page.



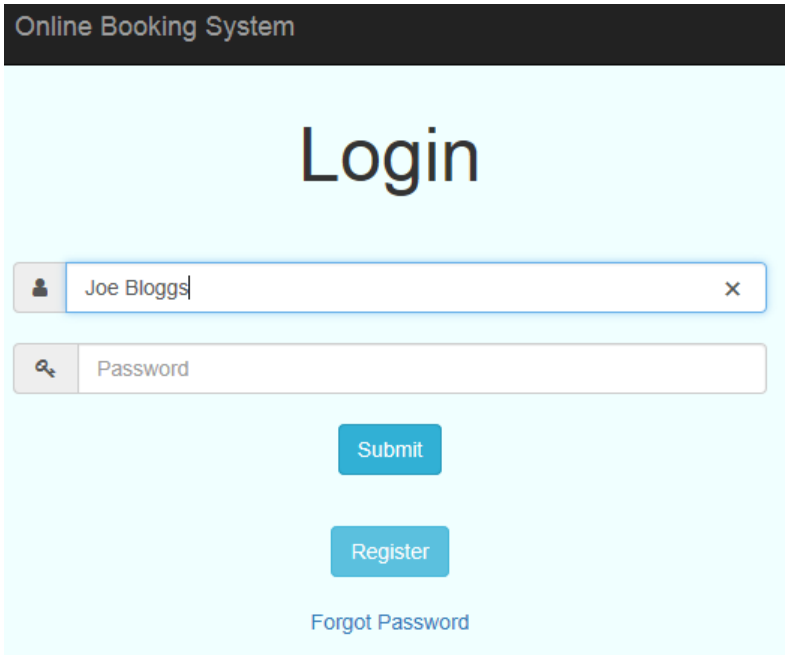
Test LP1: Attempt to log in without any credentials.



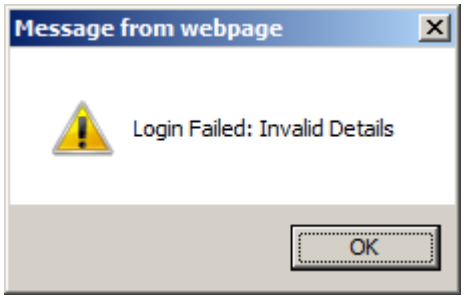
Result: The system displays the details were invalid.



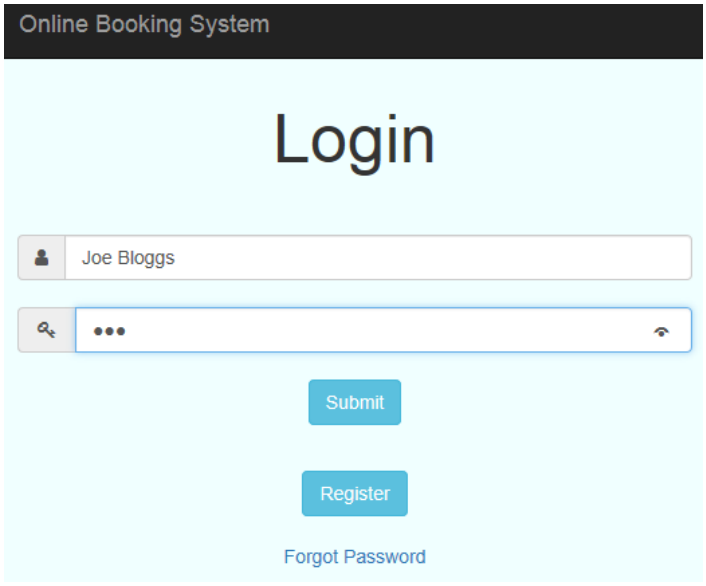
Test LP2: Attempt to log in with a valid username, but no password.



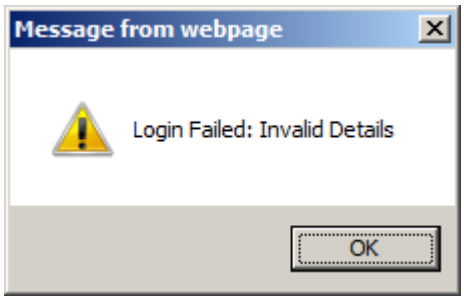
Result: The system displays the details were invalid.



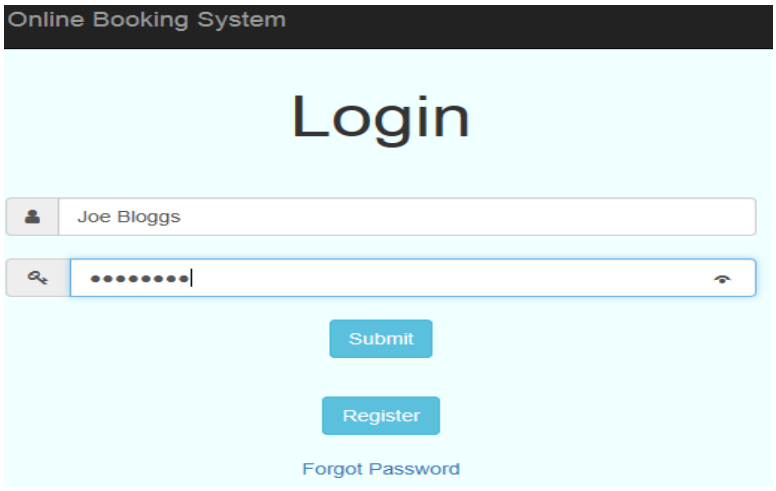
Test LP3: Attempt to log in with a valid username, but an invalid password.



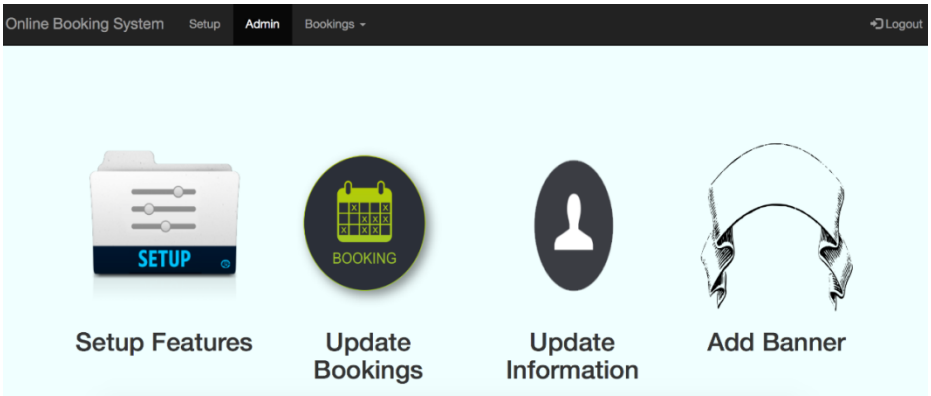
Result: The system displays the details were invalid.



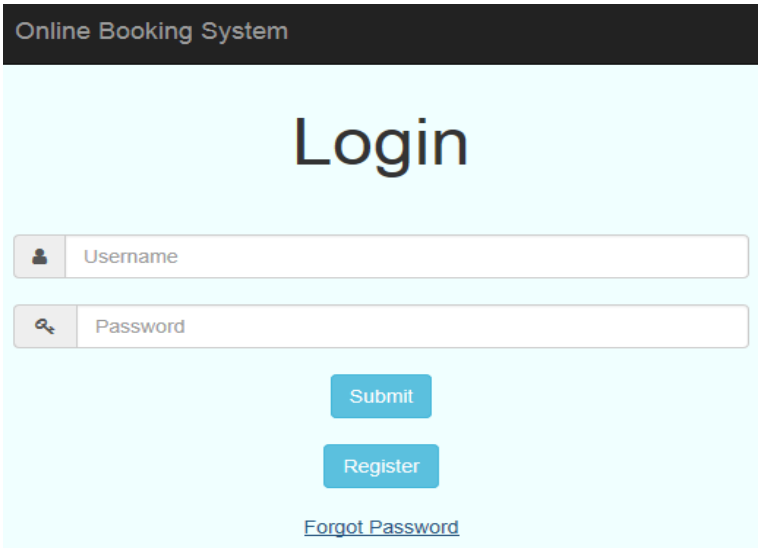
Test LP4: Attempt to log in with a valid username and password.



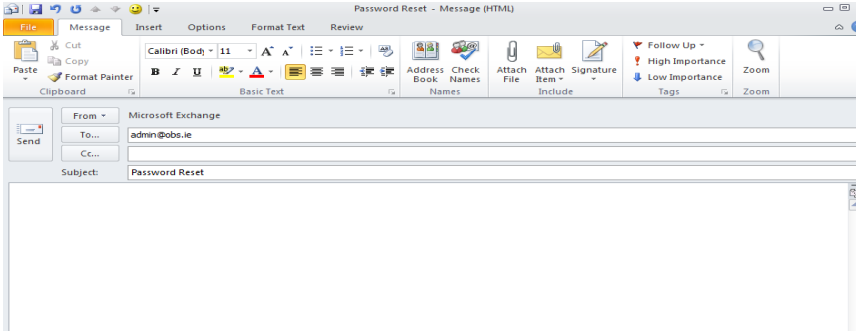
Result: The system successfully logs in.



Test LP5: Select the 'forgot password' link.

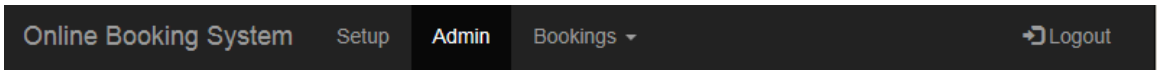


Result: The system opens up an empty email to the admin, with the subject auto-completed.

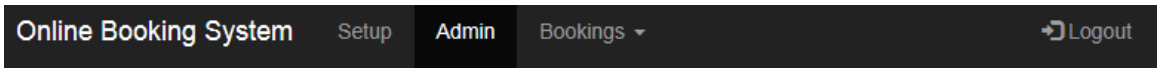


2.8.3 Navigation Bar

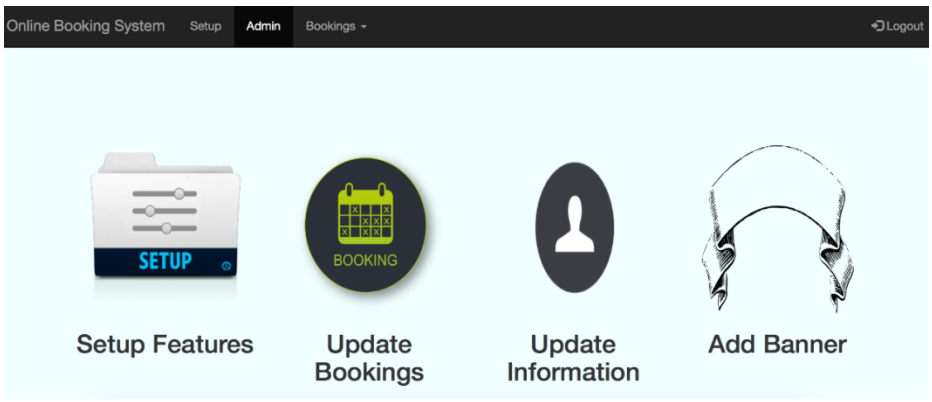
The navigation bar.



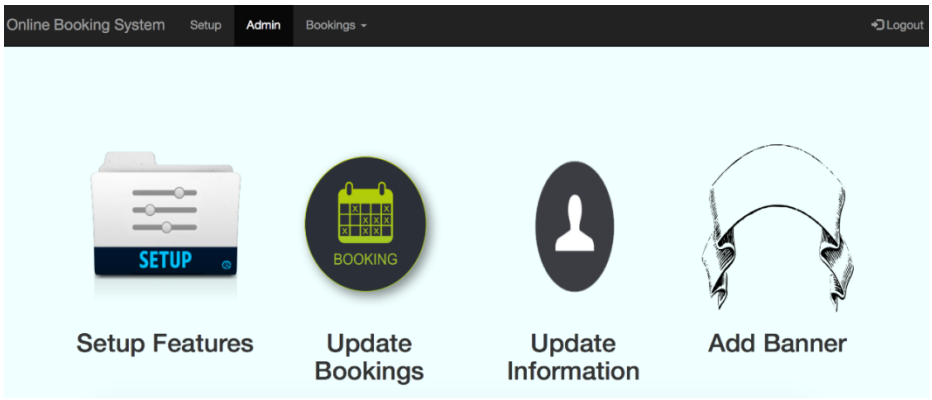
Test NB1: Click the ‘Online Booking System’ text.



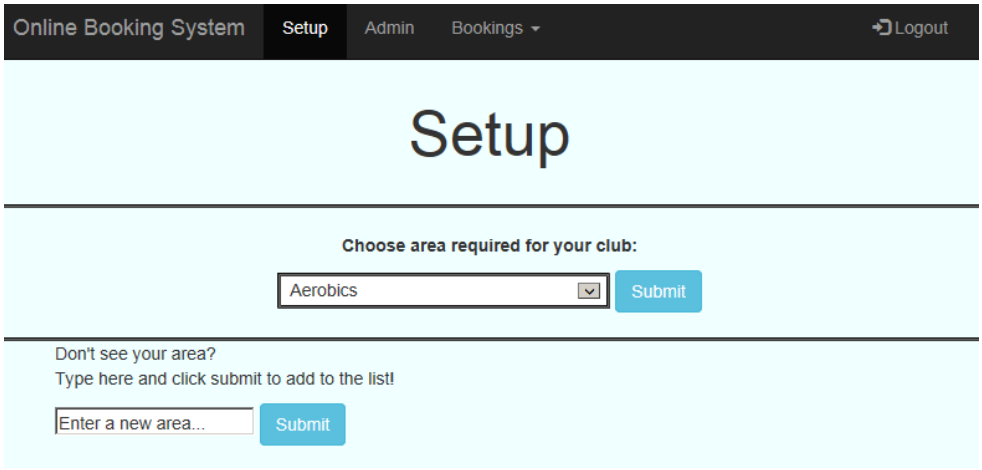
Result: The system takes no action.



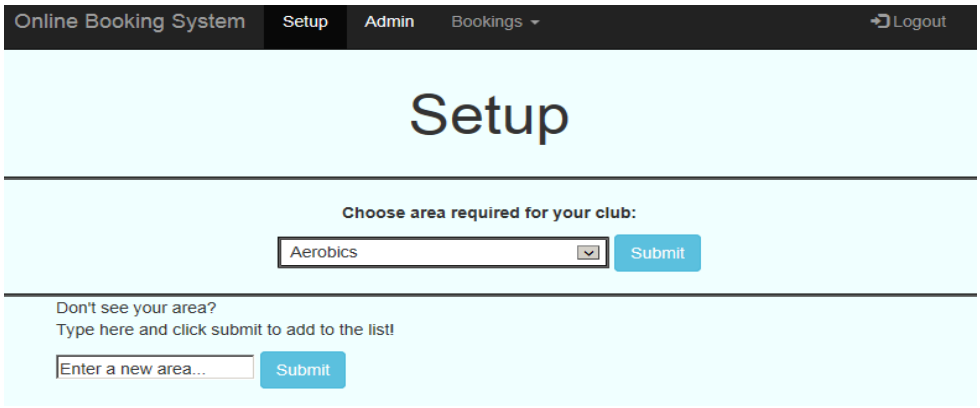
Test NB2: Click the ‘Setup’ text.



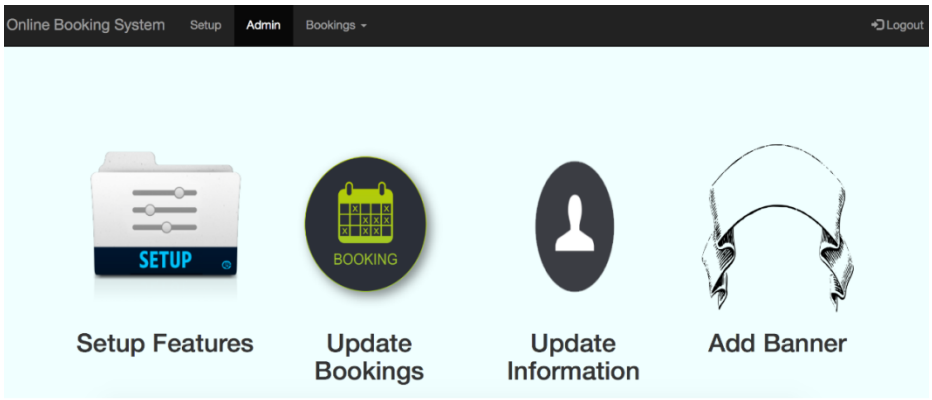
Result: The system brings you to the setup page, and the highlighted text is now ‘Setup’.



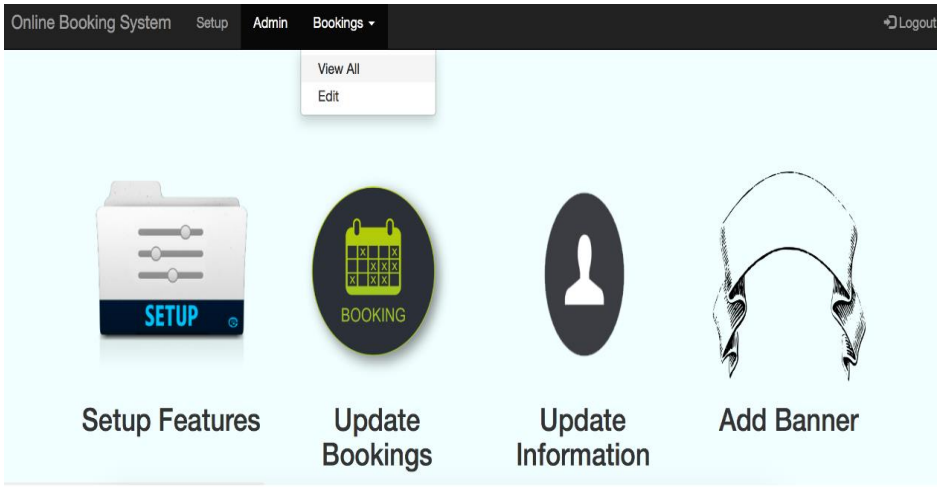
Test NB3: Click the ‘Admin’ text.



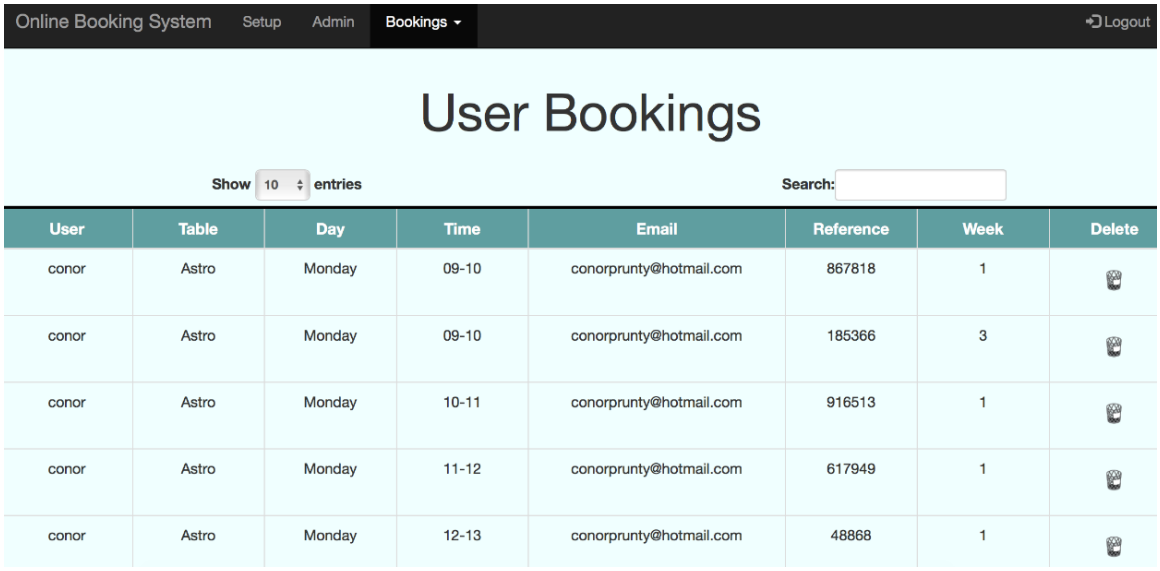
Result: The system brings you to the admin page, and the highlighted text is now ‘Admin’.



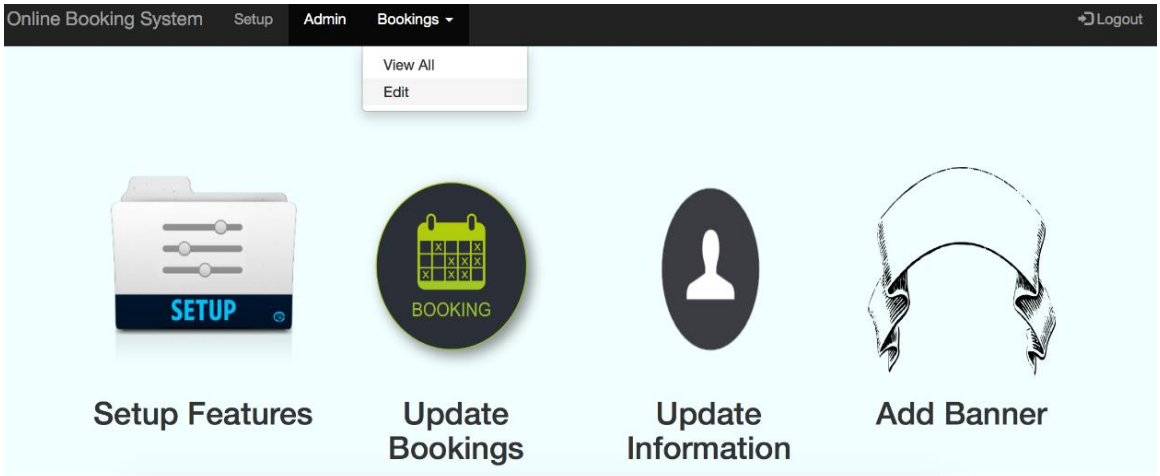
Test NB4: Click the ‘Bookings’ dropdown and select ‘View All’.



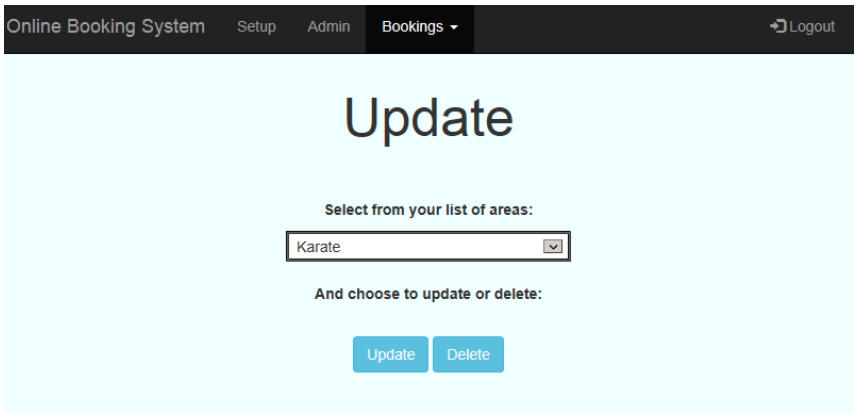
Result: The system brings you to the page of all the bookings, and the ‘Bookings’ heading is highlighted.



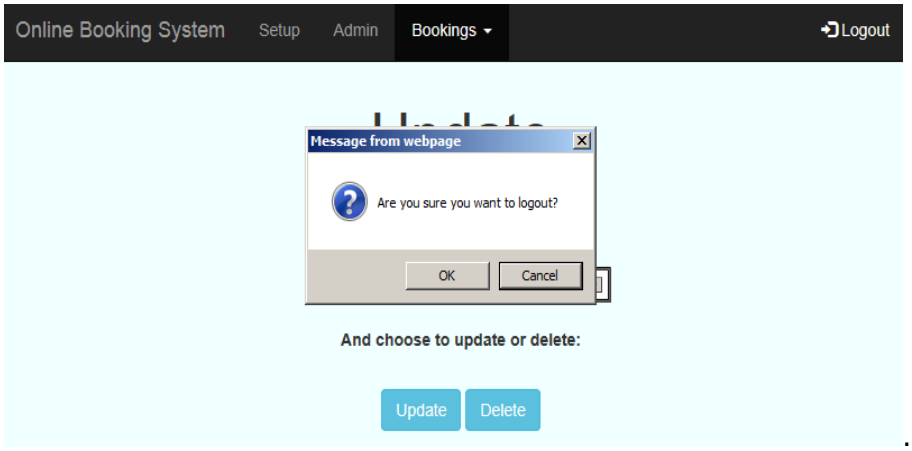
Test NB5: Click the 'Bookings' dropdown and select 'Edit'.



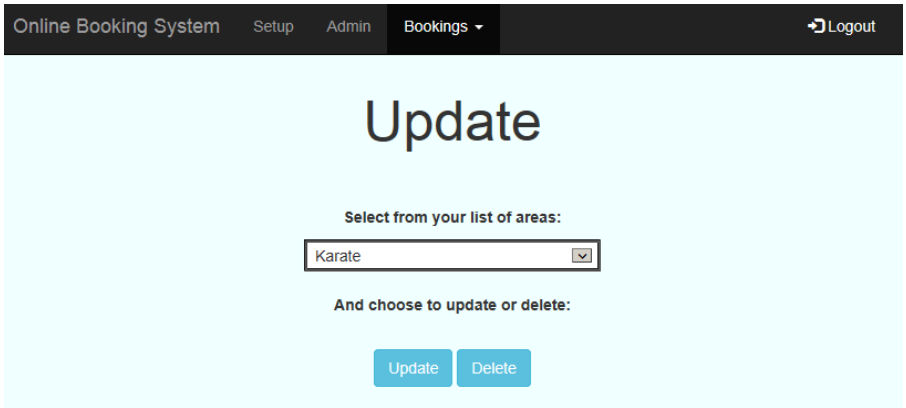
Result: The system brings you to the page to edit bookings, and the 'Bookings' heading is highlighted.



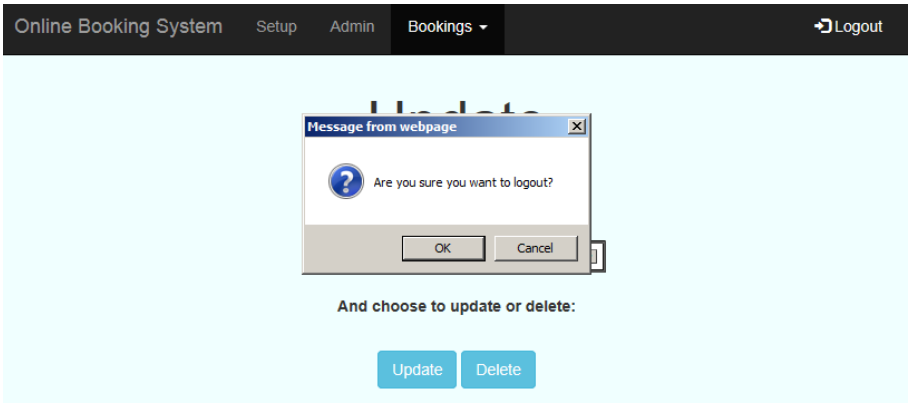
Test NB6: Click the 'Logout' button and select 'Cancel'.



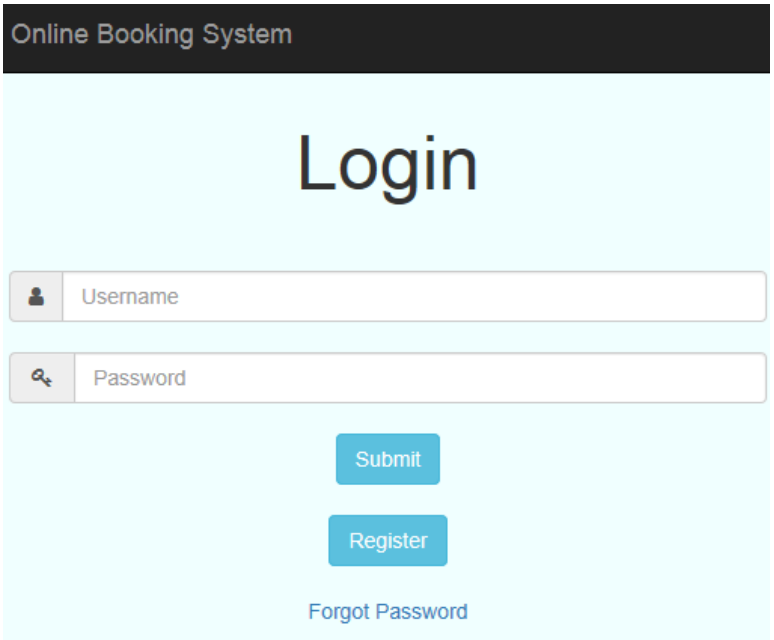
Result: The system takes no action, and returns to the page you were on.



Test NB7: Click the ‘Logout’ button and select ‘OK’.

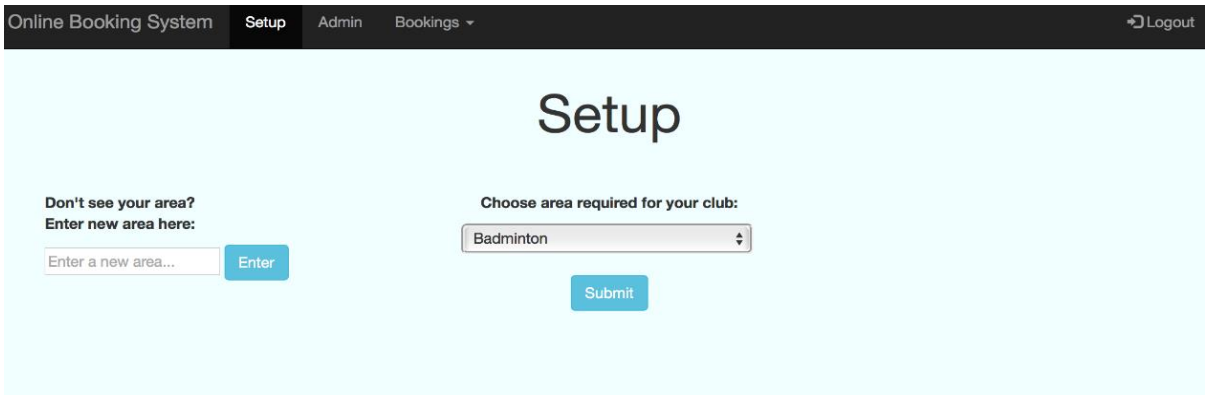


Result: The system logs you out of the system, and returns to the index page.

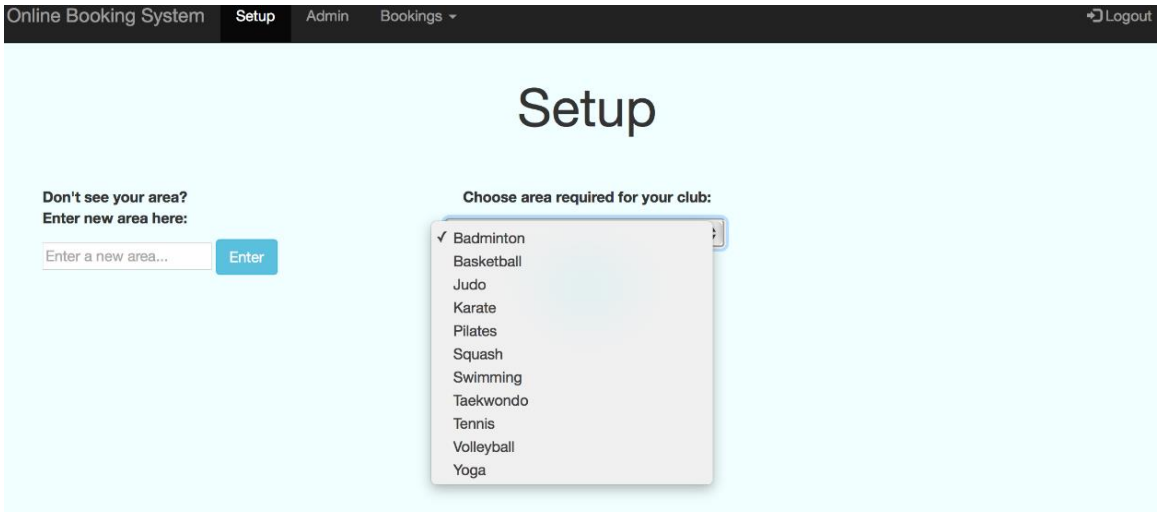


2.8.4 Setup

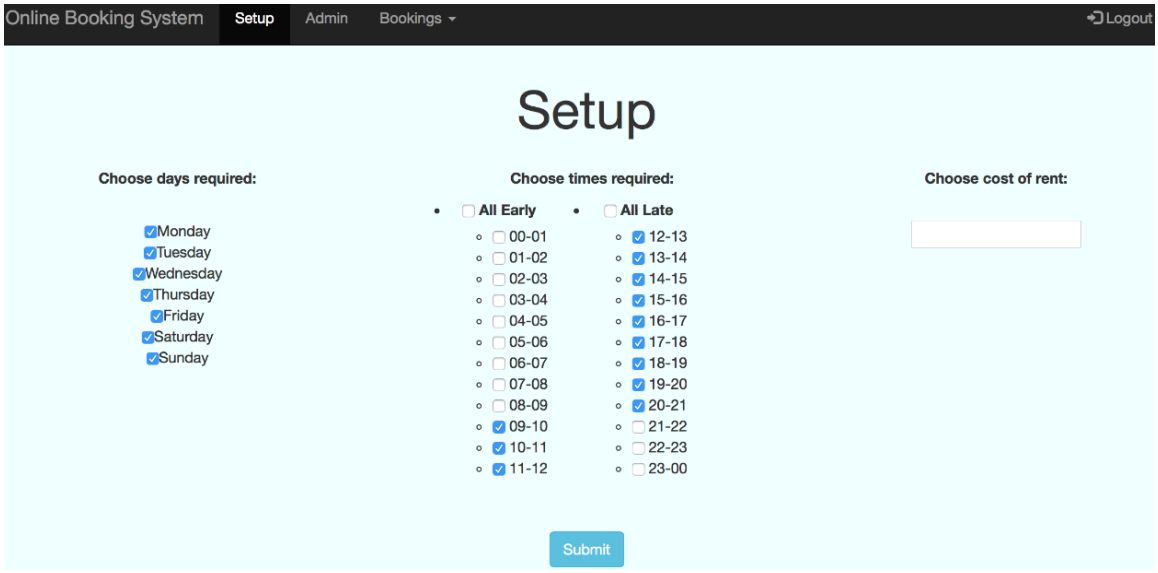
The setup page.



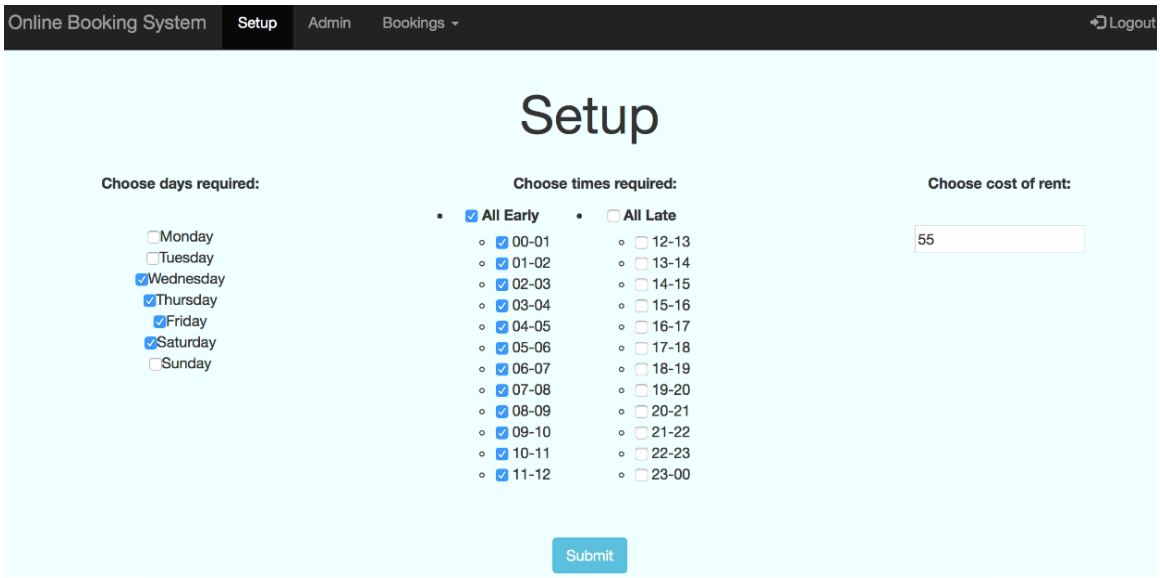
Test SP1: Click the dropdown menu and select an option.



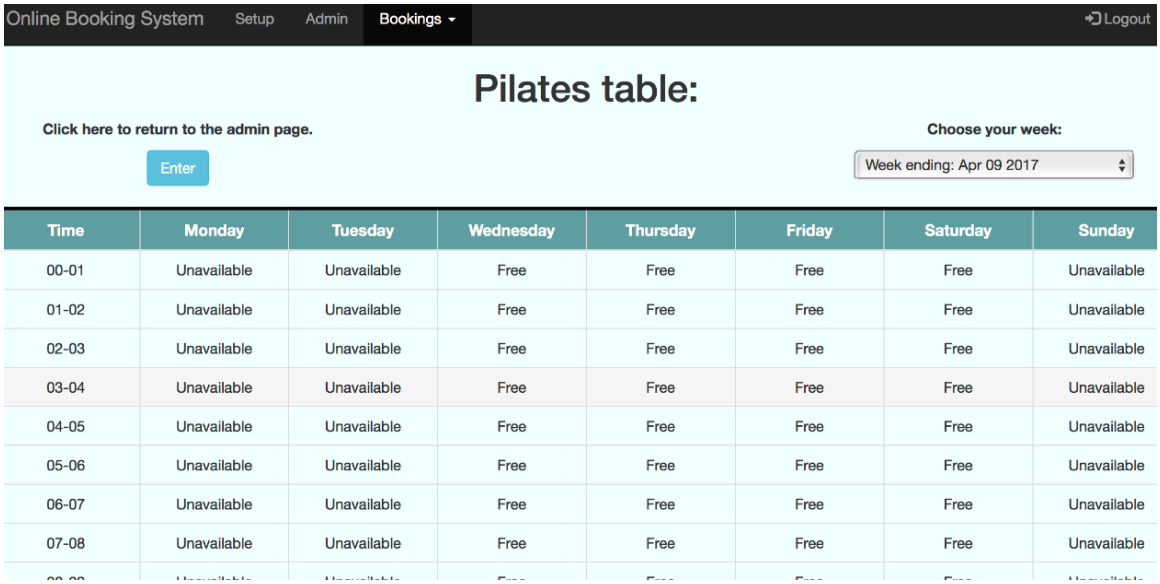
Result: The system navigates to the second setup page.



Test SP2: Choose an option of days, times and cost.

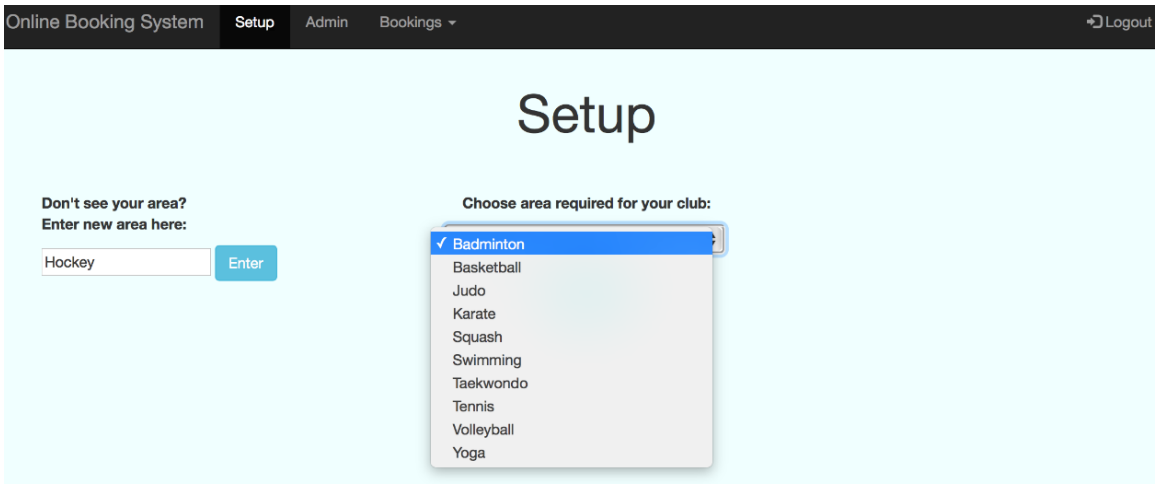


Result: The system displays the values in a table.

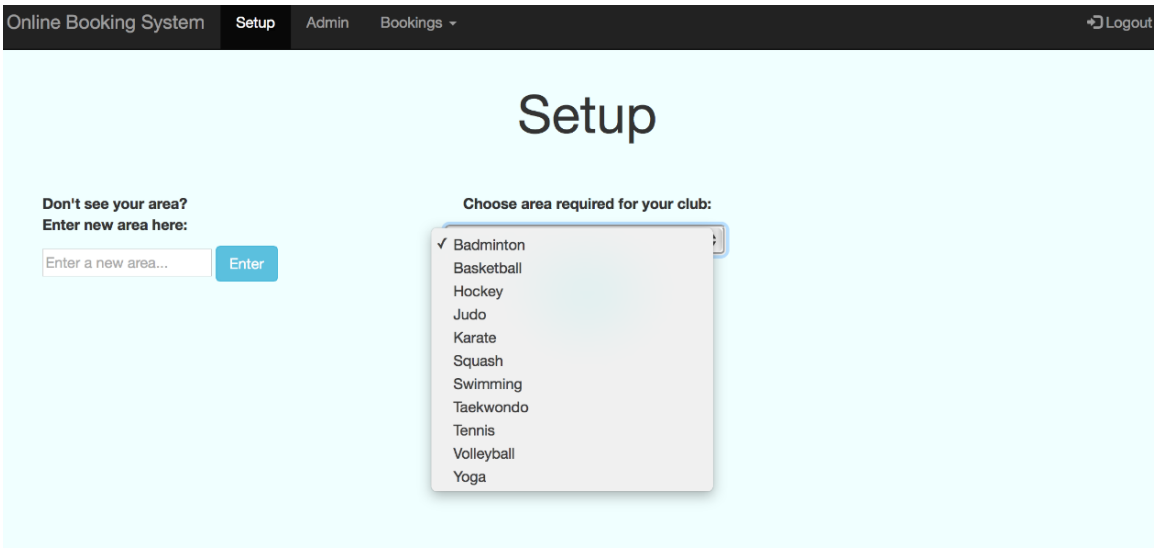


Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
00-01	Unavailable	Unavailable	Free	Free	Free	Free	Unavailable
01-02	Unavailable	Unavailable	Free	Free	Free	Free	Unavailable
02-03	Unavailable	Unavailable	Free	Free	Free	Free	Unavailable
03-04	Unavailable	Unavailable	Free	Free	Free	Free	Unavailable
04-05	Unavailable	Unavailable	Free	Free	Free	Free	Unavailable
05-06	Unavailable	Unavailable	Free	Free	Free	Free	Unavailable
06-07	Unavailable	Unavailable	Free	Free	Free	Free	Unavailable
07-08	Unavailable	Unavailable	Free	Free	Free	Free	Unavailable

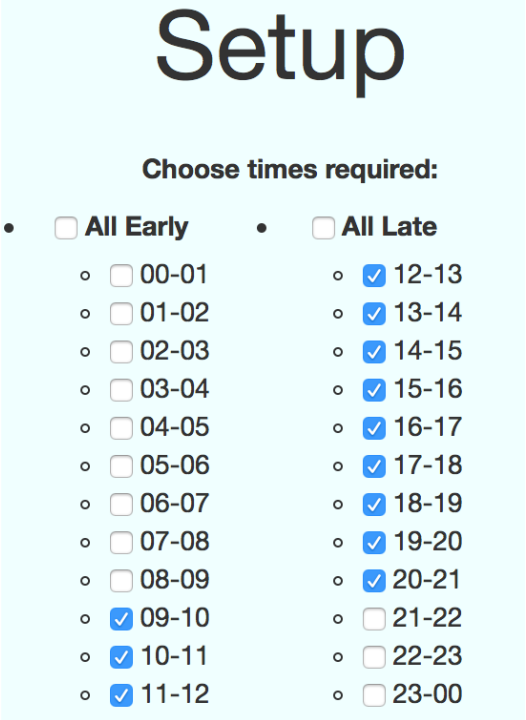
Test SP3: Add a new area to the available list.



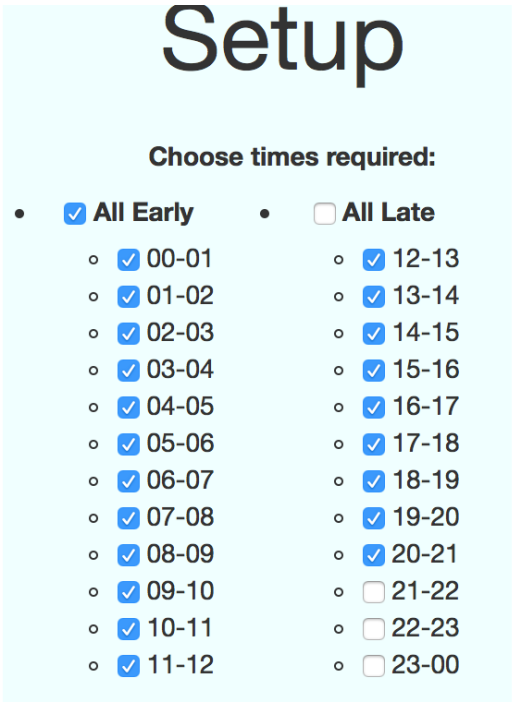
Result: The system now has the option of that area in the setup dropdown.



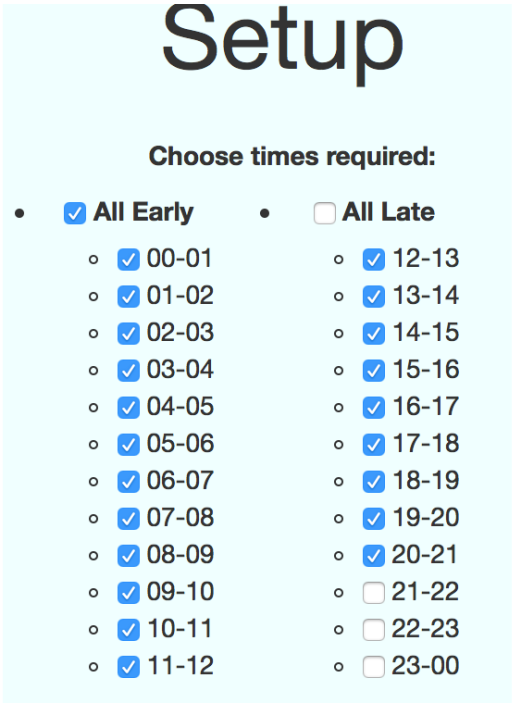
Test SP4: Choose the 'all early' button on the setup page.



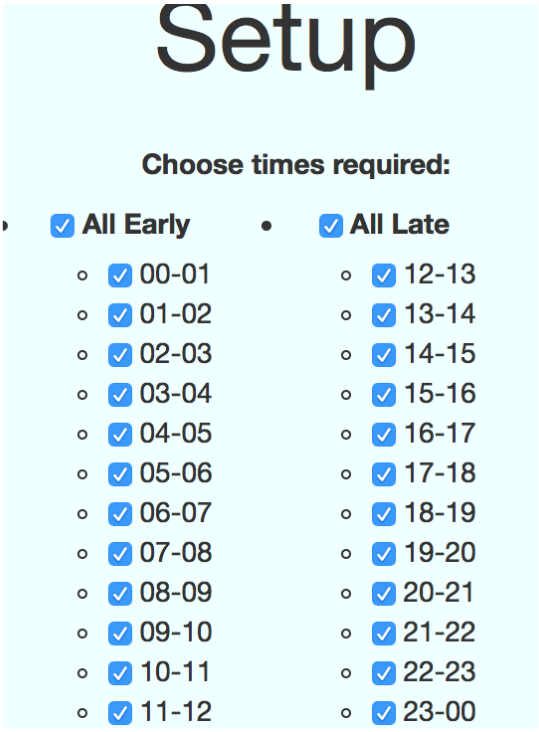
Result: The system auto-selects all from that list.



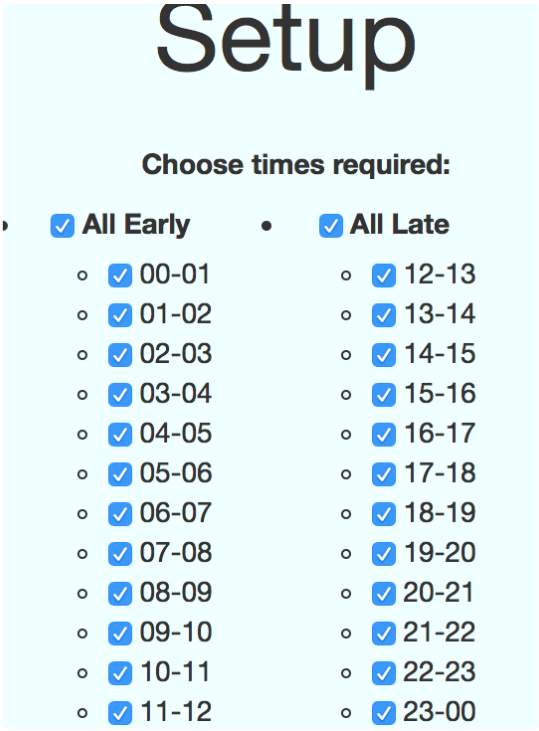
Test SP5: Choose the ‘all late’ button on the setup page.



Result: The system auto-selects all from that list.

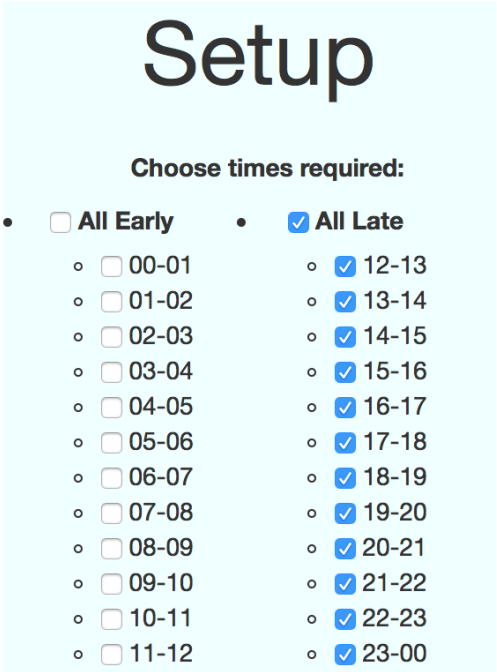


Test SP6: Re-click the 'all early' button.

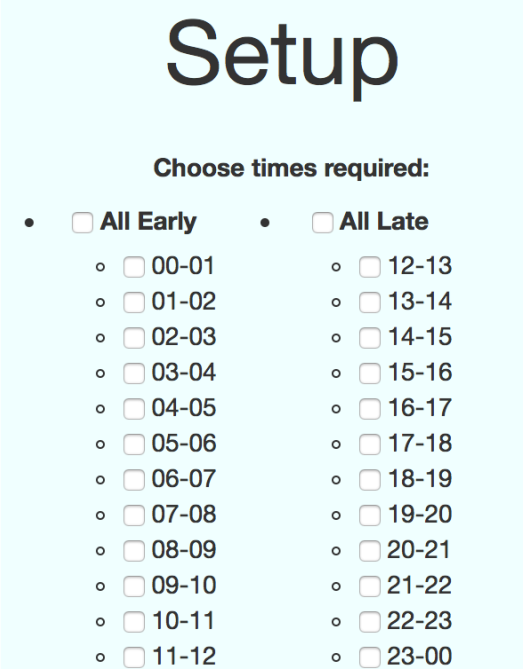


Result: The system unchecks each item in the list.

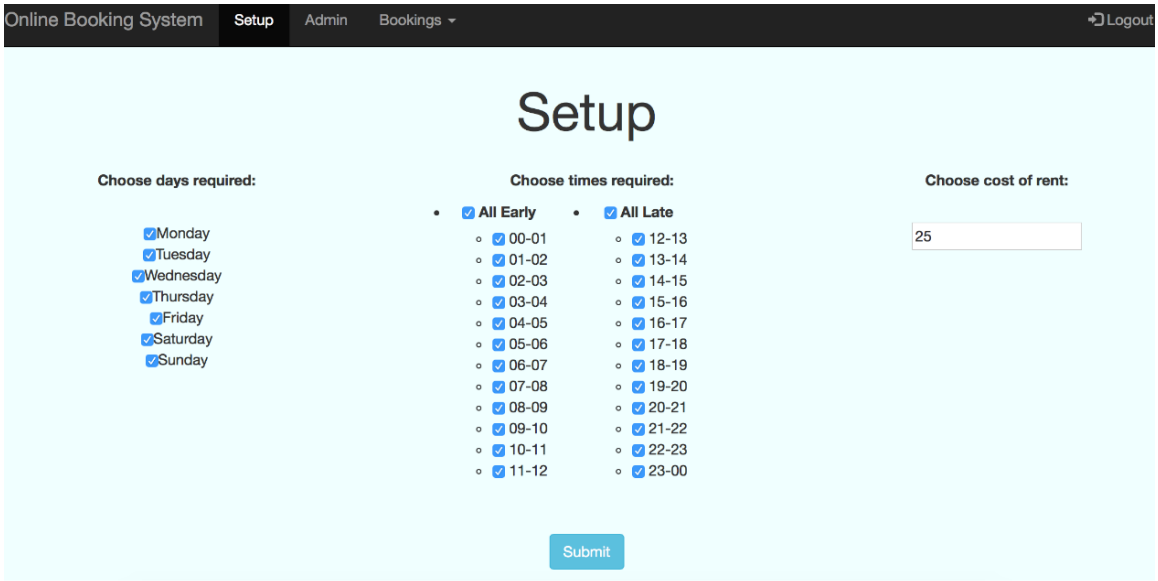
Test SP7: Re-click the ‘all late’ button.



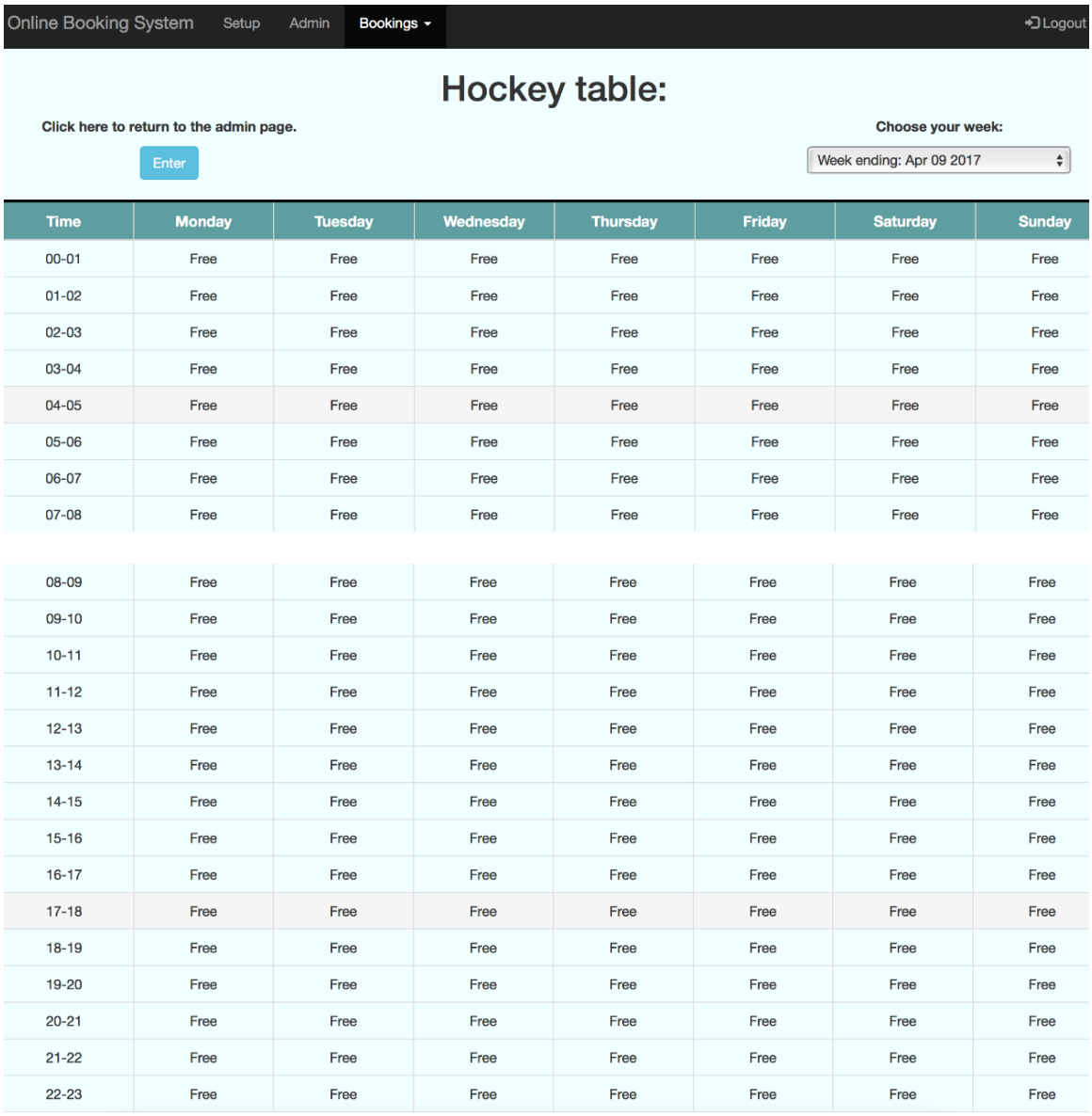
Result: The system unchecks each item in the list.



Test SP8: Complete all days and times on the setup page.

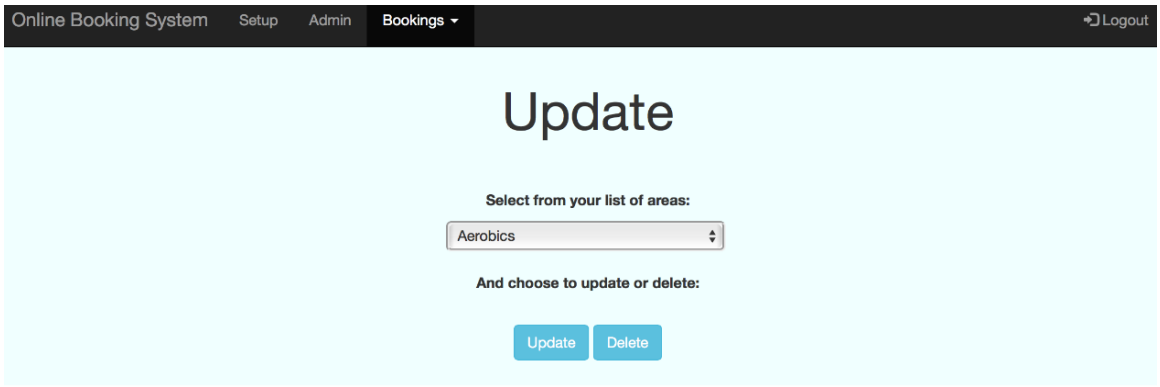


Result: The system creates the table with all options.

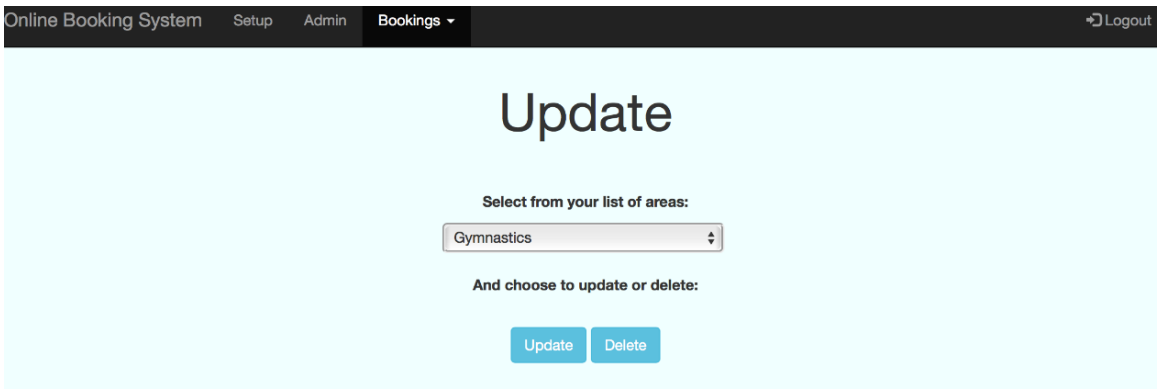

Online Booking System
Setup
Admin
Bookings
Logout
Hockey table:
Click here to return to the admin page.
Enter
Choose your week:
Week ending: Apr 09 2017
Time
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
00-01 Free Free Free Free Free Free Free
01-02 Free Free Free Free Free Free Free
02-03 Free Free Free Free Free Free Free
03-04 Free Free Free Free Free Free Free
04-05 Free Free Free Free Free Free Free
05-06 Free Free Free Free Free Free Free
06-07 Free Free Free Free Free Free Free
07-08 Free Free Free Free Free Free Free
08-09 Free Free Free Free Free Free Free
09-10 Free Free Free Free Free Free Free
10-11 Free Free Free Free Free Free Free
11-12 Free Free Free Free Free Free Free
12-13 Free Free Free Free Free Free Free
13-14 Free Free Free Free Free Free Free
14-15 Free Free Free Free Free Free Free
15-16 Free Free Free Free Free Free Free
16-17 Free Free Free Free Free Free Free
17-18 Free Free Free Free Free Free Free
18-19 Free Free Free Free Free Free Free
19-20 Free Free Free Free Free Free Free
20-21 Free Free Free Free Free Free Free
21-22 Free Free Free Free Free Free Free
22-23 Free Free Free Free Free Free Free

2.8.5 Update Bookings

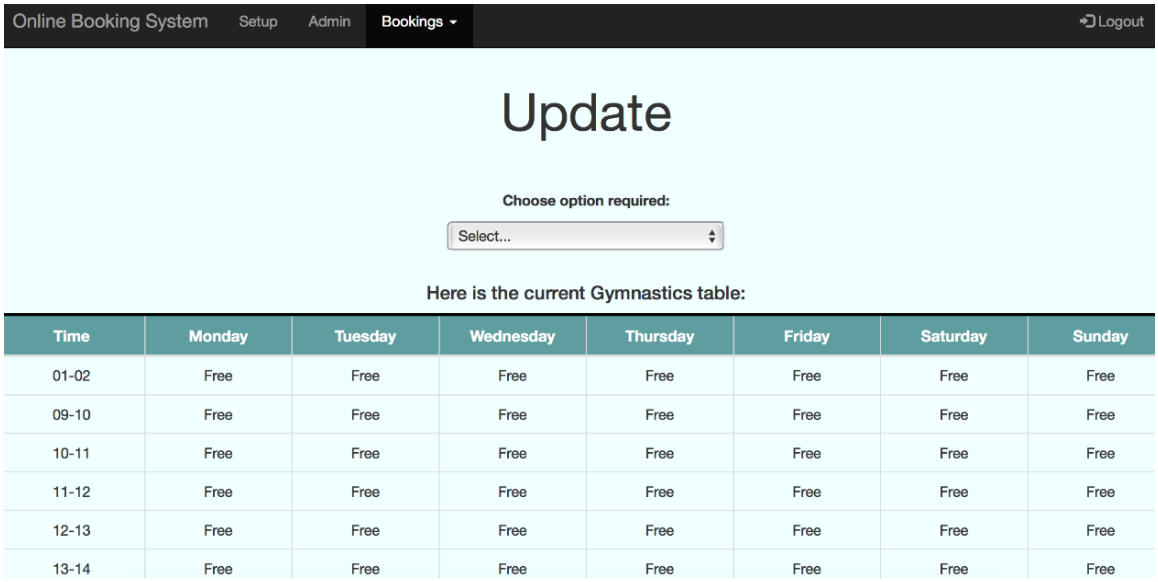
The update bookings page.



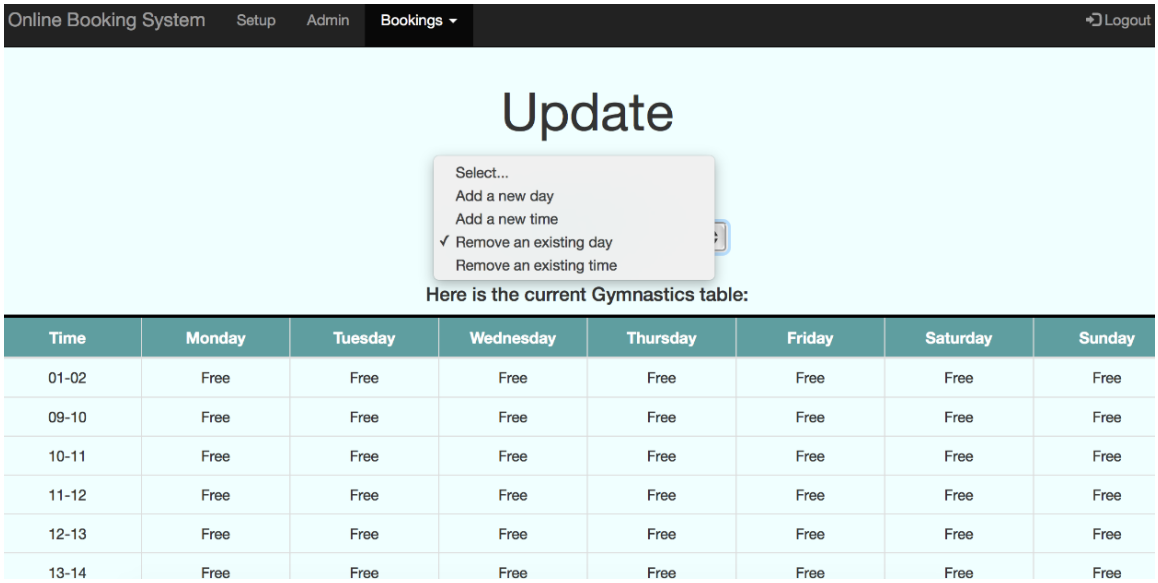
Test UB1: Choose an area from the dropdown list.



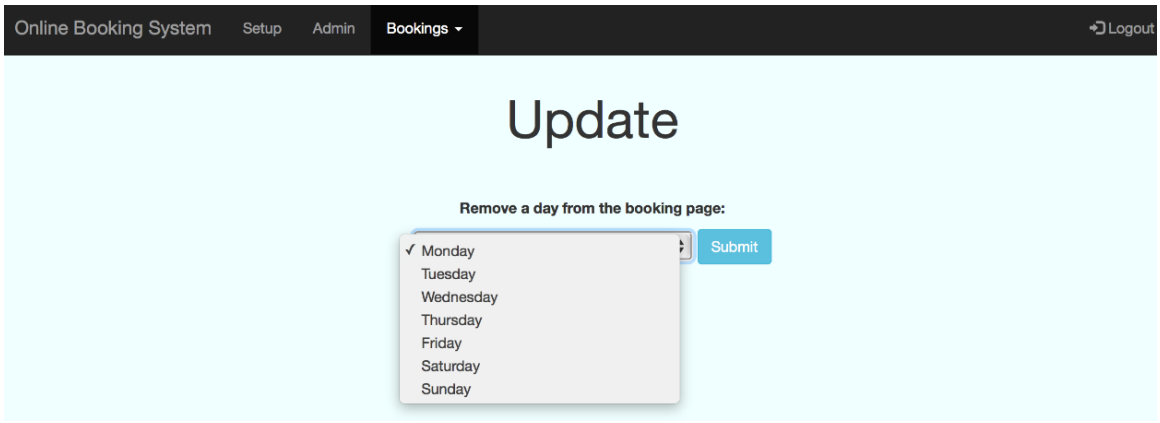
Result: The system displays the table with a list of possible options.



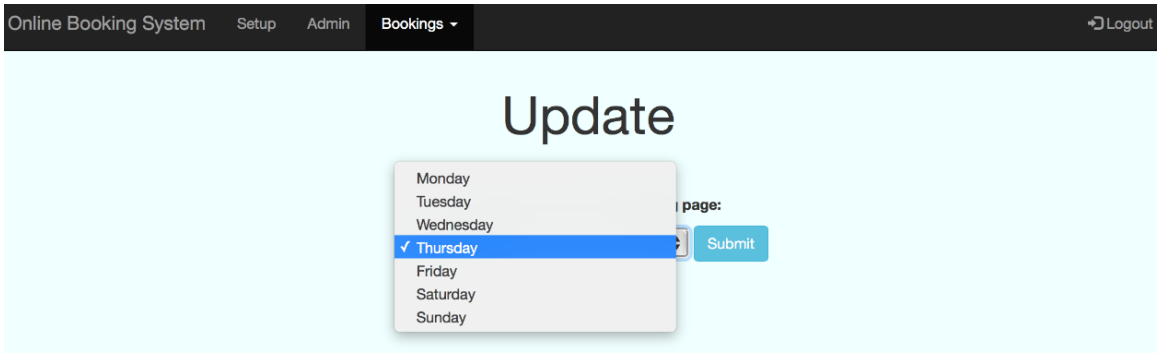
Test UB2: Choose option from dropdown menu.



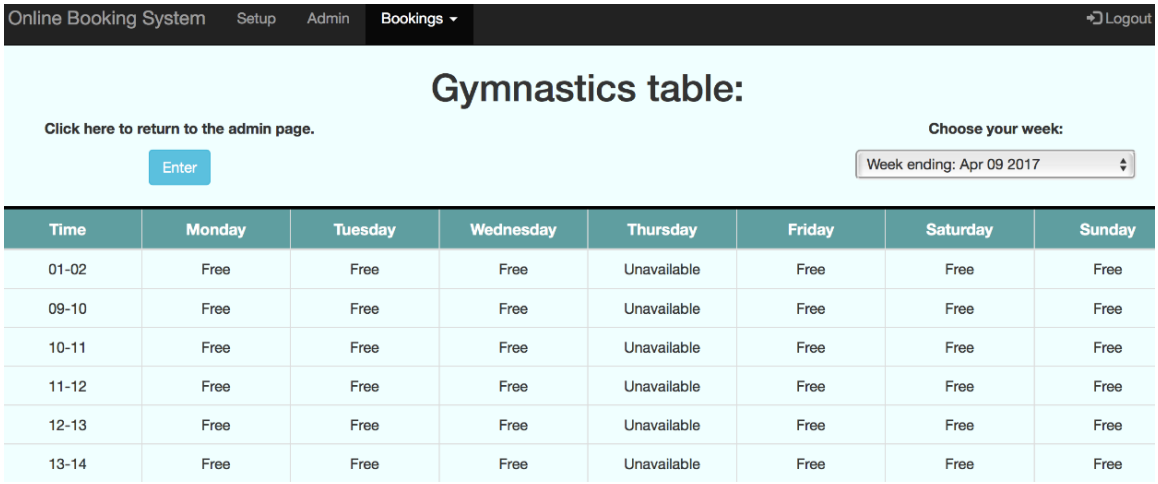
Result: The system provides a list of days available to remove.



Test UB3: Choose a day from the dropdown list (Thursday).

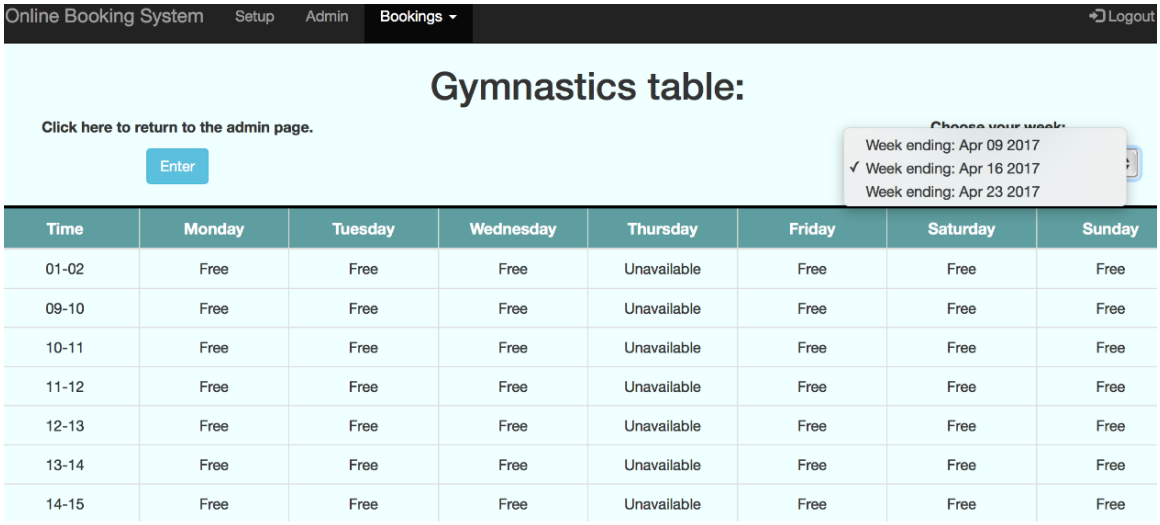


Result: The system displays the table without the selected day.



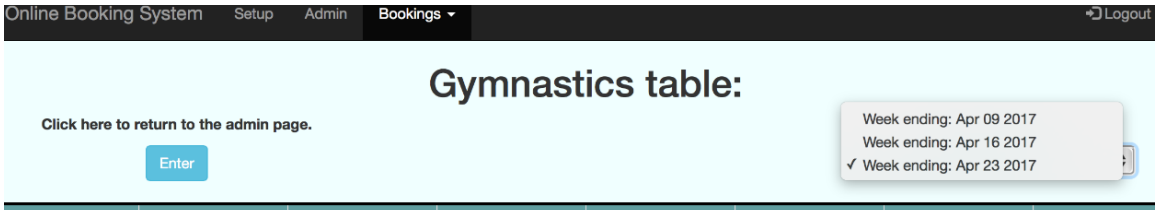
Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
01-02	Free	Free	Free	Unavailable	Free	Free	Free
09-10	Free	Free	Free	Unavailable	Free	Free	Free
10-11	Free	Free	Free	Unavailable	Free	Free	Free
11-12	Free	Free	Free	Unavailable	Free	Free	Free
12-13	Free	Free	Free	Unavailable	Free	Free	Free
13-14	Free	Free	Free	Unavailable	Free	Free	Free

Test UB4: Change the week from the list on right hand side.



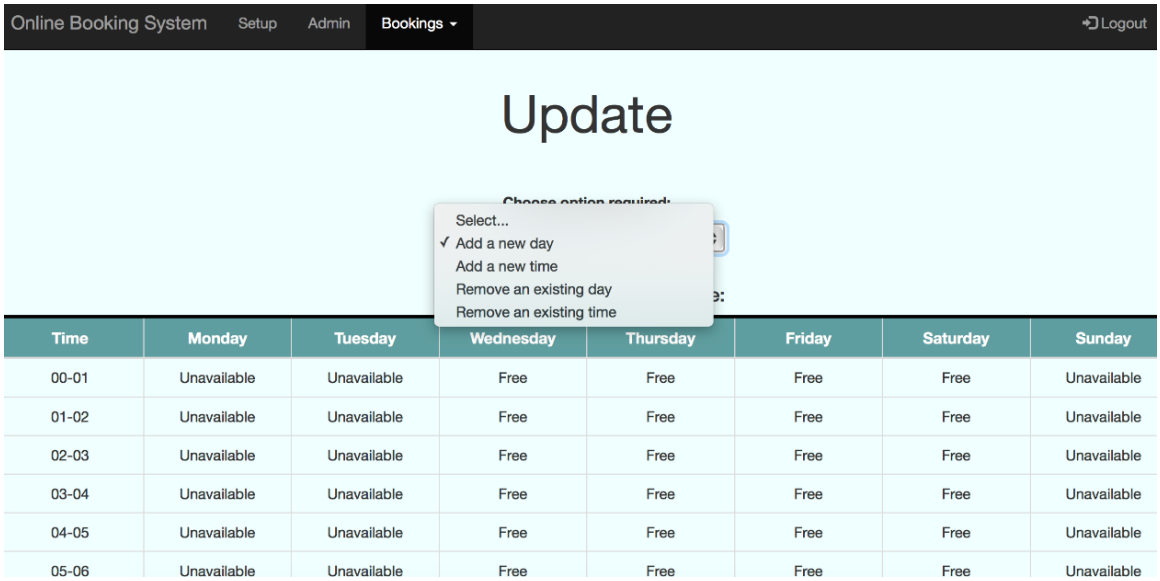
Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
01-02	Free	Free	Free	Unavailable	Free	Free	Free
09-10	Free	Free	Free	Unavailable	Free	Free	Free
10-11	Free	Free	Free	Unavailable	Free	Free	Free
11-12	Free	Free	Free	Unavailable	Free	Free	Free
12-13	Free	Free	Free	Unavailable	Free	Free	Free
13-14	Free	Free	Free	Unavailable	Free	Free	Free
14-15	Free	Free	Free	Unavailable	Free	Free	Free

Result: The system has removed the day from all weeks.

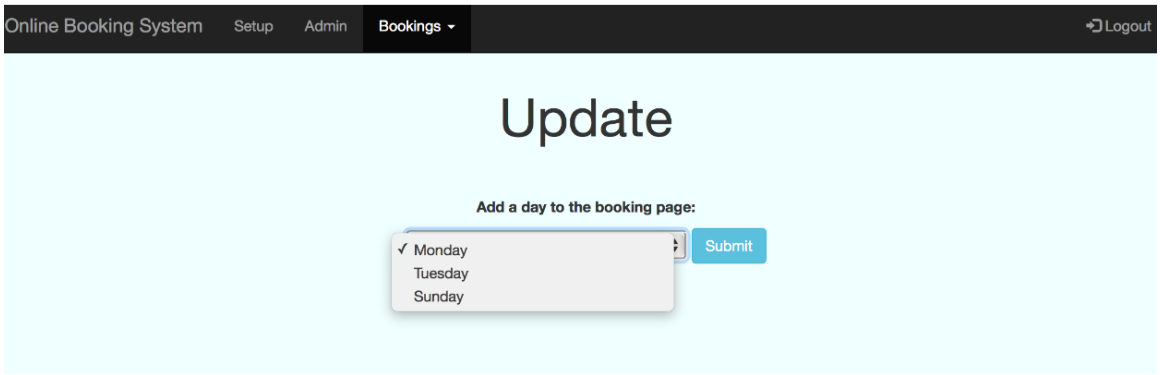


Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
01-02	Free	Free	Free	Unavailable	Free	Free	Free
09-10	Free	Free	Free	Unavailable	Free	Free	Free
10-11	Free	Free	Free	Unavailable	Free	Free	Free
11-12	Free	Free	Free	Unavailable	Free	Free	Free
12-13	Free	Free	Free	Unavailable	Free	Free	Free
13-14	Free	Free	Free	Unavailable	Free	Free	Free
14-15	Free	Free	Free	Unavailable	Free	Free	Free

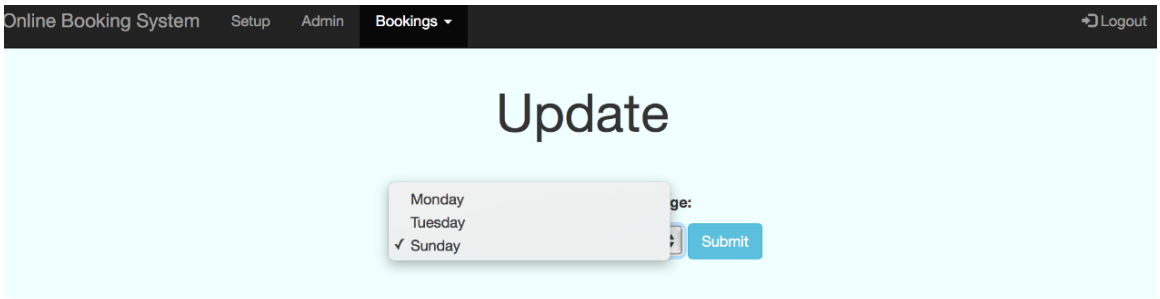
Test UB5: Select to add a day from the dropdown menu.



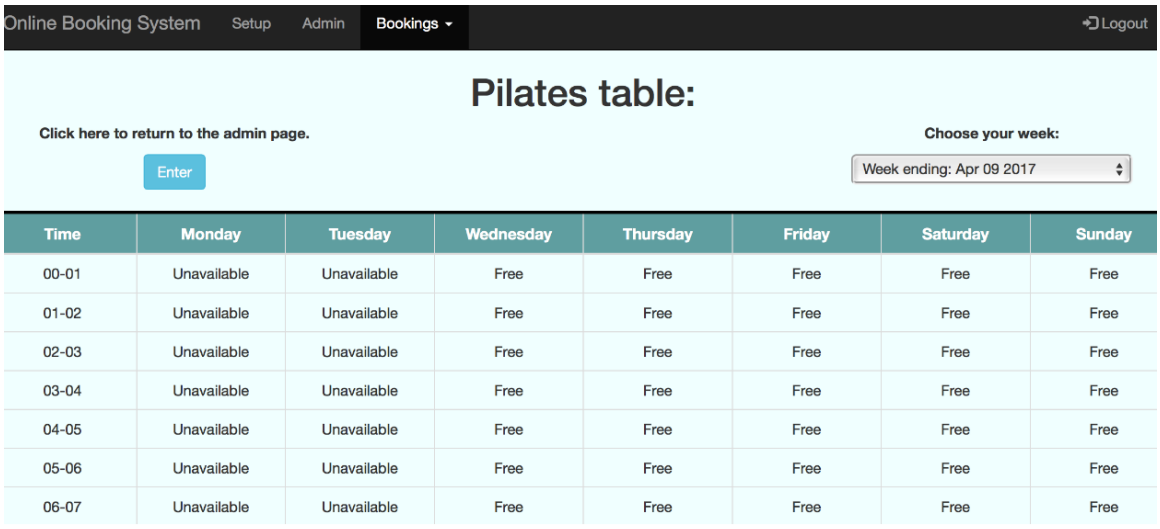
Result: The system provides a list of days that can be added.



Test UB6: Choose a day from the list.

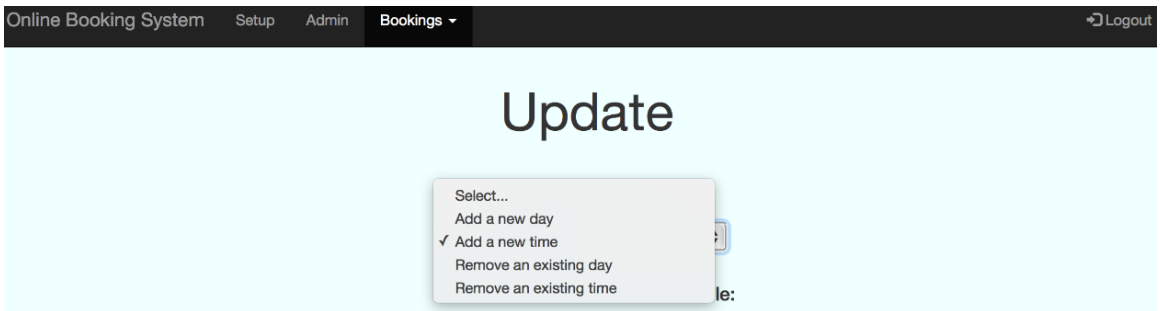


Result: The system includes the chosen day on the table as ‘Free’.



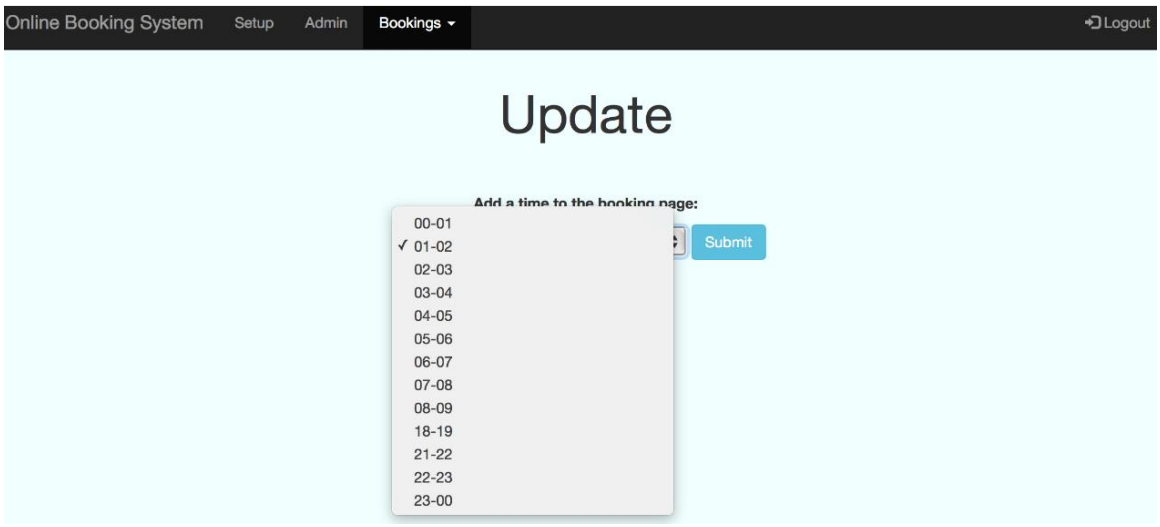
Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
00-01	Unavailable	Unavailable	Free	Free	Free	Free	Free
01-02	Unavailable	Unavailable	Free	Free	Free	Free	Free
02-03	Unavailable	Unavailable	Free	Free	Free	Free	Free
03-04	Unavailable	Unavailable	Free	Free	Free	Free	Free
04-05	Unavailable	Unavailable	Free	Free	Free	Free	Free
05-06	Unavailable	Unavailable	Free	Free	Free	Free	Free
06-07	Unavailable	Unavailable	Free	Free	Free	Free	Free

Test UB7: Choose to add a time to a table.

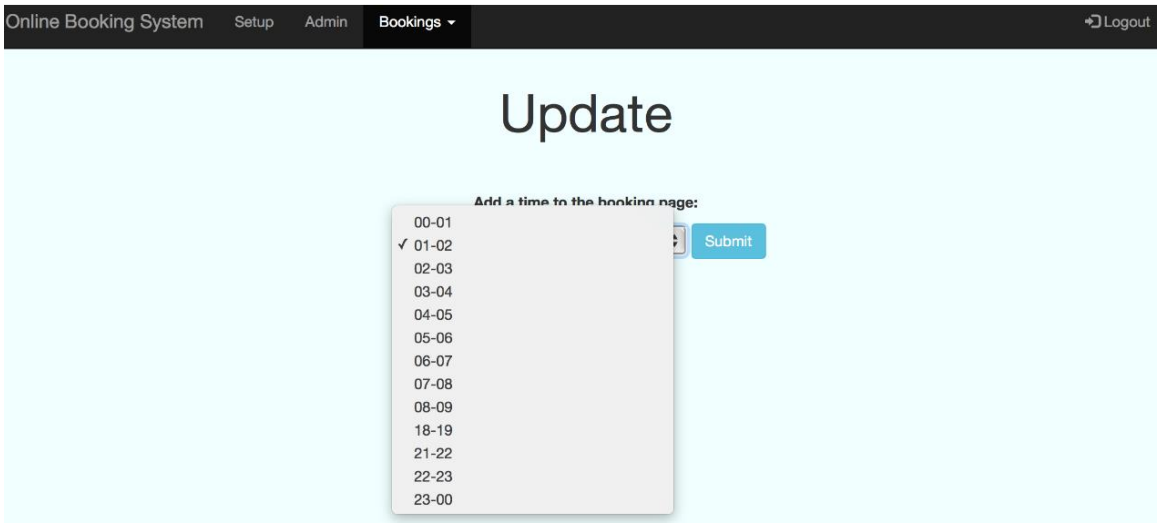


Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
09-10	Free	Free	Free	Free	Free	Free	Free
10-11	Free	Free	Free	Free	Free	Free	Free
11-12	Free	Free	Free	Free	Free	Free	Free
12-13	Free	Free	Free	Free	Free	Free	Free
13-14	Free	Free	Free	Free	Free	Free	Free

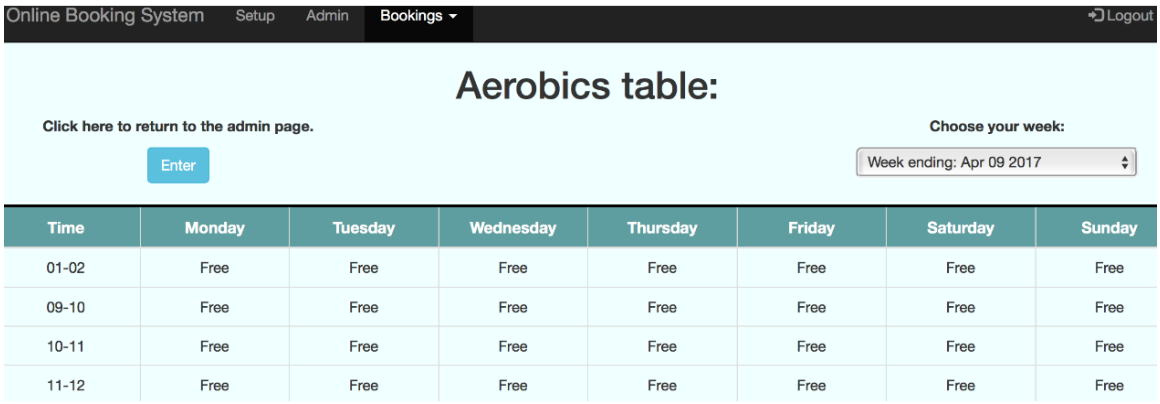
Result: The system displays a list of times that can be added.



Test UB8: Choose a time from the list.

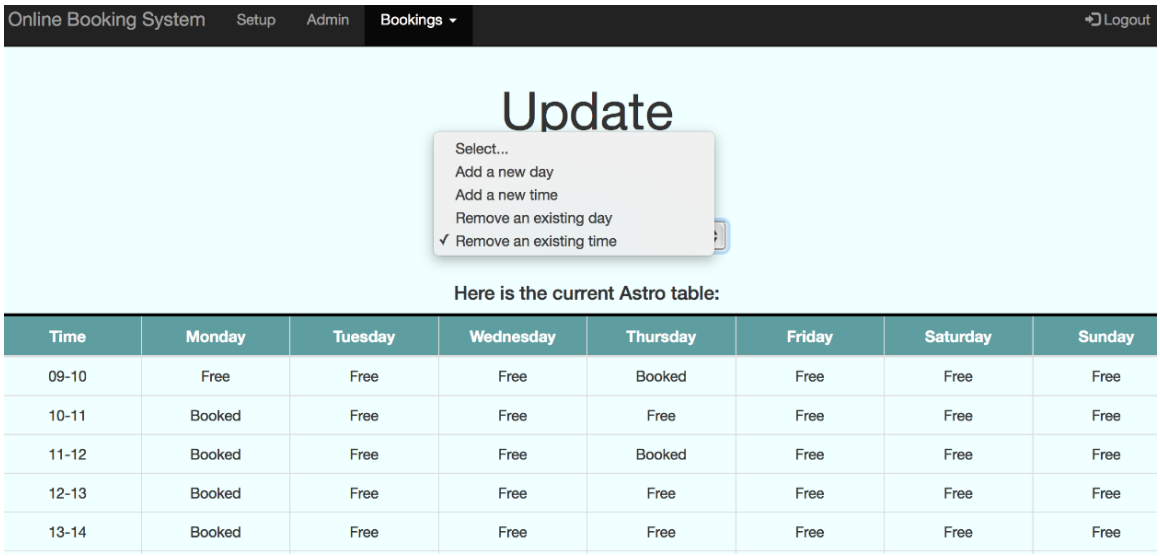


Result: The system updates the table to include the chosen time.



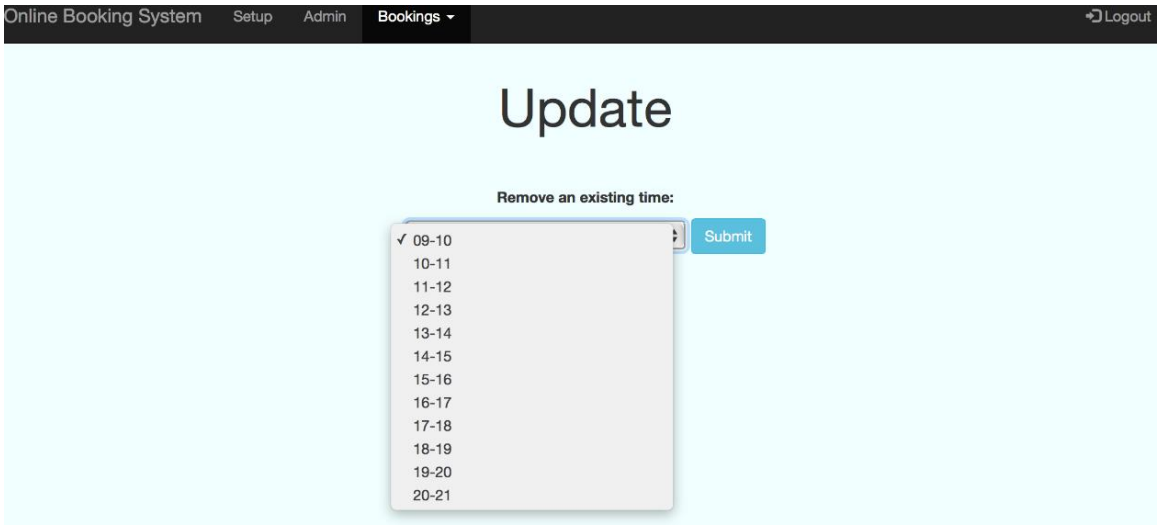
Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
01-02	Free	Free	Free	Free	Free	Free	Free
09-10	Free	Free	Free	Free	Free	Free	Free
10-11	Free	Free	Free	Free	Free	Free	Free
11-12	Free	Free	Free	Free	Free	Free	Free

Test UB9: Choose to remove a time from the list.

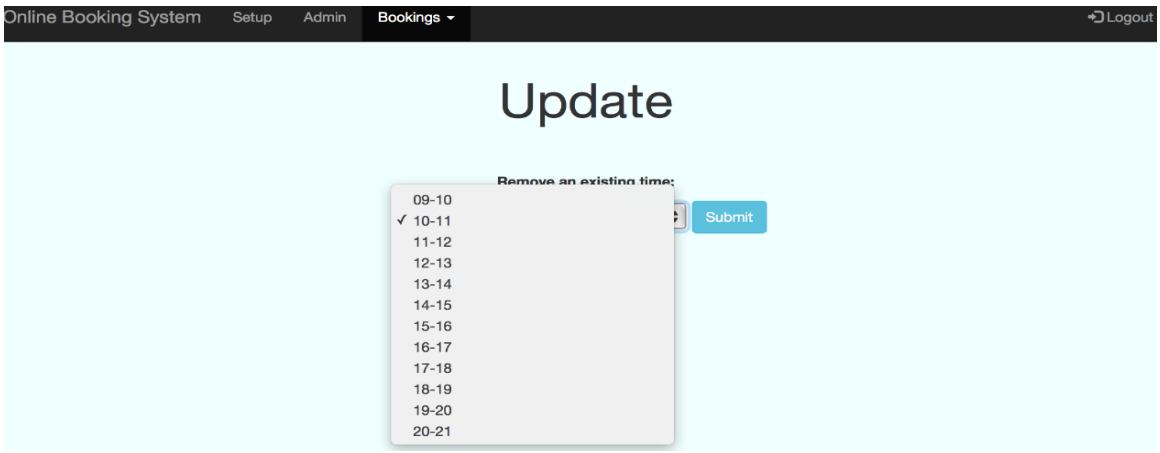


Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
09-10	Free	Free	Free	Booked	Free	Free	Free
10-11	Booked	Free	Free	Free	Free	Free	Free
11-12	Booked	Free	Free	Booked	Free	Free	Free
12-13	Booked	Free	Free	Free	Free	Free	Free
13-14	Booked	Free	Free	Free	Free	Free	Free

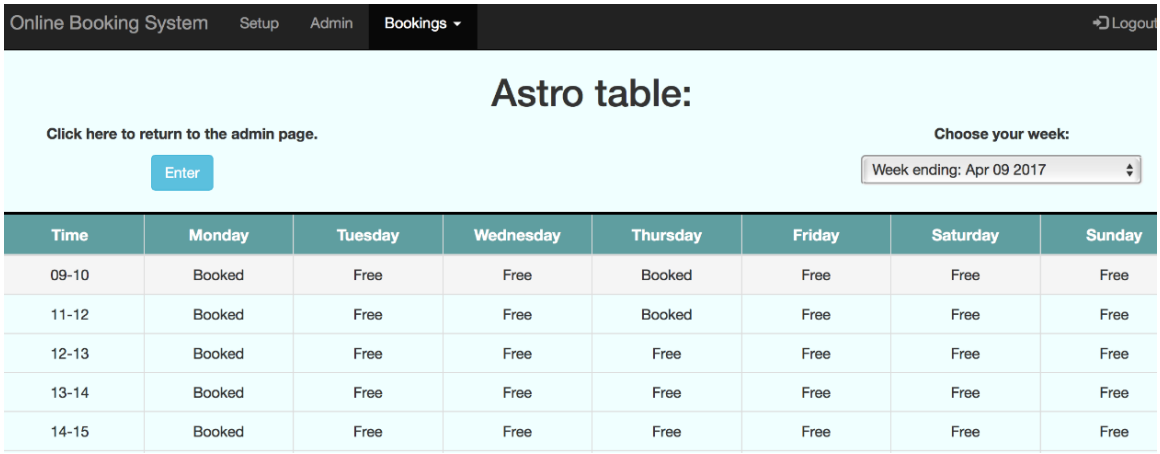
Result: The system displays a list of times that can be removed.



Test UB10: Choose a time to be removed.



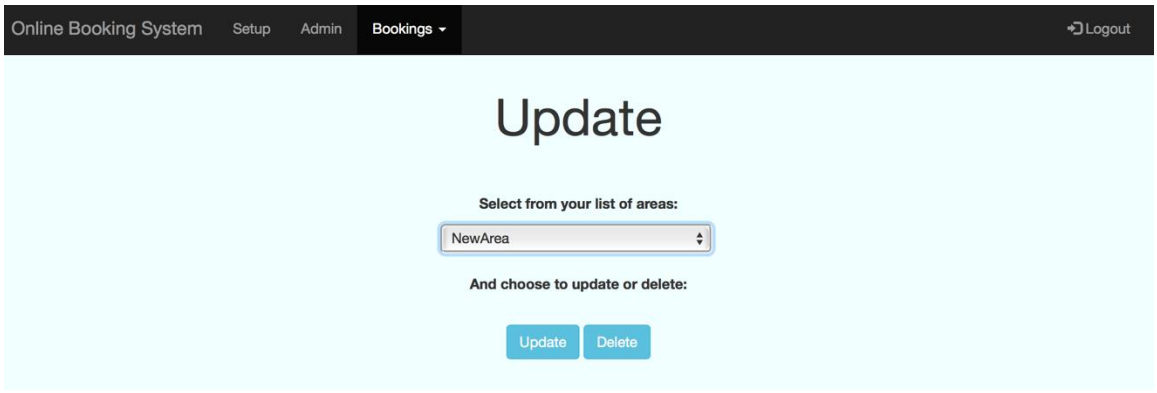
Result: The system removes the time chosen.



Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
09-10	Booked	Free	Free	Booked	Free	Free	Free
11-12	Booked	Free	Free	Booked	Free	Free	Free
12-13	Booked	Free	Free	Free	Free	Free	Free
13-14	Booked	Free	Free	Free	Free	Free	Free
14-15	Booked	Free	Free	Free	Free	Free	Free

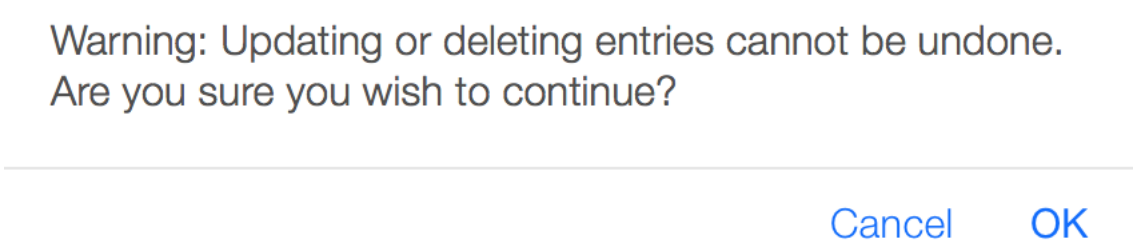
2.8.6 Delete Bookings

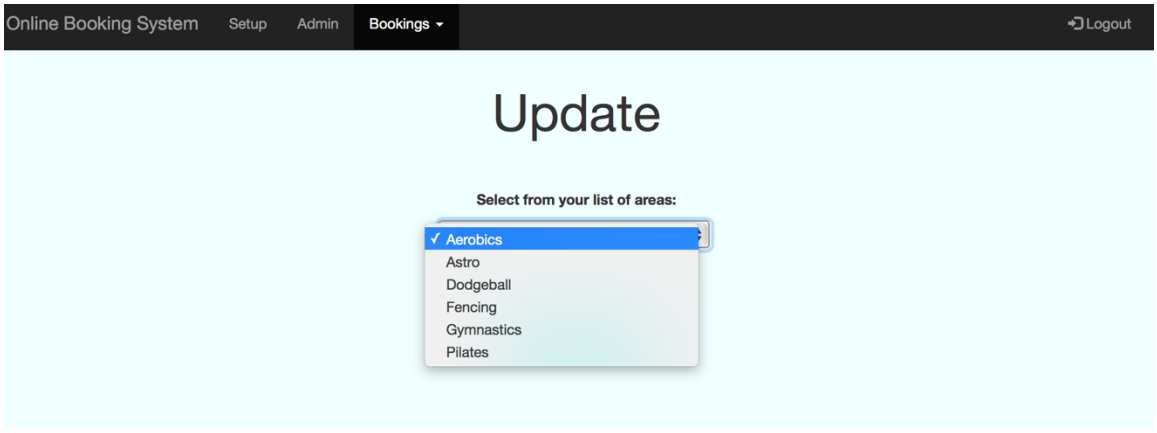
The delete bookings page.



Test DB1: Select an area and click the delete button.

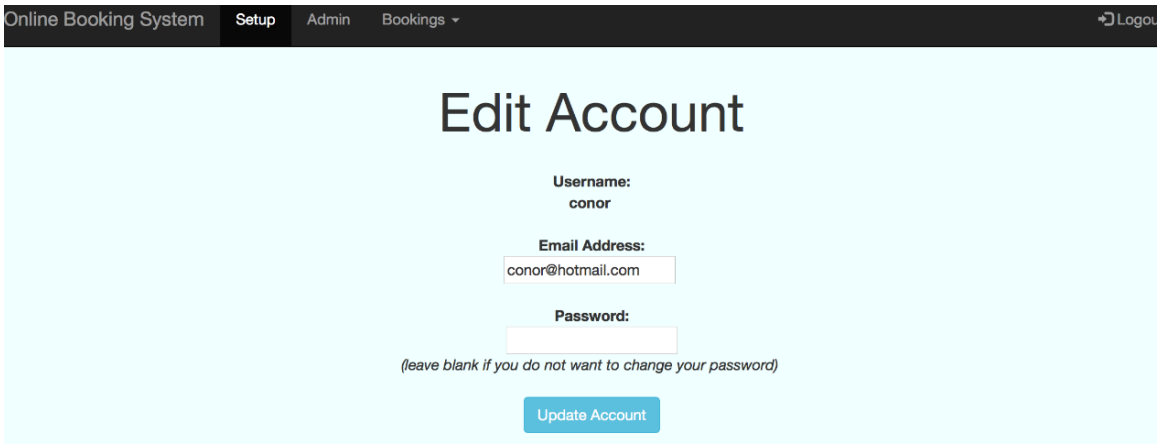
Result: The area has been removed from the list of available areas (Note: a warning message is displayed first).



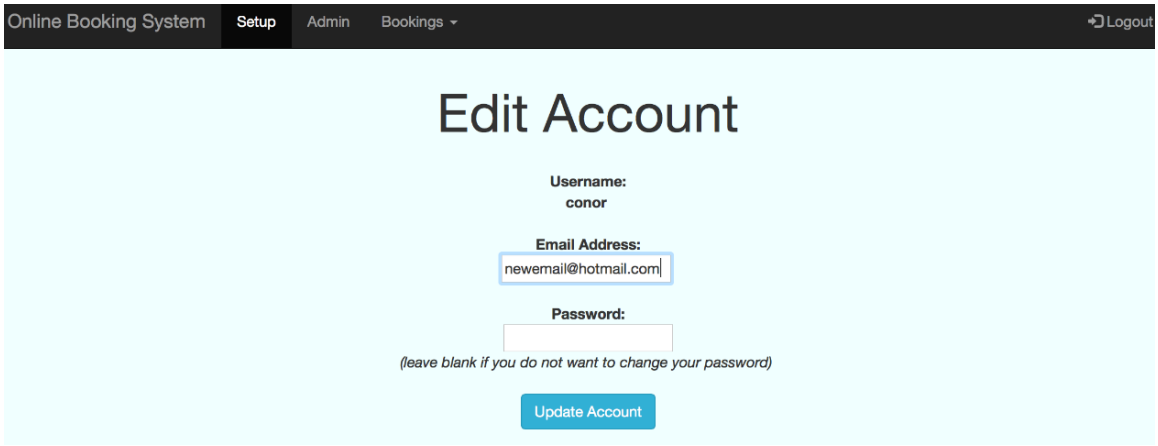


2.8.7 Update Information

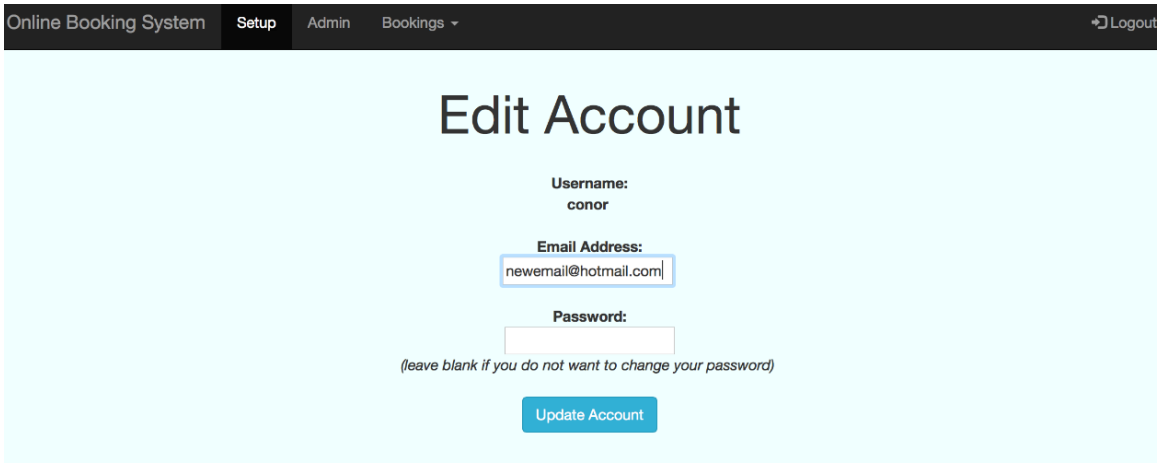
The update information page.



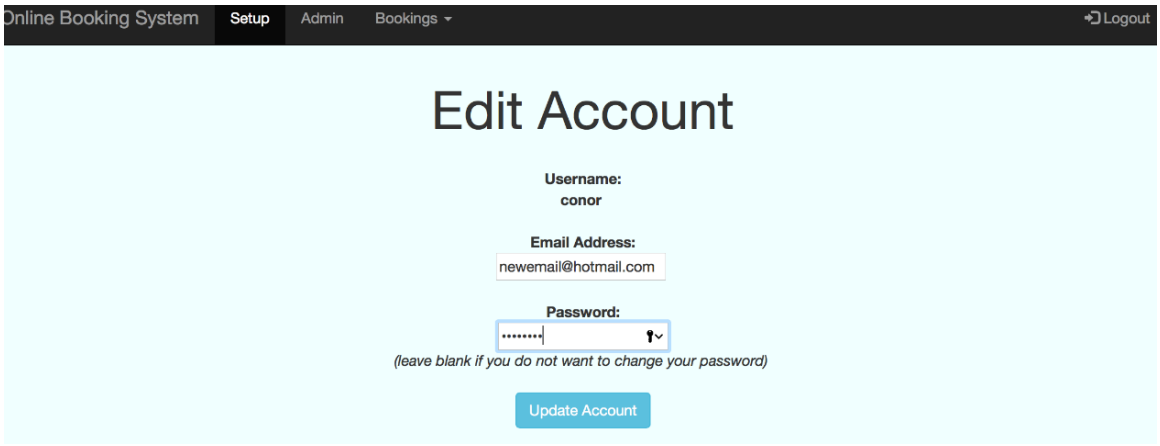
Test UI1: Enter a new email address.



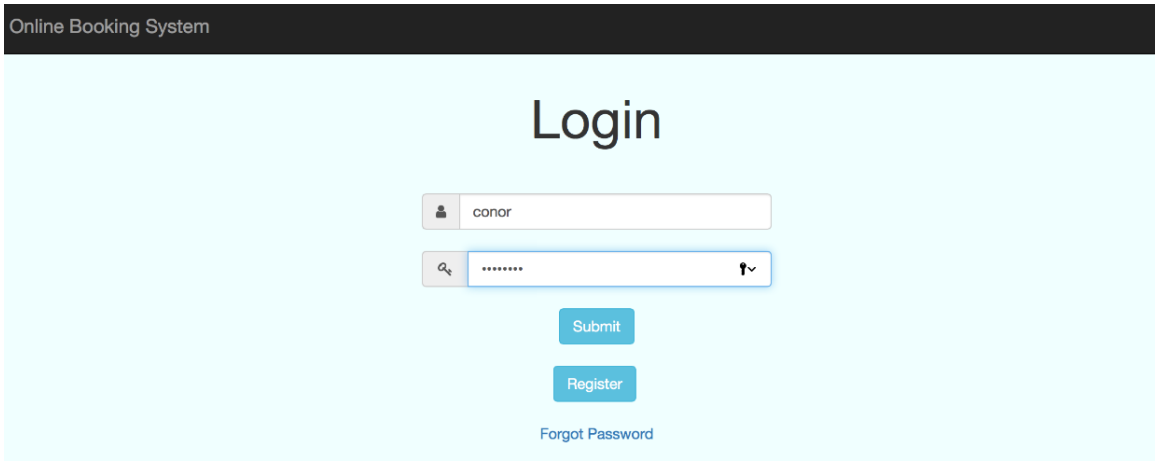
Result: The system updates the email address and displays it when back on the edit account page.

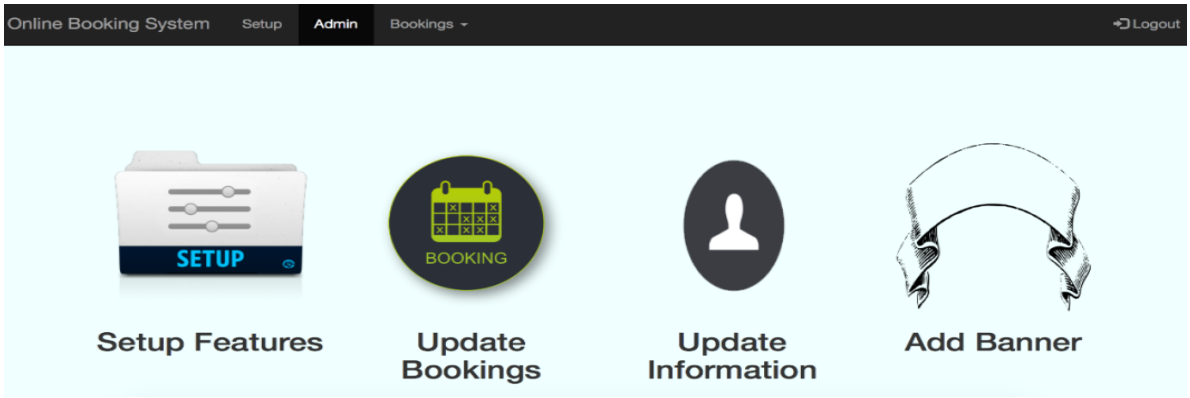


Test UI2: Update the password.

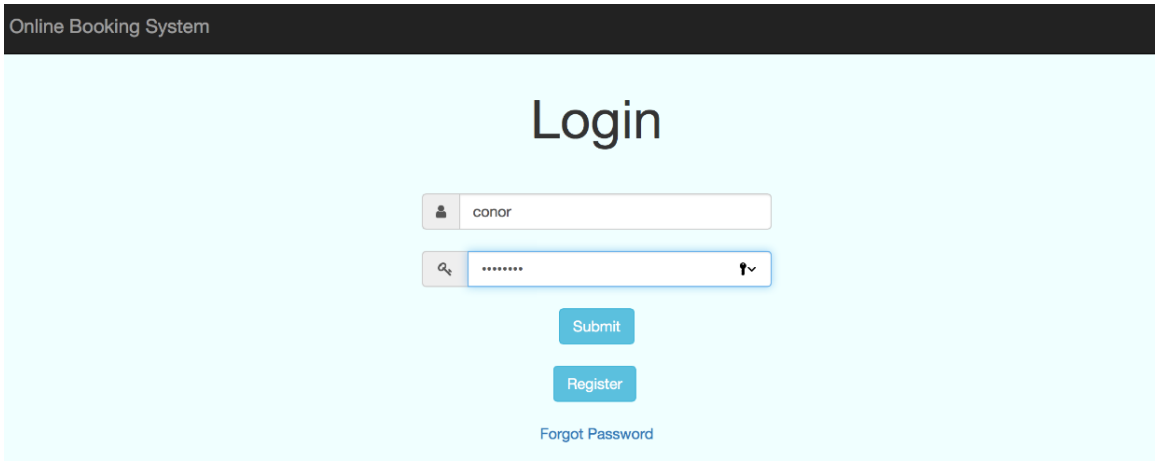


Result: The system updates the password and allows you to login with the new credentials.

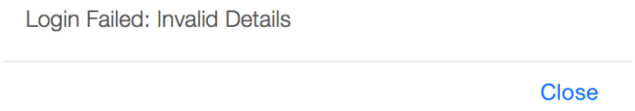




Test UI3: Try log in with the old password.

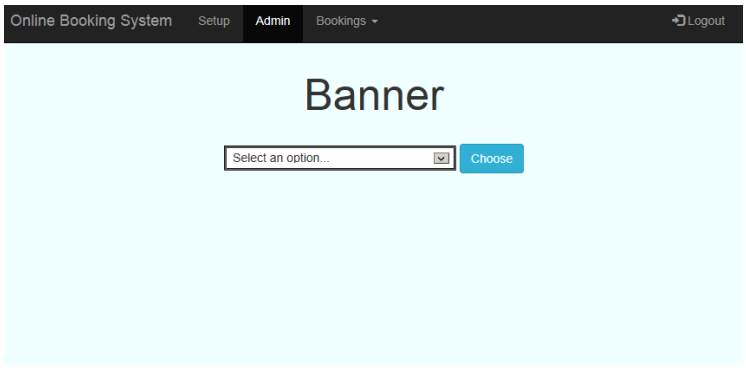


Result: The system rejects the login attempt.

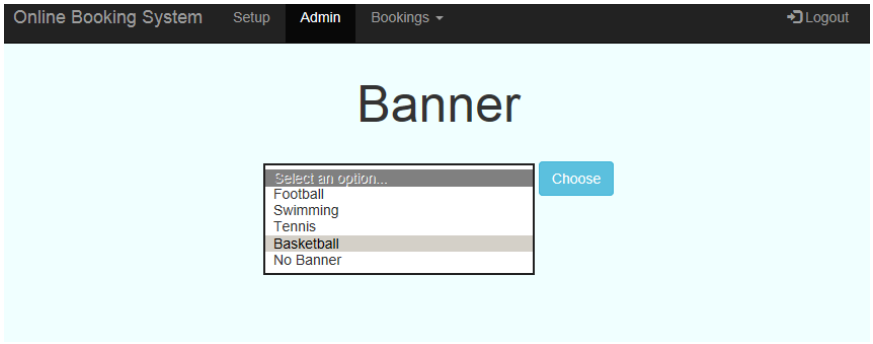


2.8.8 Add Banner

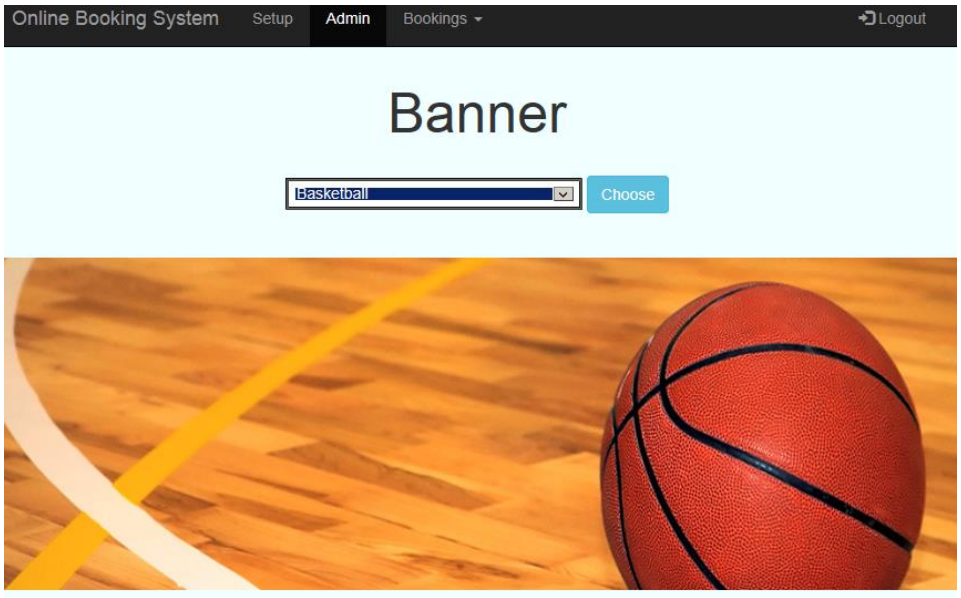
The add banner page.



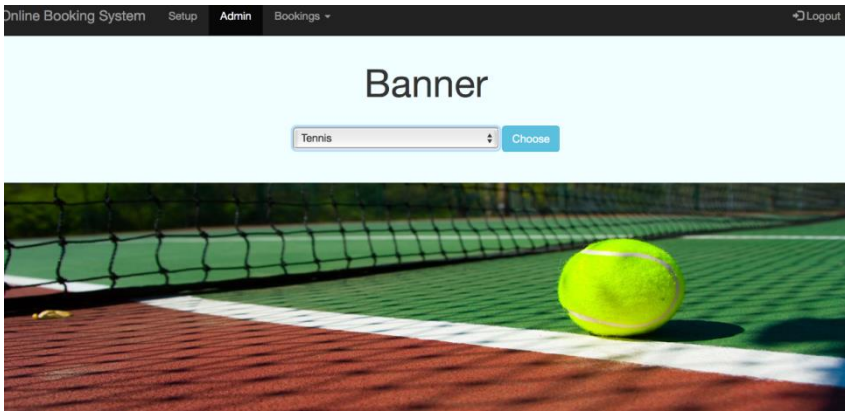
Test AB1: Select an option from the dropdown menu.



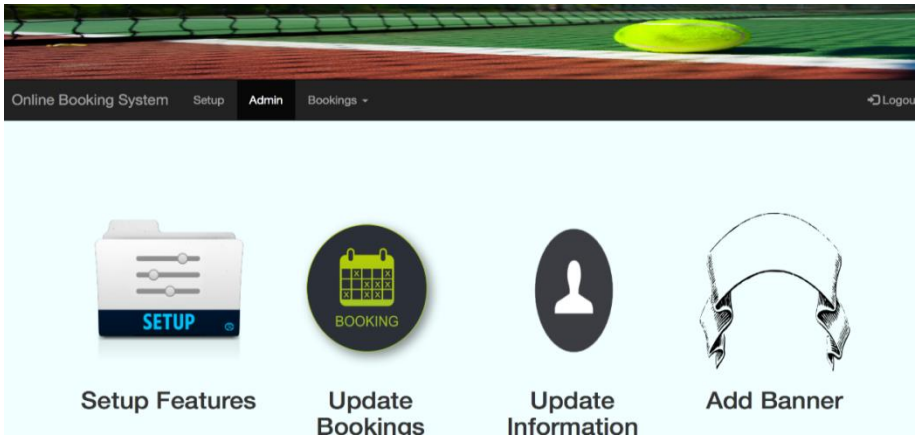
Result: The system displays the option on screen.



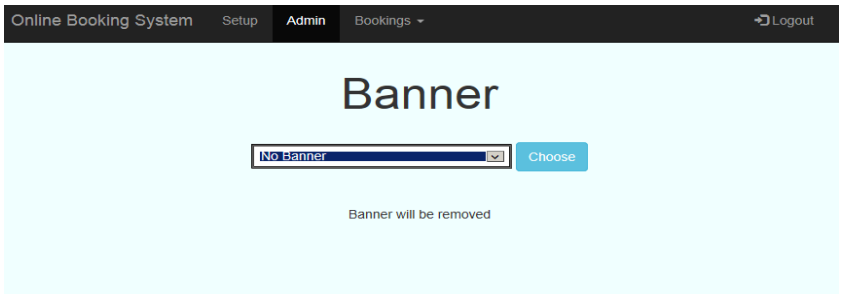
Test AB2: Select an option from the dropdown menu and click ‘Choose’.



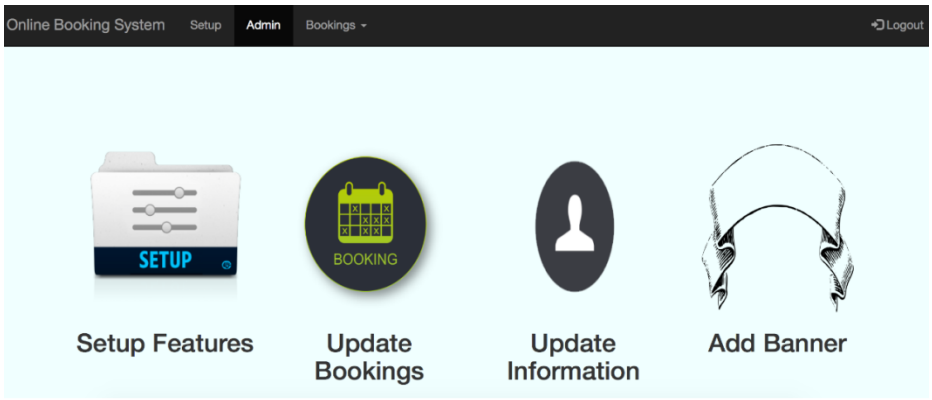
Result: The system returns to the admin page with the chosen banner on display.



Test AB3: Choose the option of ‘No Banner’.

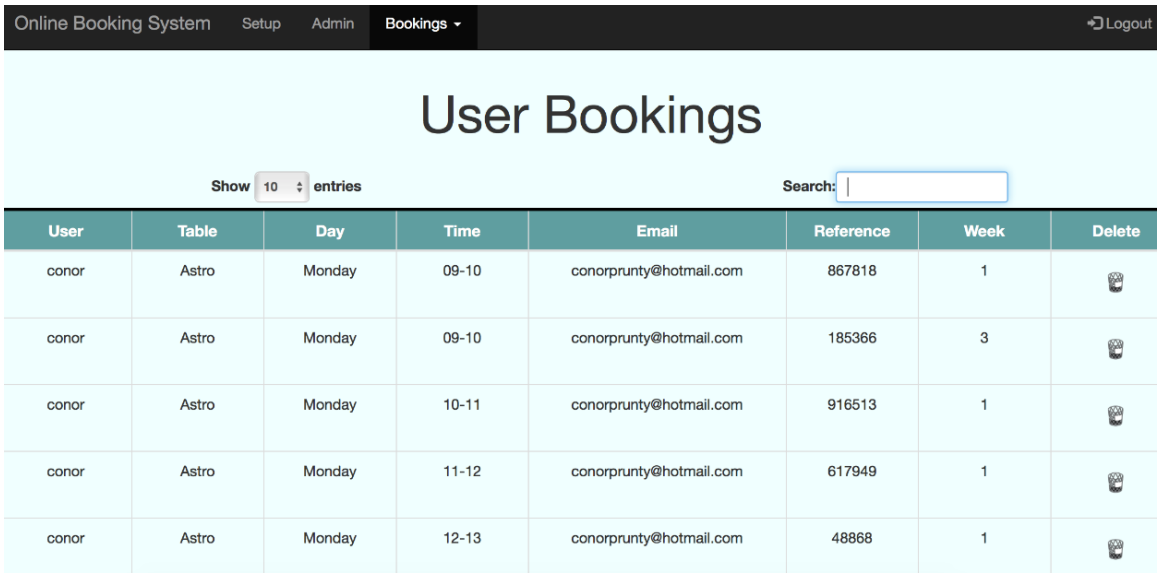


Result: The system returns to the admin page and no banner is displayed.

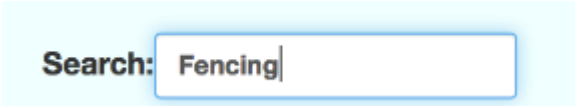


2.8.9 View All Bookings

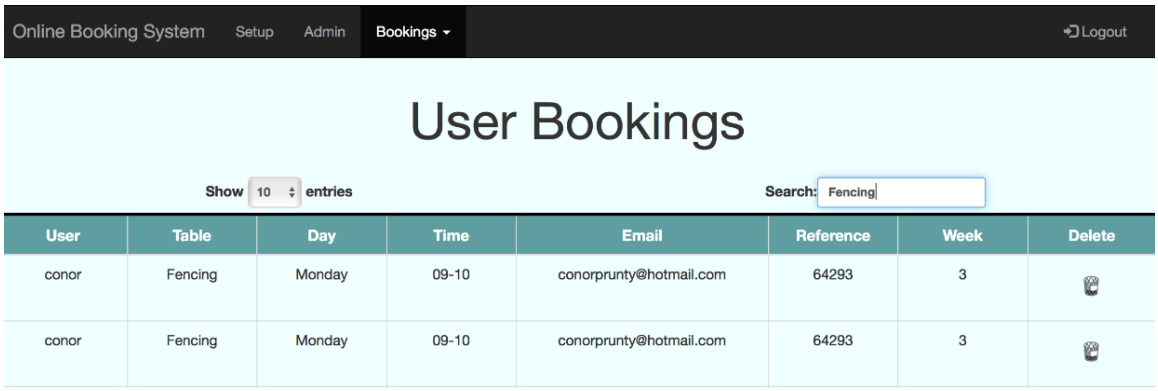
The view all bookings page.



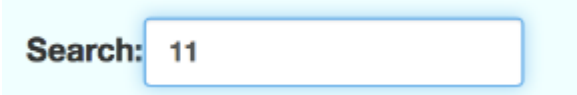
Test VA1: Enter an area in the search field.



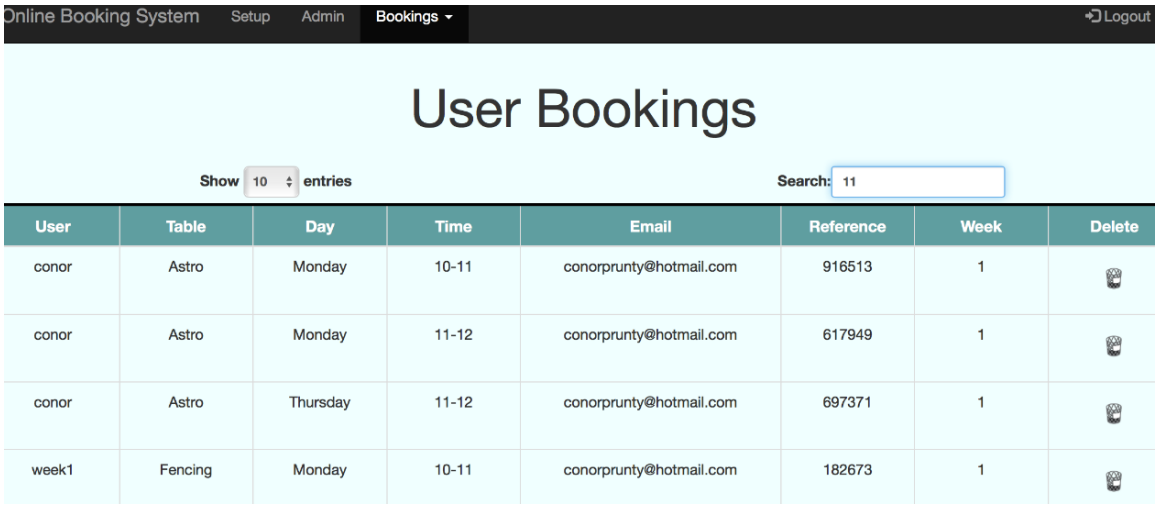
Result: The system displays all results with the searched term.



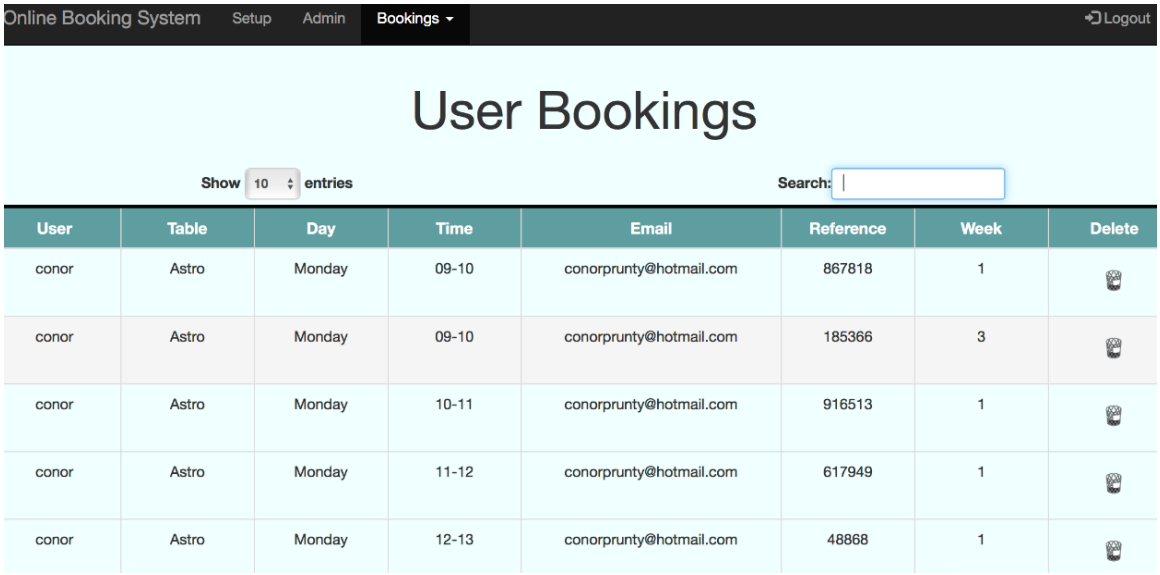
Test VA2: Enter a numeric value.



Result: The system displays the results that include this (this example from the 'time' column).



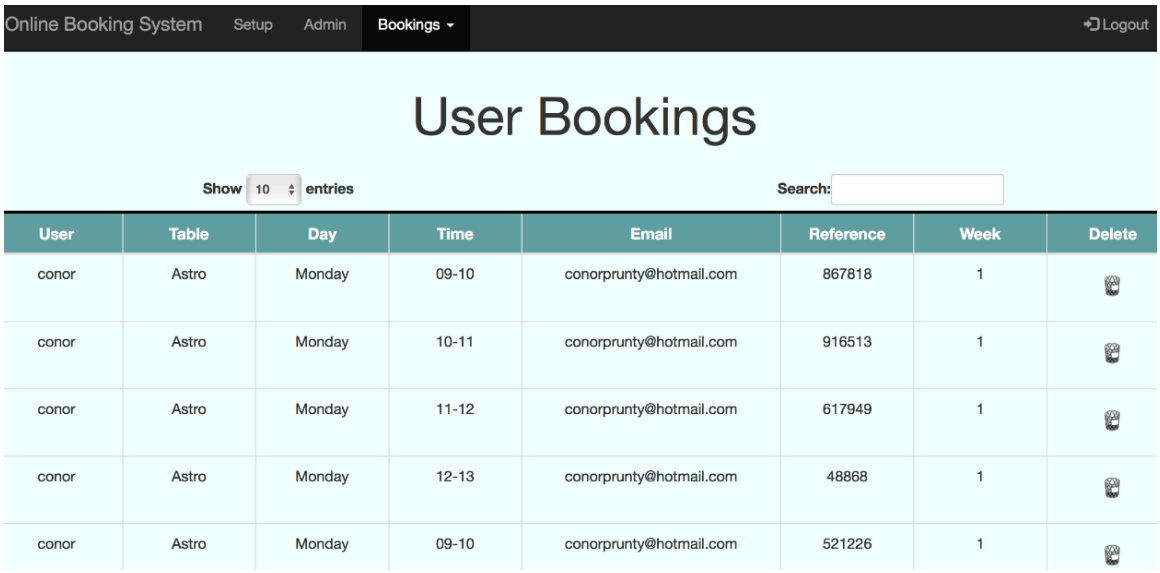
Test VA3: Select an option to delete (row 2).



Result: The system provides a confirmation dialog.

Test VA4: Choose the ‘OK’ button.

Result: The system refreshes the page without the deleted row.



2.9 Customer testing

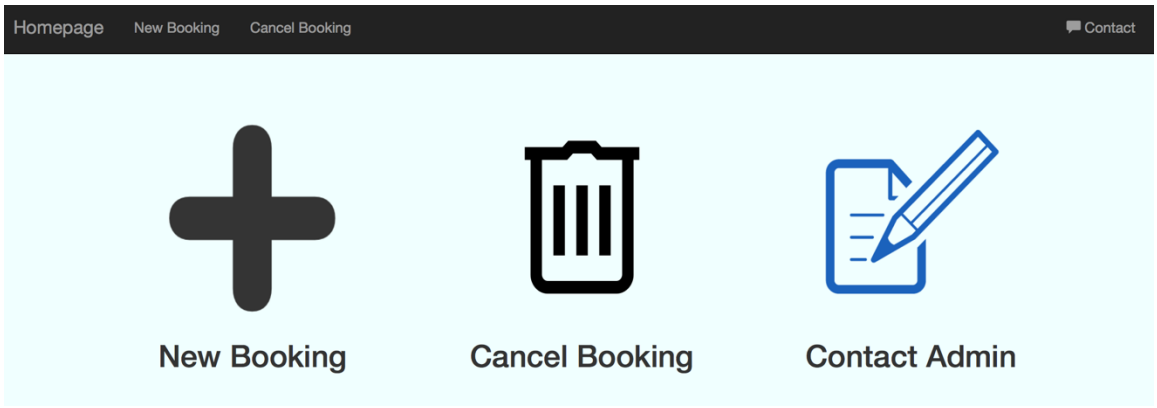
Full testing was completed whilst using the application as an external customer.

The following sections were tested as a customer:

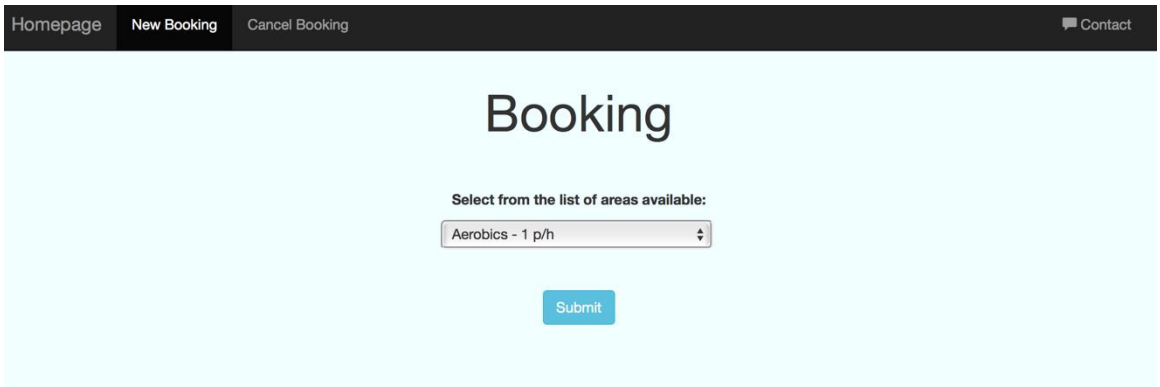
- Homepage
- Navigation Bar
- New Booking
- Cancel Booking
- Contact Form

2.9.1 Homepage

The customer homepage.

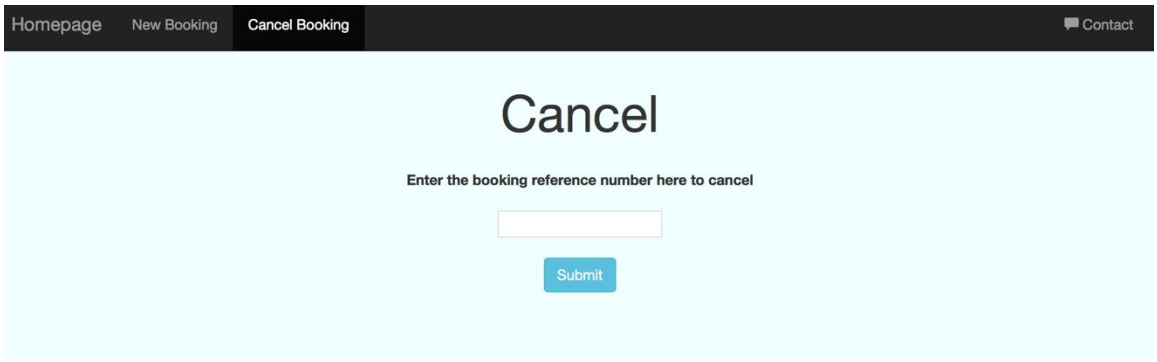


Test CHP1: Select the new booking icon.



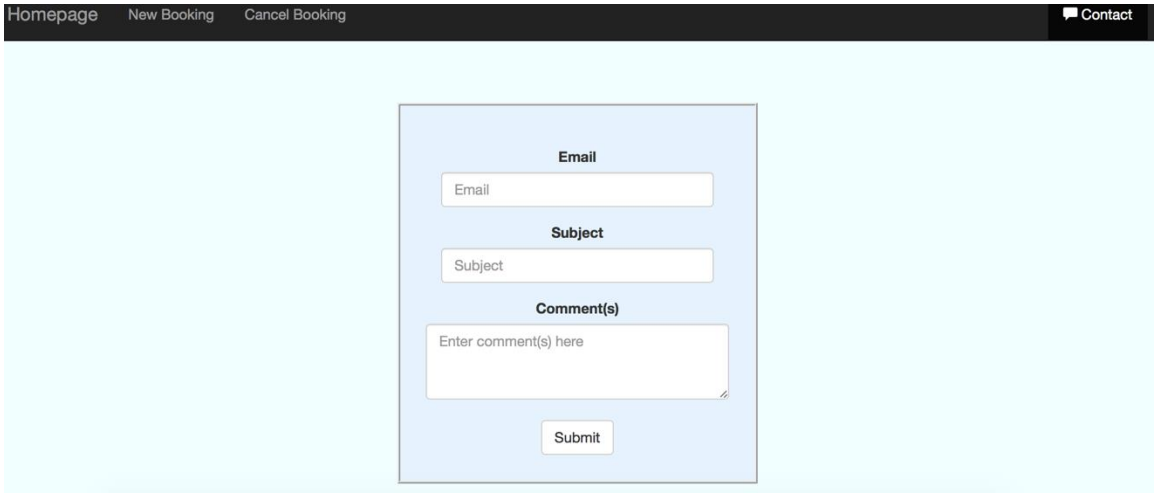
Result: The system brings you to the new booking page.

Test CHP2: Select the cancel booking icon.



Result: The system brings you to the cancel booking page.

Test CHP3: Select the contact admin icon.



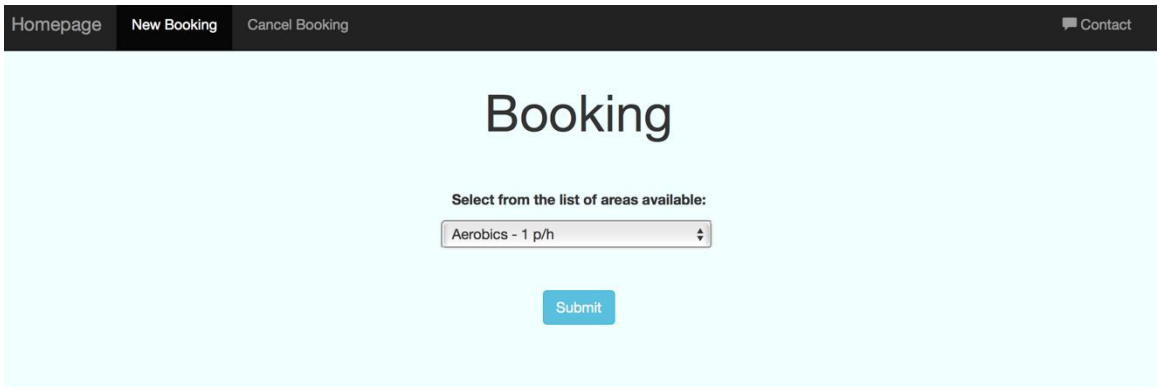
Result: The system brings you to the contact page.

2.9.2 Navigation Bar

The customer navigation bar.

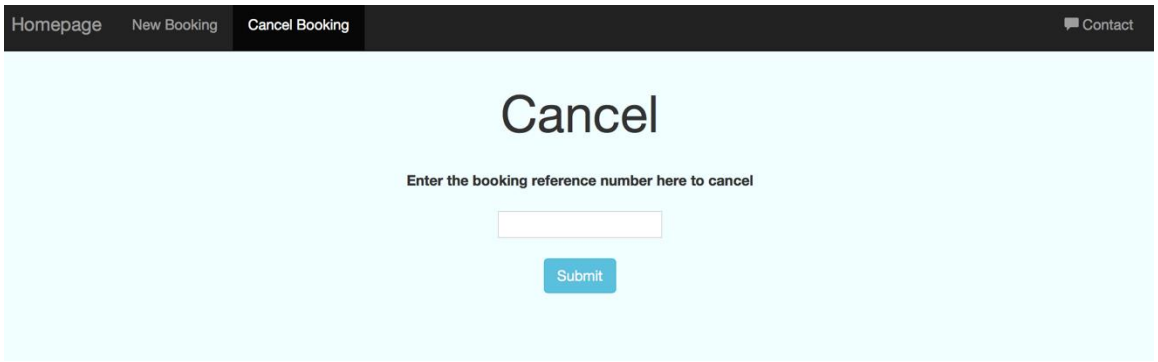
Test CNB1: Select the new booking text.

Result: The system brings you to the new booking page with the navbar highlighting the correct page.



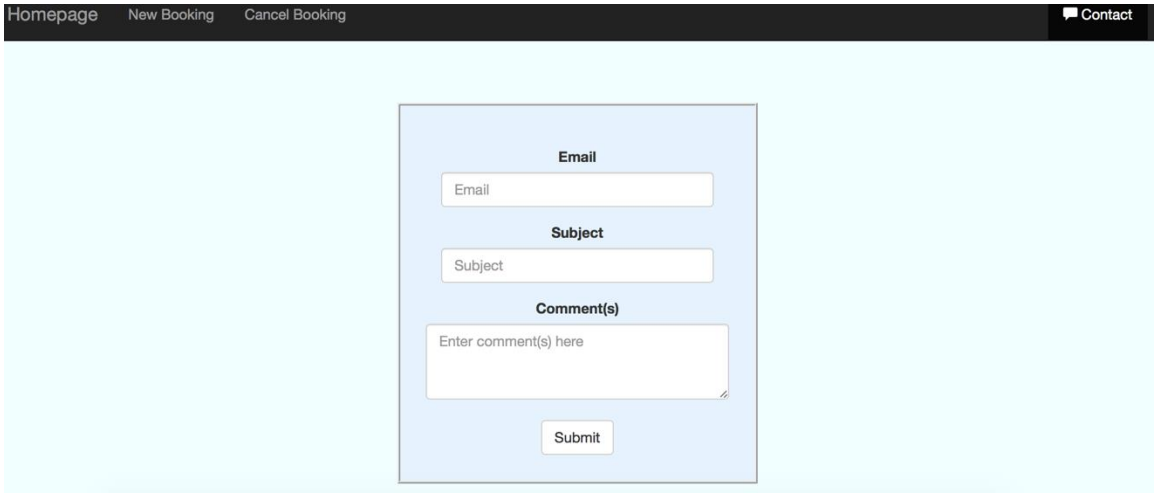
Test CNB2: Select the cancel booking text.

Result: The system brings you to the cancel booking page with the navbar highlighting the correct page.



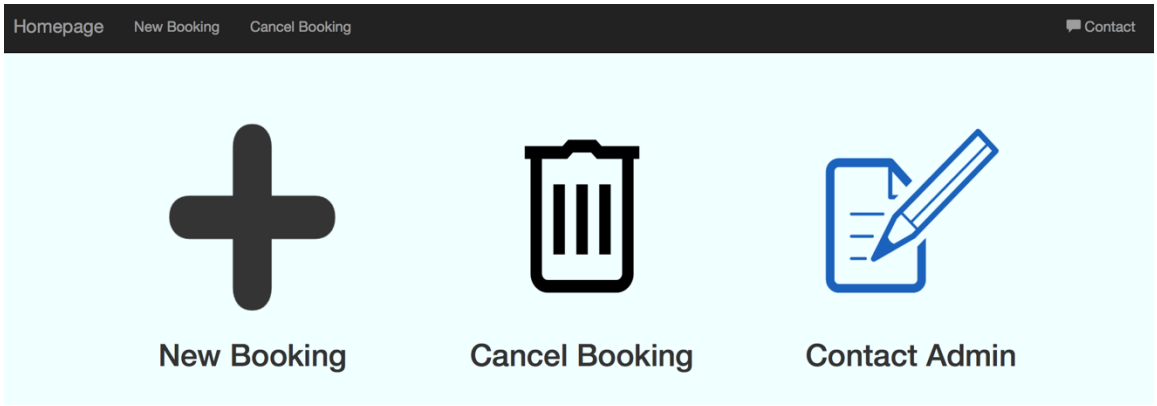
Test CNB3: Select the contact text.

Result: The system brings you to the contact page with the navbar highlighting the correct page.



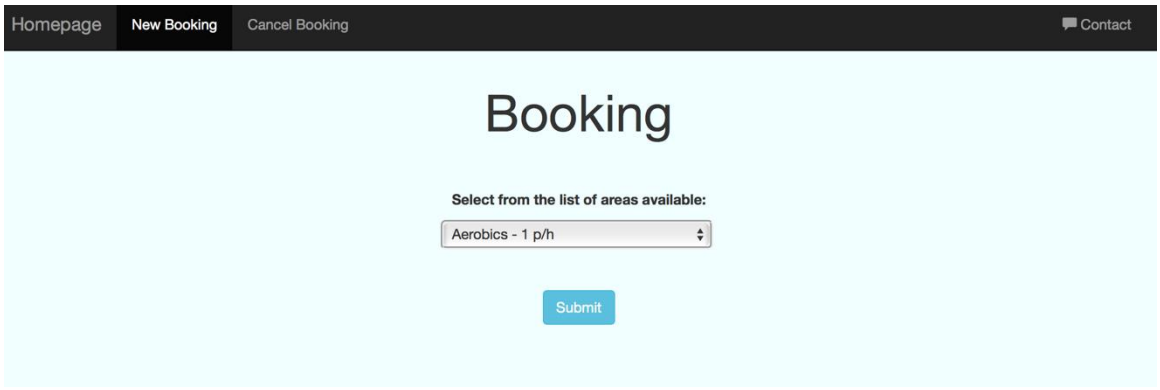
Test CNB4: Select the homepage text.

Result: The system brings you to back to the homepage.

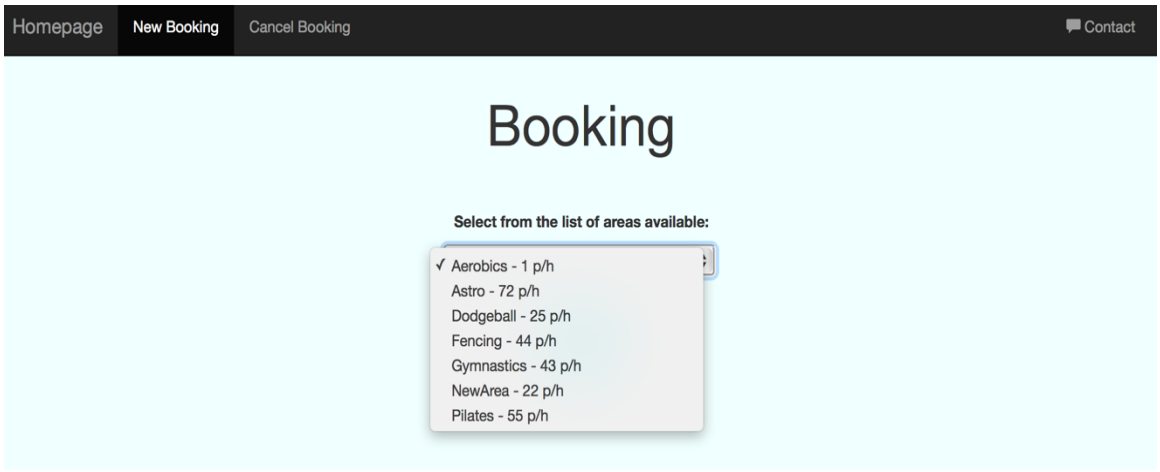


2.9.3 New Booking

The customer new booking page.

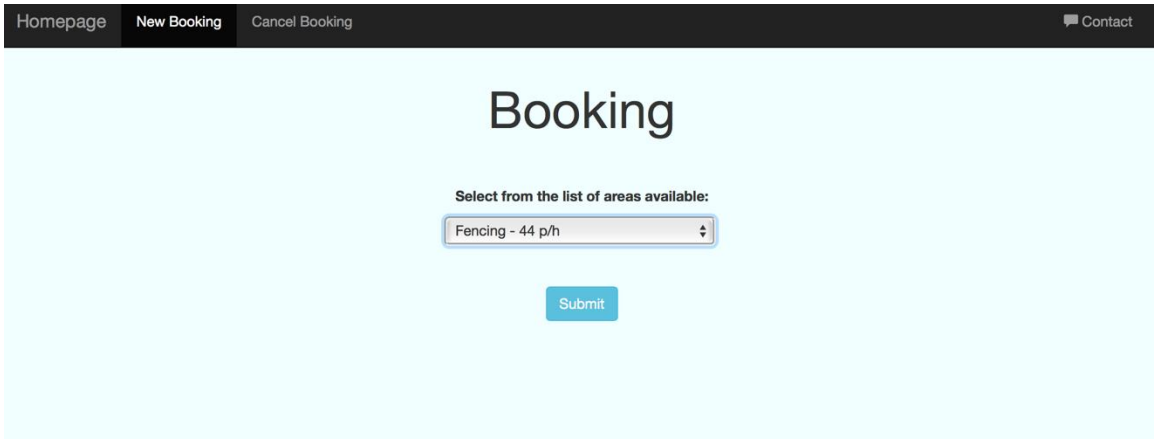


Test CNB1: Click on the dropdown menu.

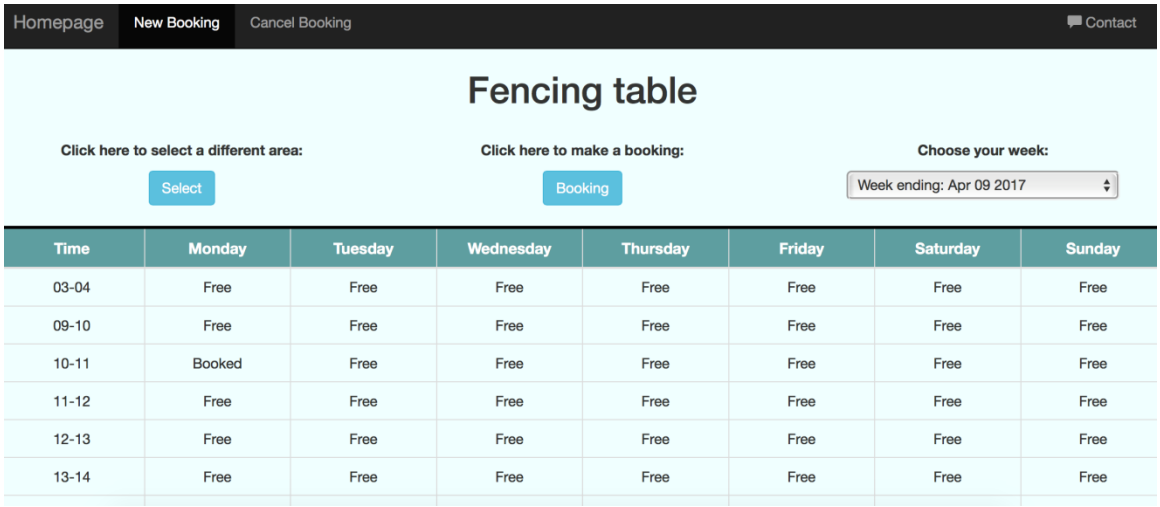


Result: The system displays all the options the club has available, along with the price per hour.

Test CNB2: Choose an option from the dropdown menu.

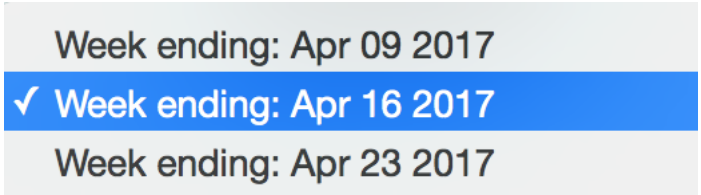


Result: The system shows the table for that area.

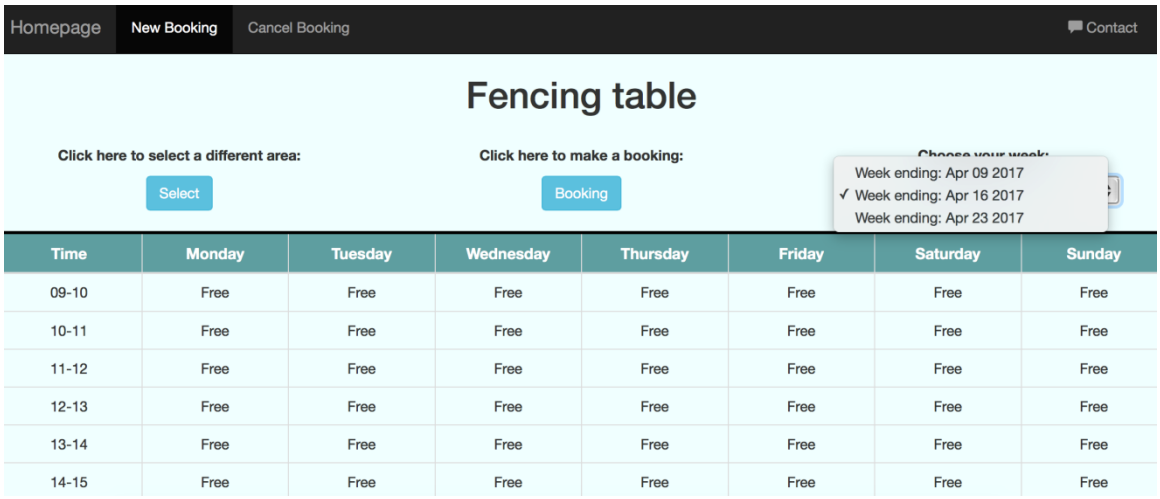


Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
03-04	Free	Free	Free	Free	Free	Free	Free
09-10	Free	Free	Free	Free	Free	Free	Free
10-11	Booked	Free	Free	Free	Free	Free	Free
11-12	Free	Free	Free	Free	Free	Free	Free
12-13	Free	Free	Free	Free	Free	Free	Free
13-14	Free	Free	Free	Free	Free	Free	Free

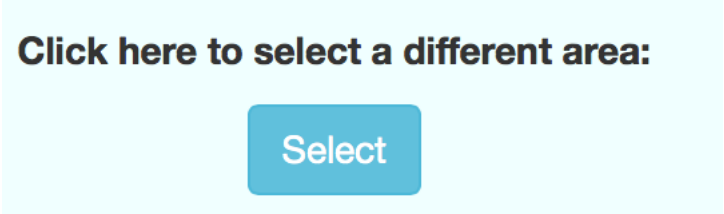
Test CNB3: Choose a different week on the right hand side.



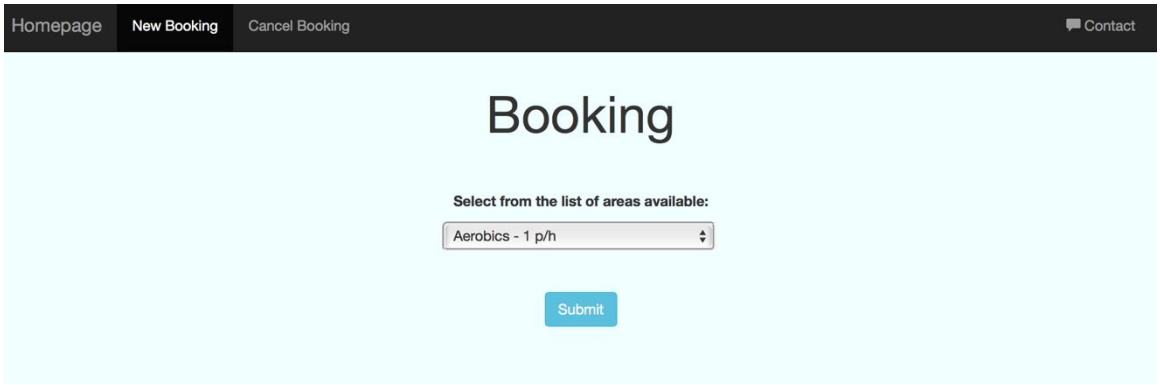
Result: The system shows the table for that week.



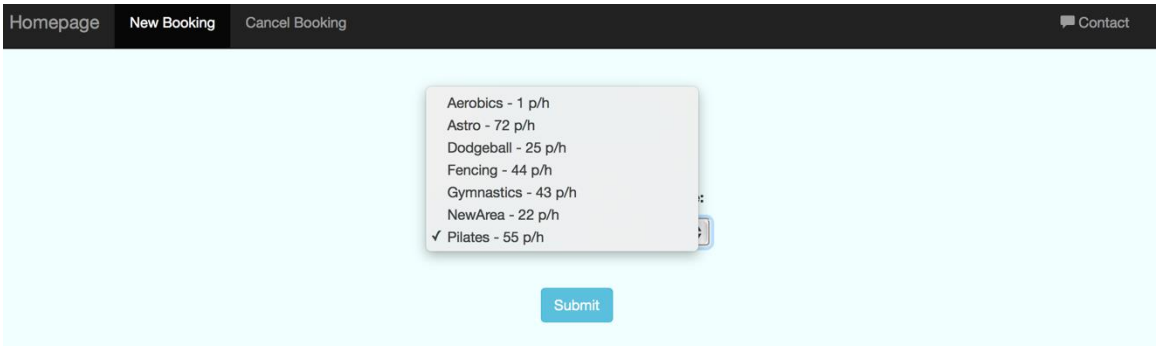
Test CNB4: Choose to select a different area.



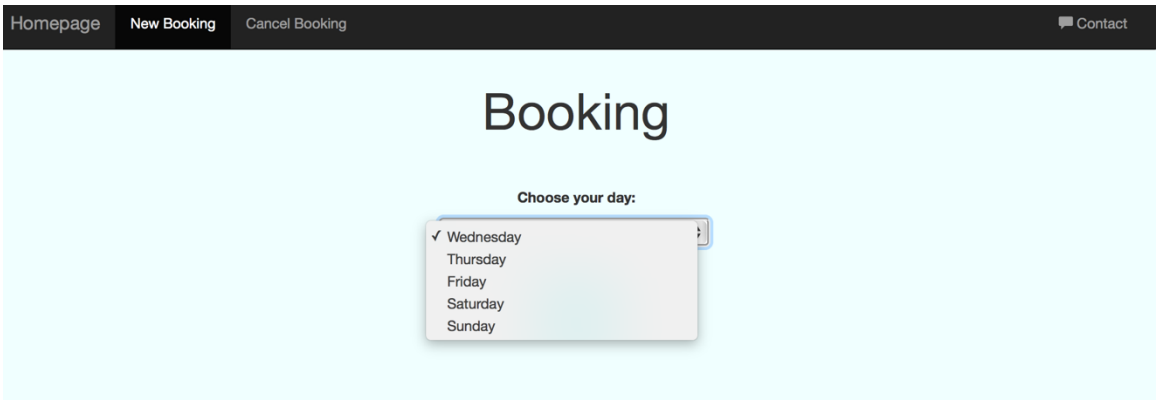
Result: The system re-presents the areas available.



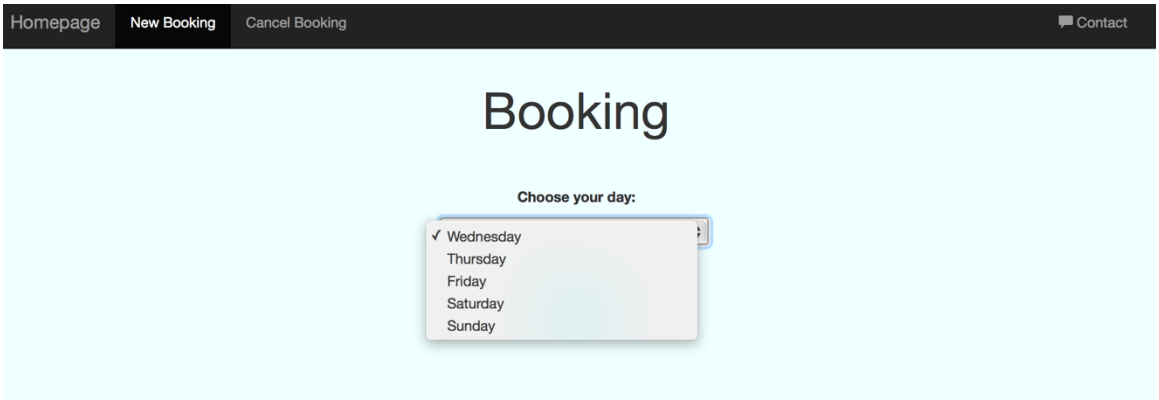
Test CNB5: Choose an area and click to make a booking.



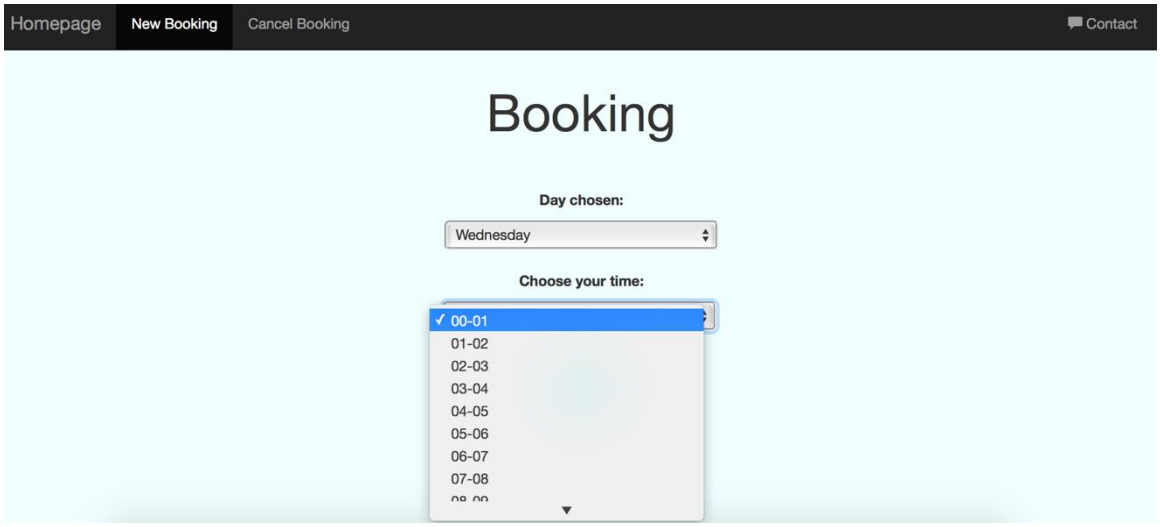
Result: The system presents the days available.



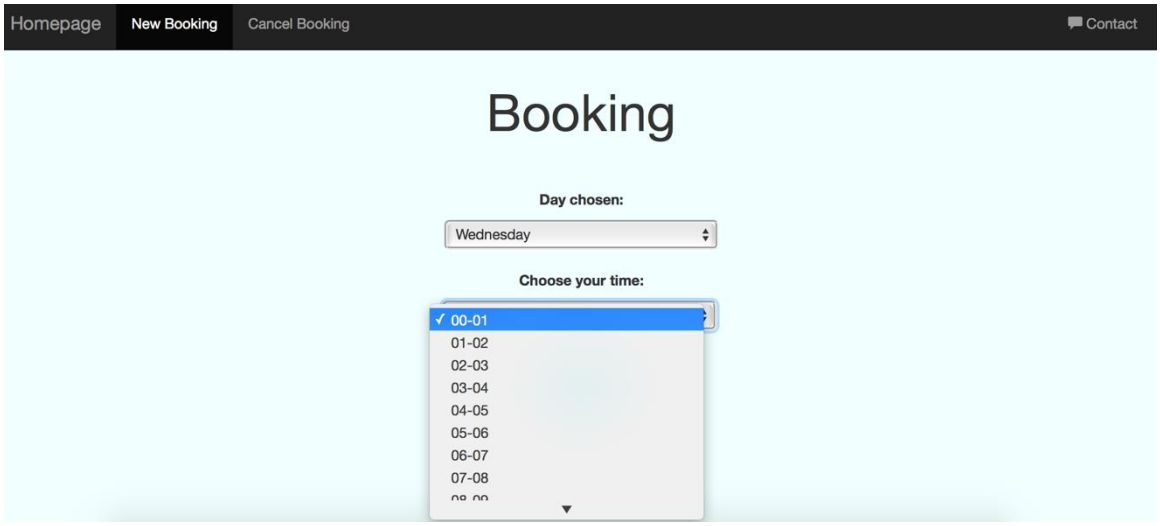
Test CNB6: Choose a day from the dropdown menu.



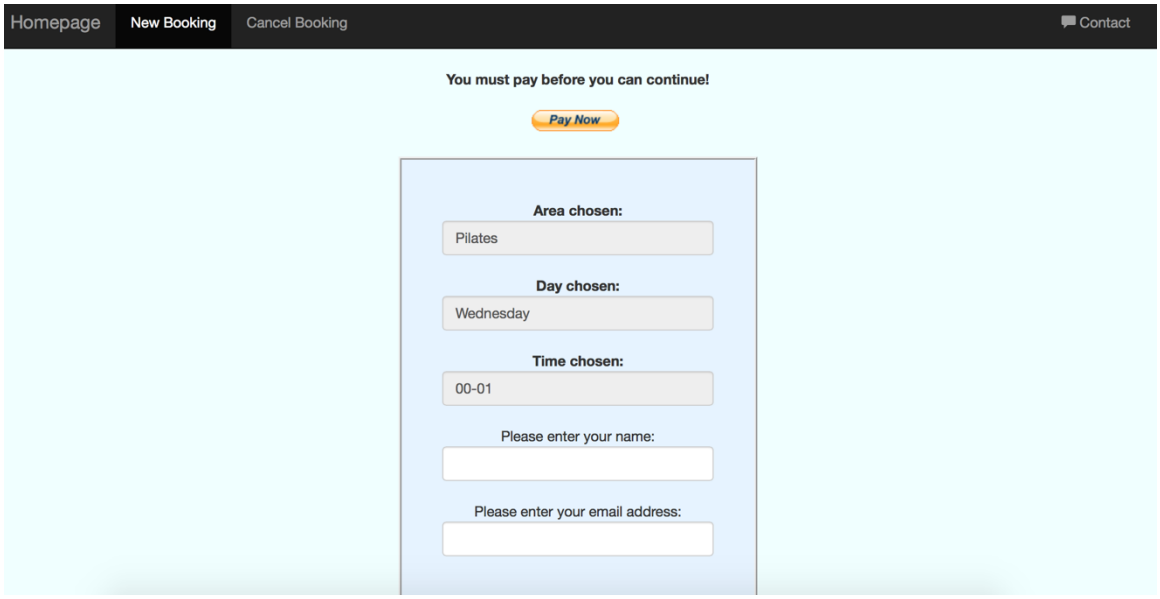
Result: The system displays the times available for that day.



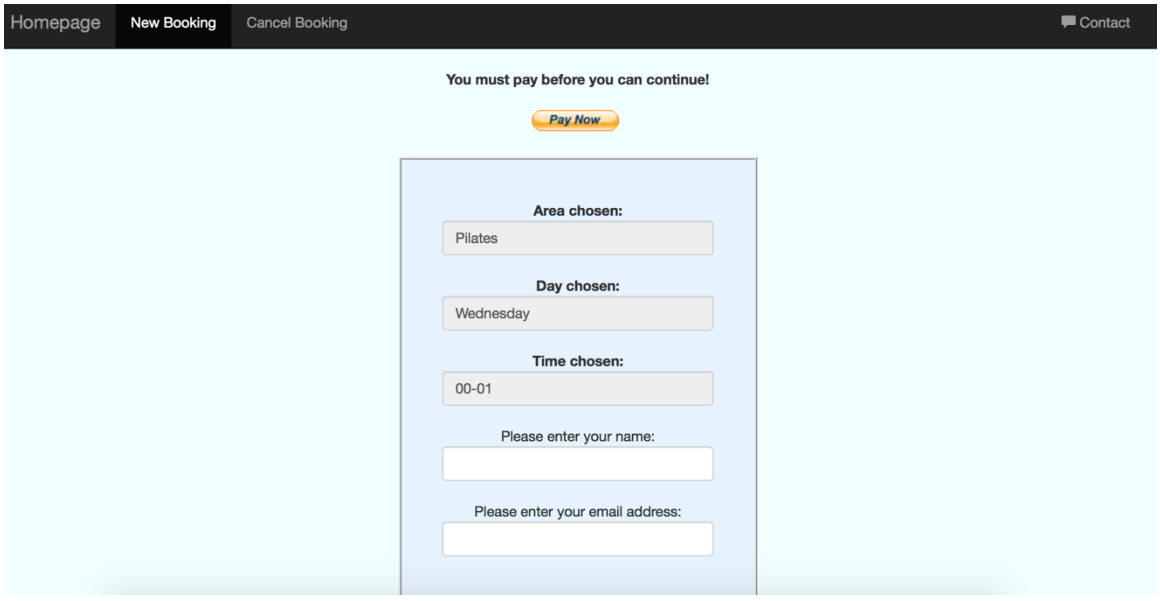
Test CNB7: Choose a time from the dropdown menu.



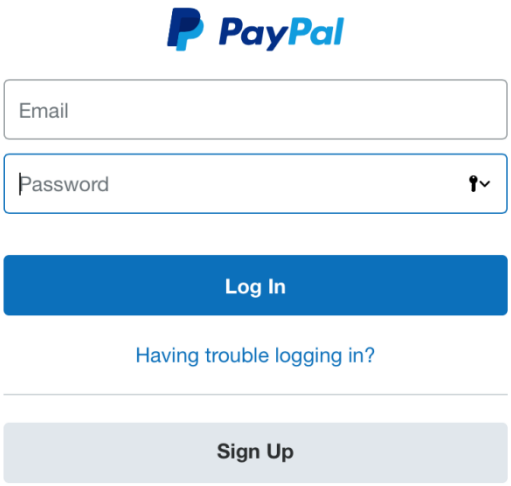
Result: The system presents a form with the area, day and time pre-populated.



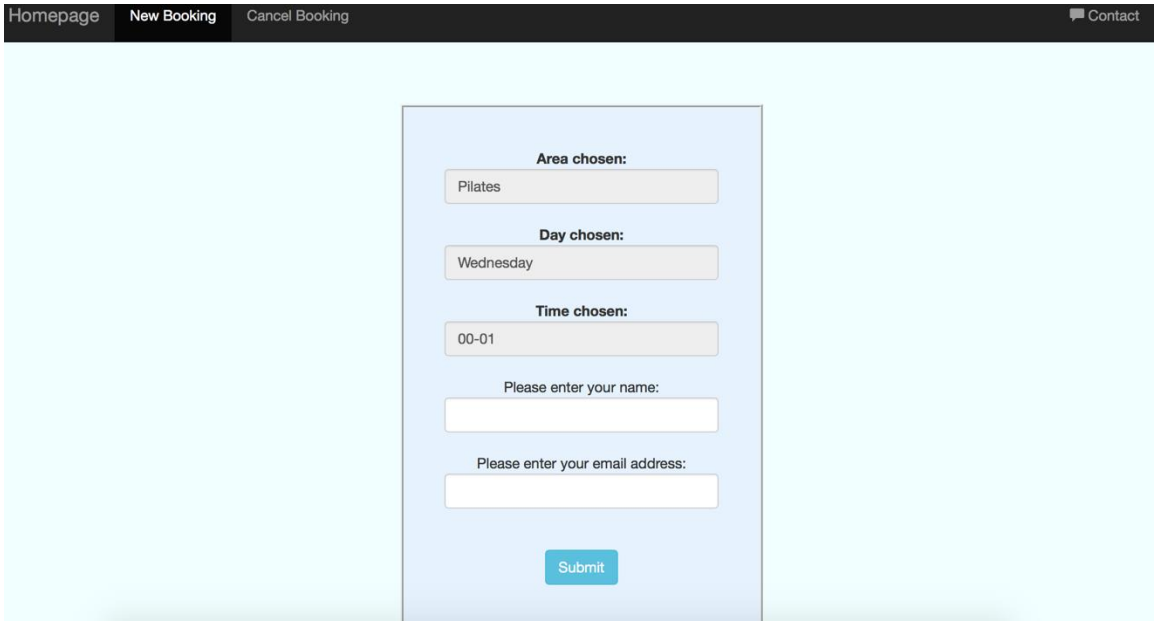
Test CNB8: Select the PayPal link to pay before completing booking.



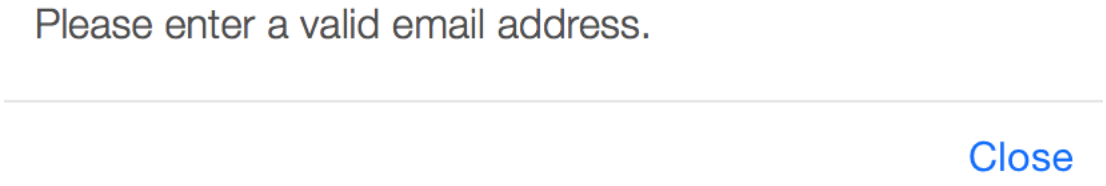
Result: The system brings you to PayPal externally.



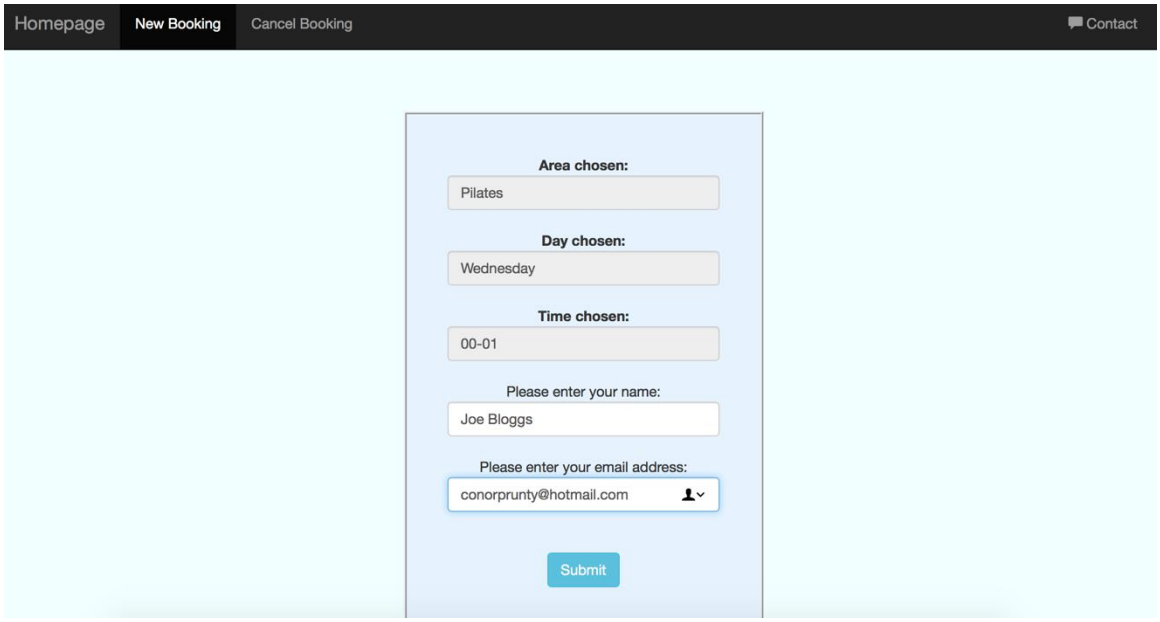
Test CNB9: Once PayPal has been selected, click 'submit' on the form.



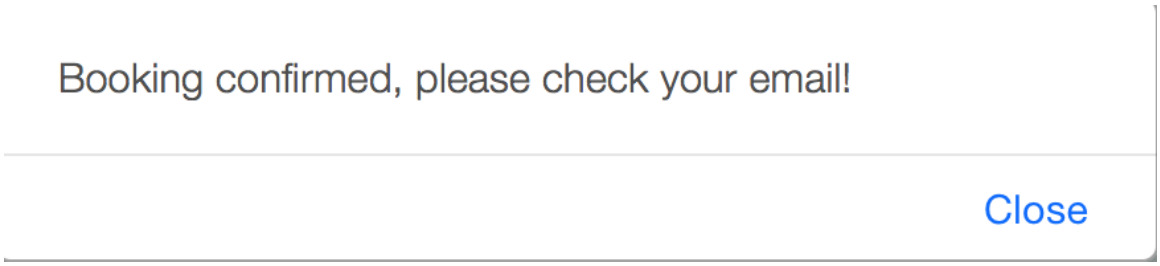
Result: The system asks you to complete the name section.



Test CNB10: Complete the name and email sections correctly and clicks submit.



Result: The system sends you an email confirming the booking and brings you to the booking complete page.



Hi Joe Bloggs,

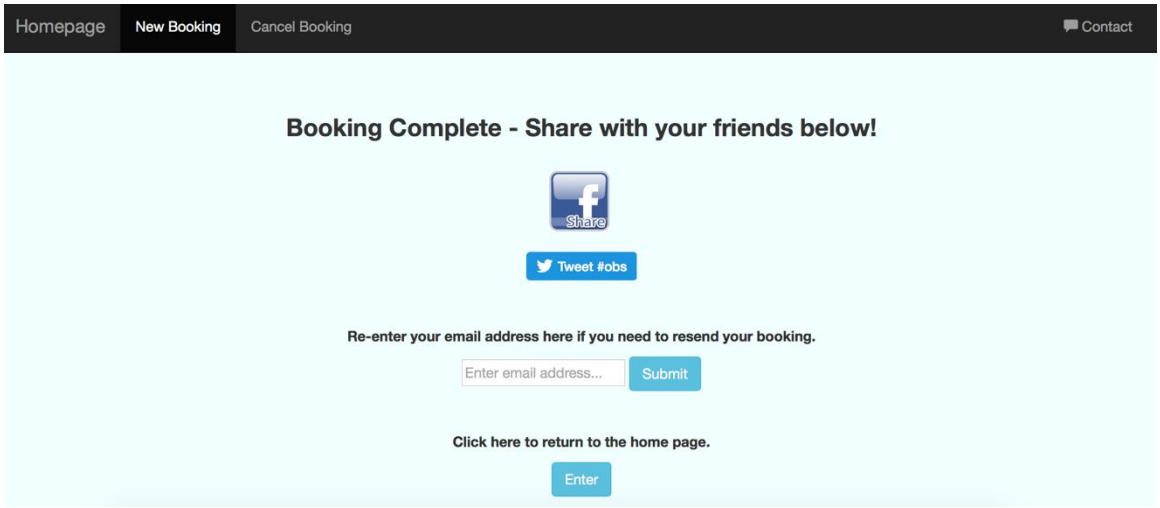
Your booking for Pilates on Wednesday at 00-01 has been confirmed for the week ending 2017-04-09.

Your unique booking reference is 291771.

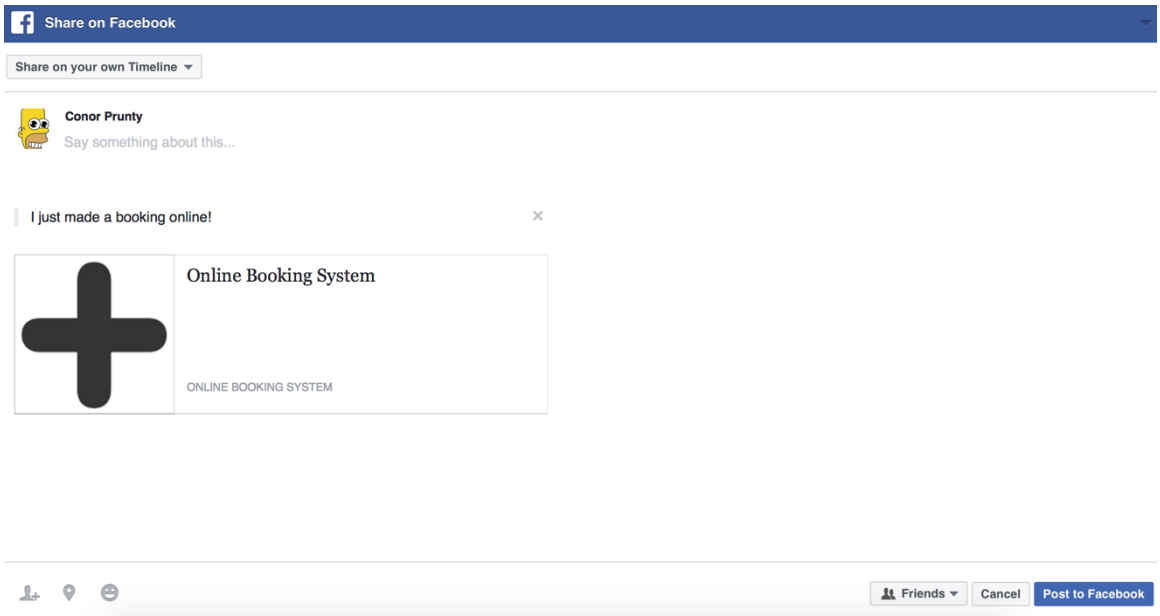
Please use this when arriving or in the event of a cancellation.

If there are any issues with this, or you did not request a booking, please contact conorprunty@hotmail.com

Thanks.

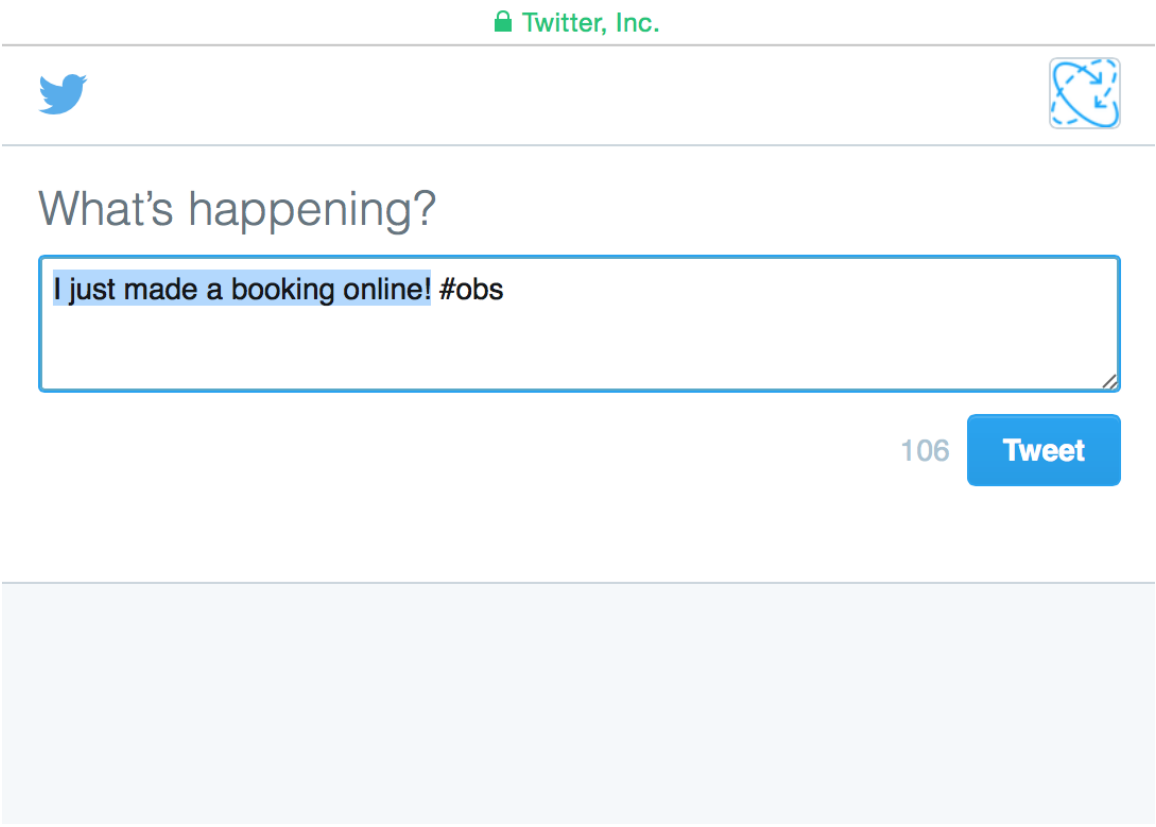


Test CNB11: Click the Facebook icon.



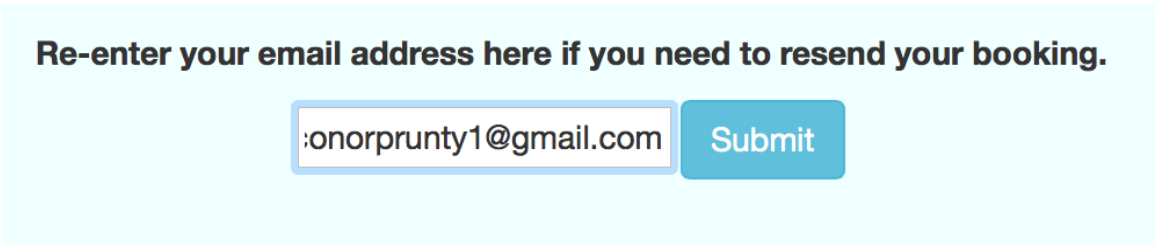
Result: The system opens Facebook with a default message to be shared.

Test CNB12: Click the Twitter icon.

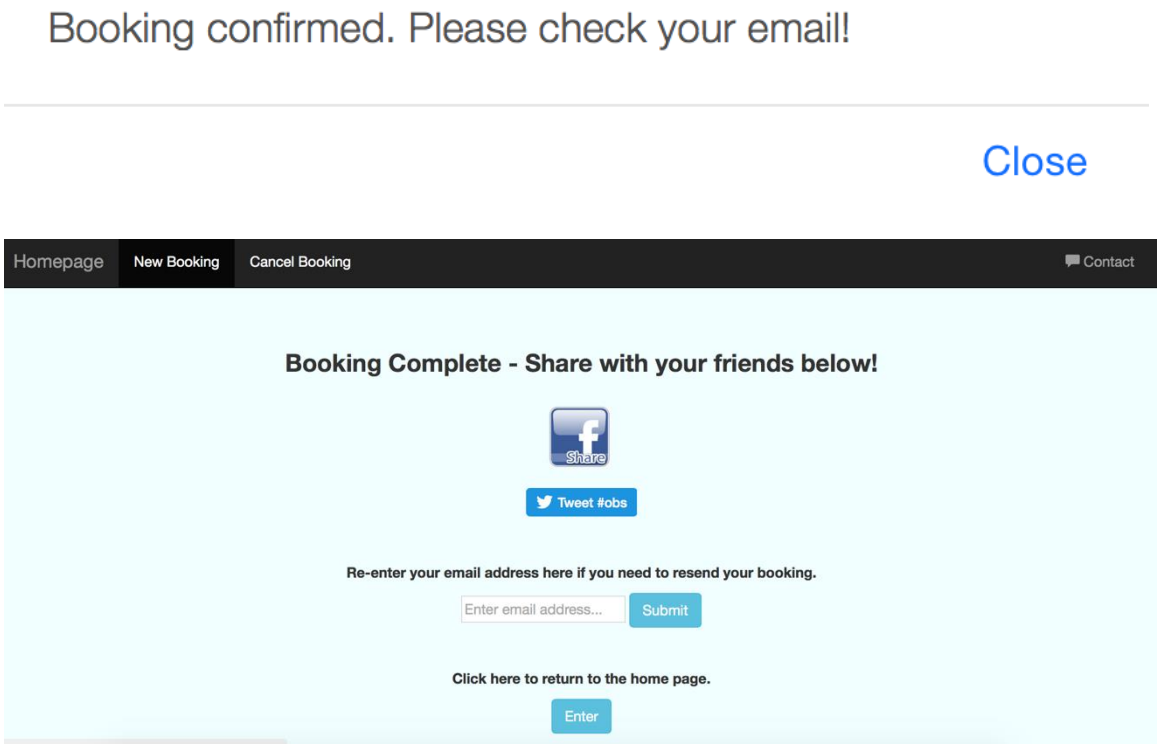


Result: The system opens Twitter with a default message to be shared.

Test CNB13: Enter an email address in the re-send section.



Result: The system re-sends the booking confirmation, and returns to the booking complete page.



Hi Joe Bloggs,

Your booking for Pilates on Wednesday at 00-01 has been confirmed.

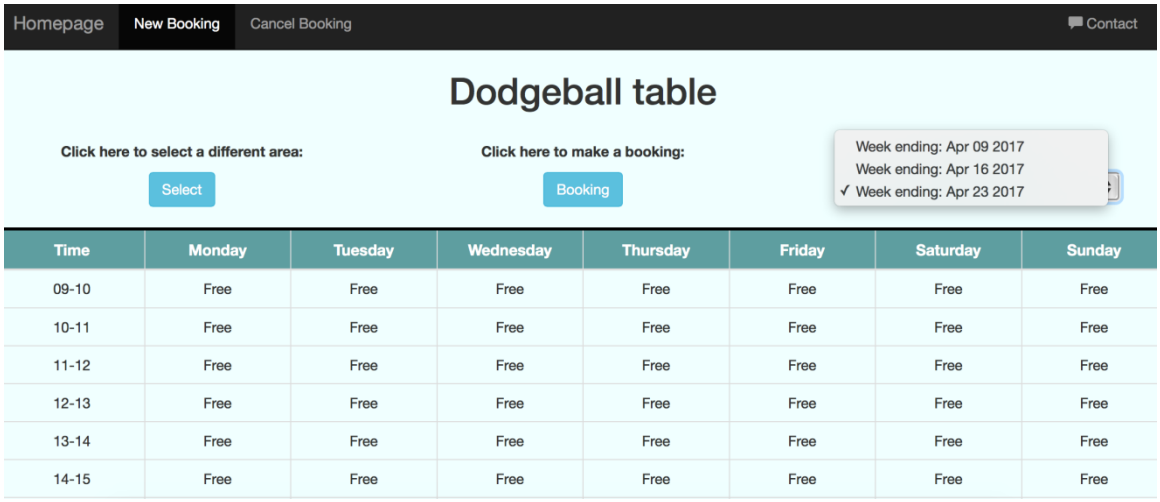
Your unique booking reference is 291771.

Please use this when arriving or in the event of a cancellation.

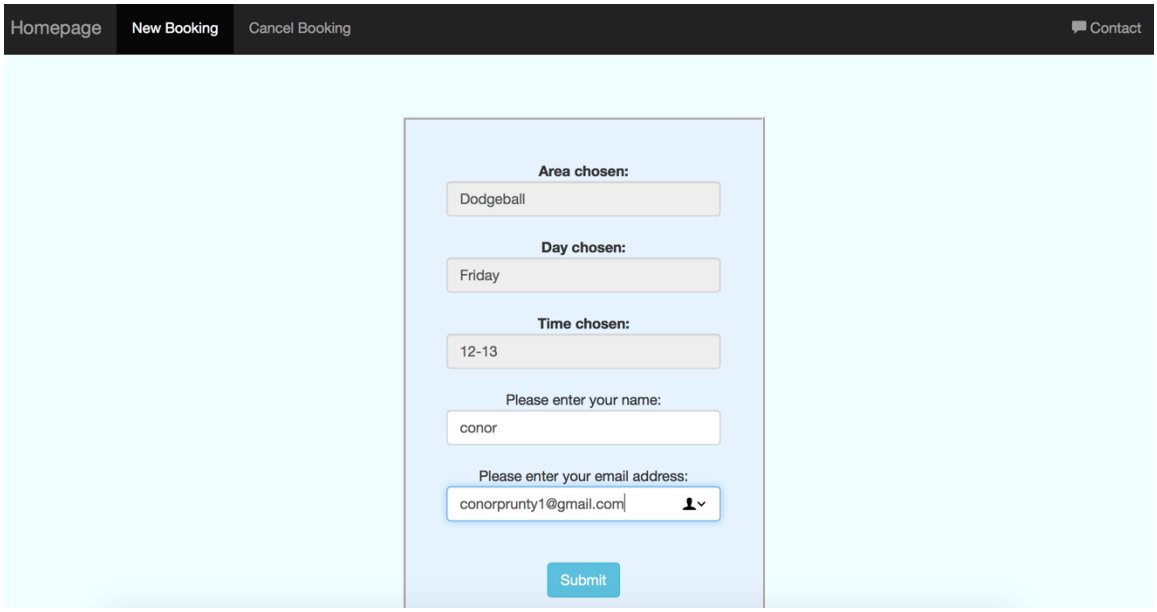
If there are any issues with this, or you did not request a booking, please contact conorprunty@hotmail.com

Thanks.

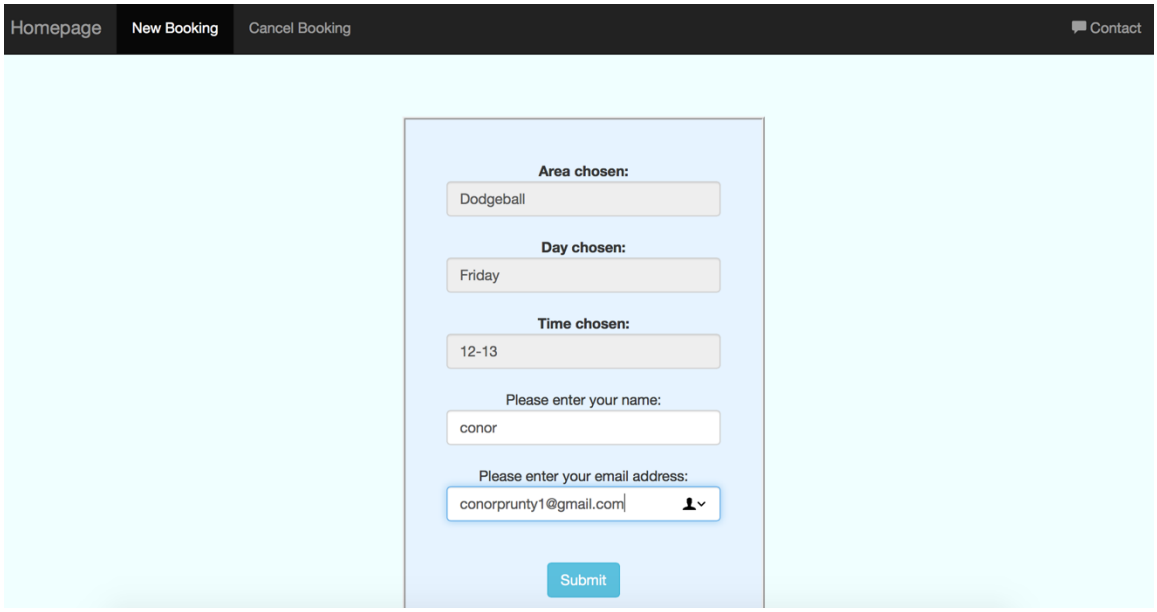
Test CNB14: Make a booking using one of the different week options.



Result: The system does as per previous booking and gets as far as presenting the options in the booking form.



Test CNB16: Enter details in the booking form.

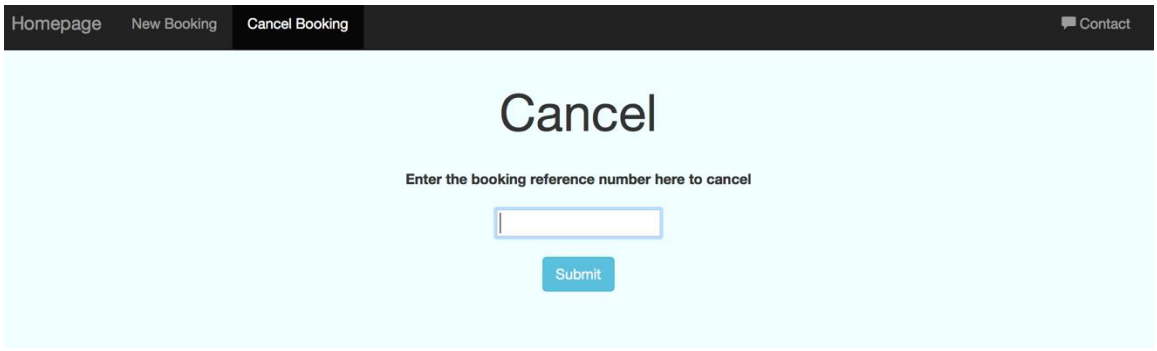


Result: The system sends an email confirming the booking, noting the correct week that was chosen.

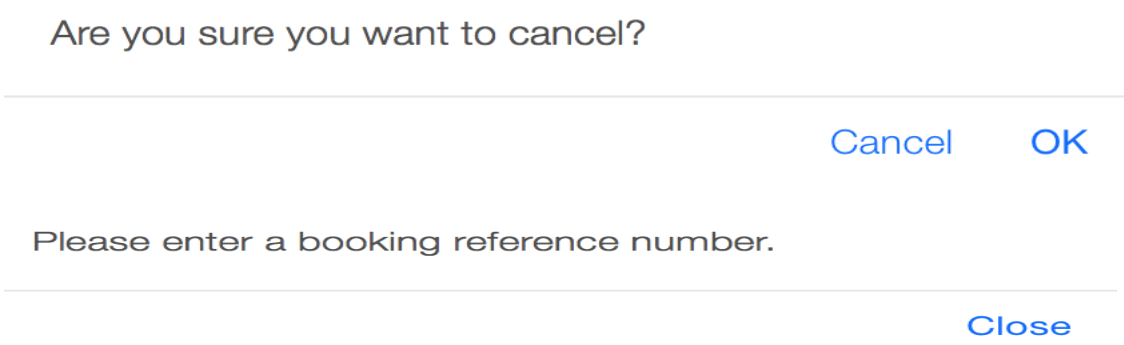
> Hi conor,
>
> Your booking for Dodgeball on Friday at 12-13 has been confirmed for the week ending 2017-04-23.
>
> Your unique booking reference is 871588.
>
> Please use this when arriving or in the event of a cancellation.
>
> If there are any issues with this, or you did not request a booking, please contact conorprunty@hotmail.com
>
> Thanks.

2.9.4 Cancel Booking

The customer cancel booking page.

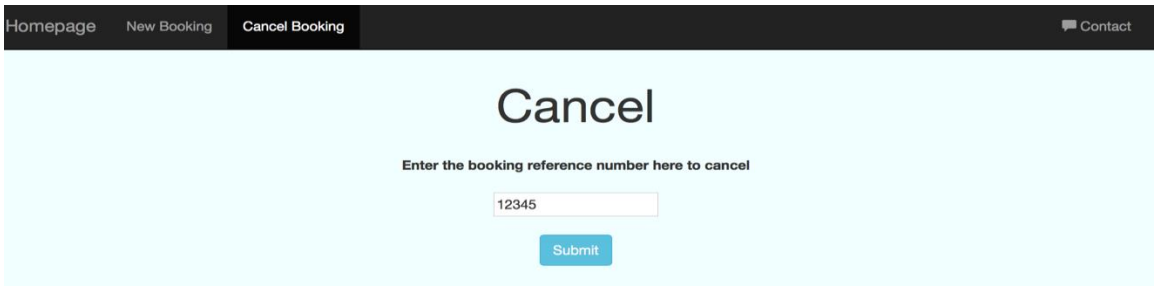


Test CCB1: Click 'submit' without entering a booking reference (Note: an 'are you sure' pop-up is always presented first).

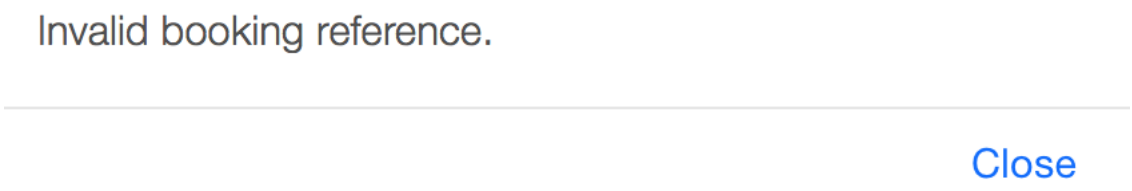


Result: The system asks you for a booking reference.

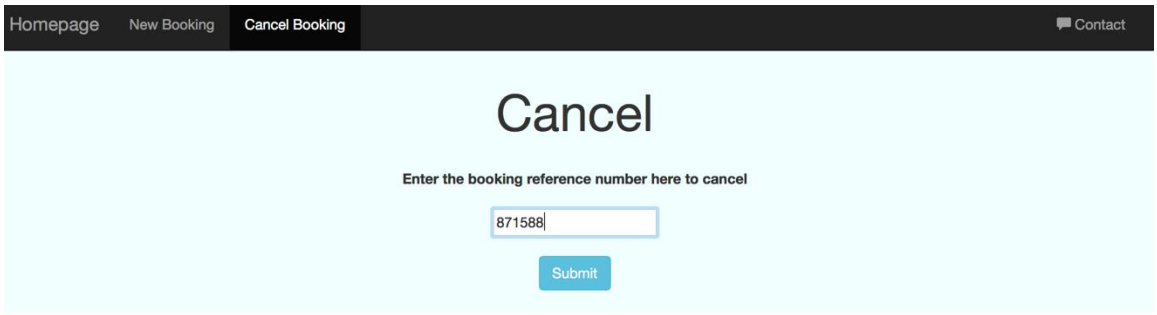
Test CCB2: Enter an invalid booking reference.



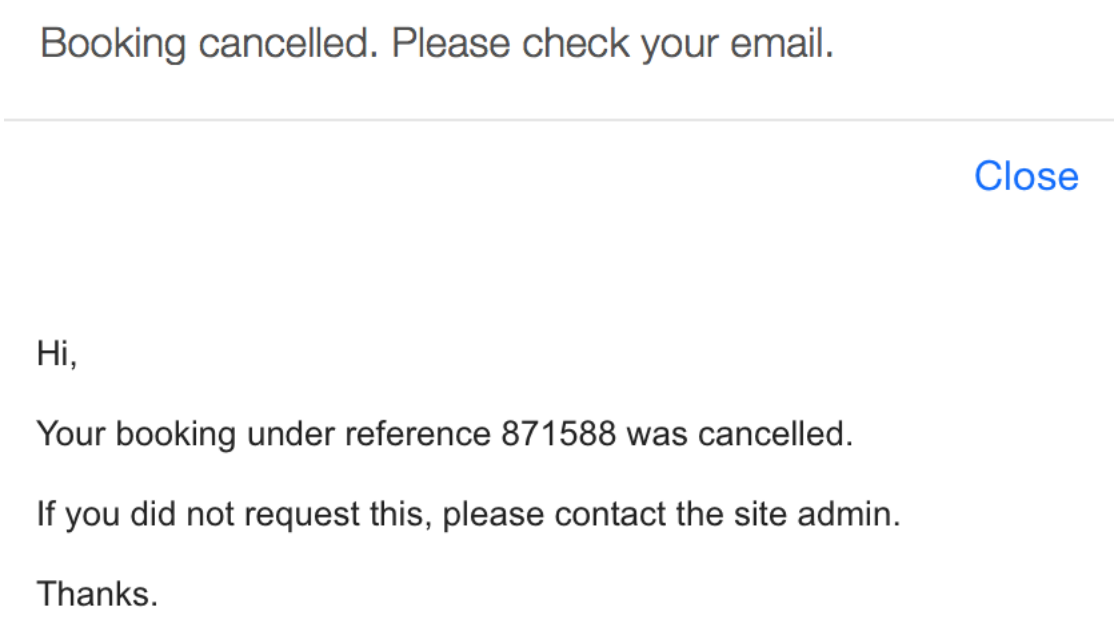
Result: The system displays ‘Invalid booking reference’.



Test CCB3: Enter a valid booking reference to be cancelled.

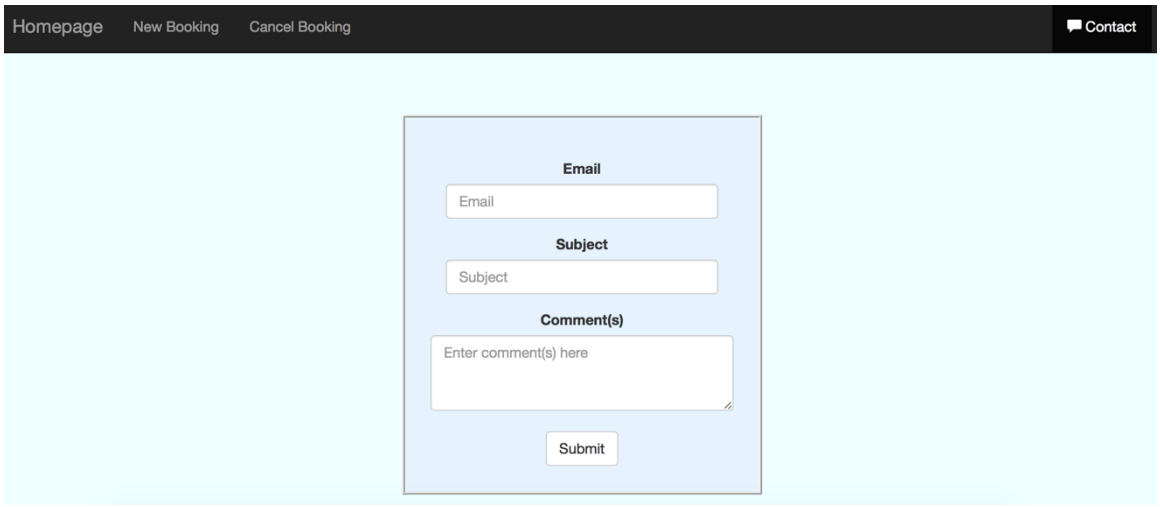


Result: The system deletes the booking reference and sends an email to the email address used upon booking to confirm it has been cancelled.

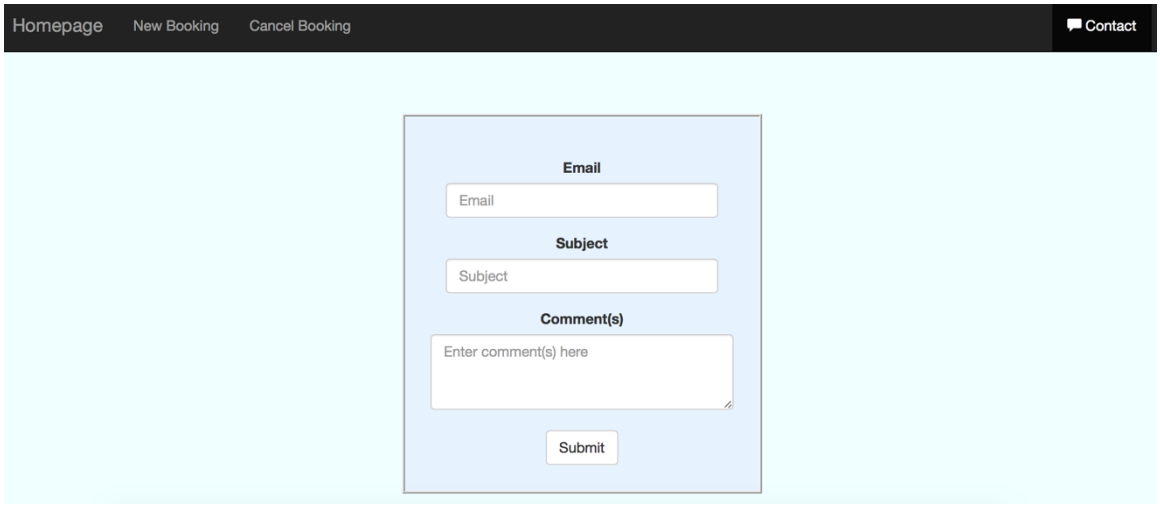


2.9.5 Contact Form

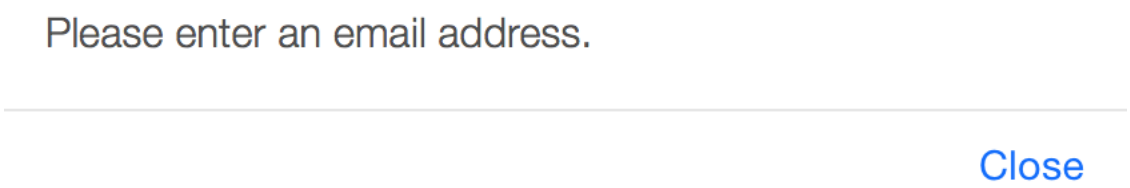
The customer contact form page.



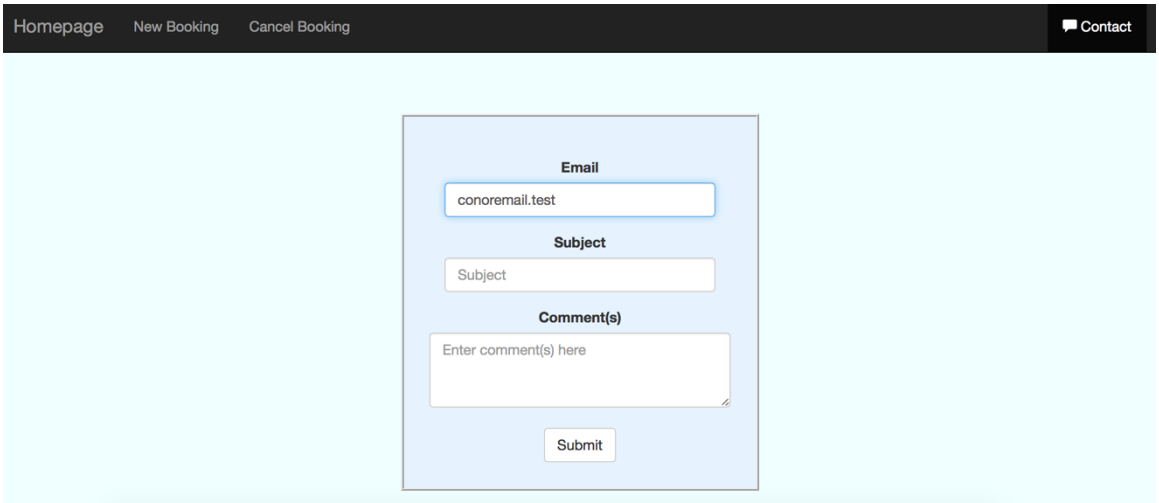
Test CCF1: Click submit without completing any details.



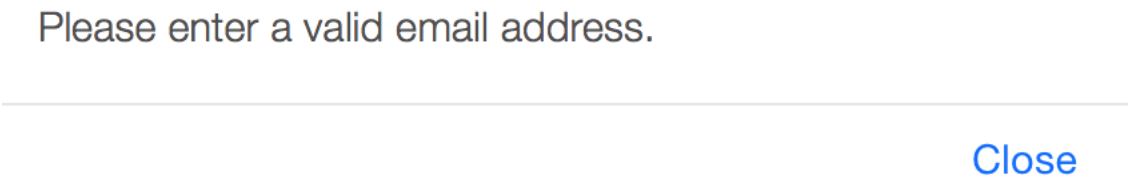
Result: The system asks for an email address.



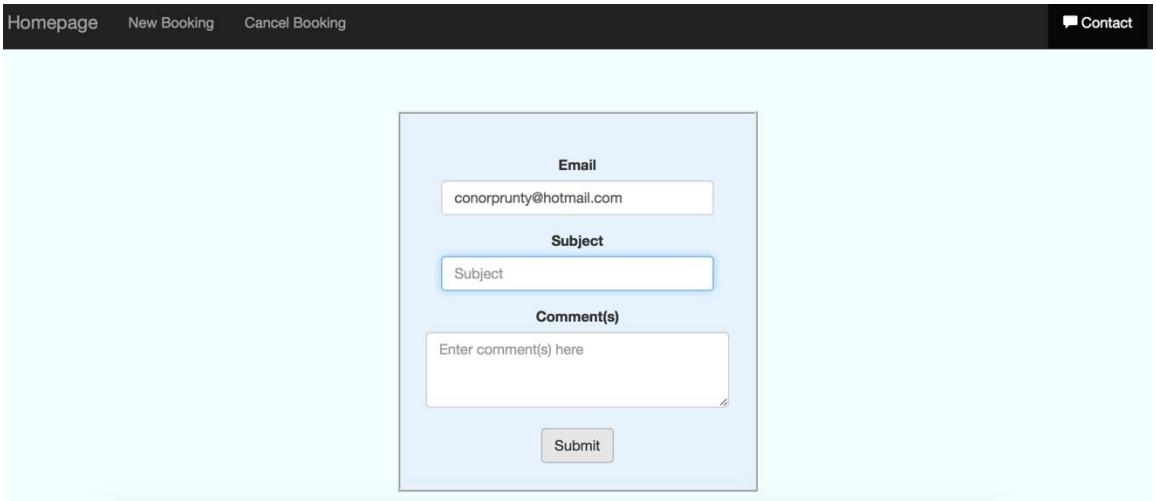
Test CCF2: Enter an invalid email address.



Result: The system asks for a valid email address.

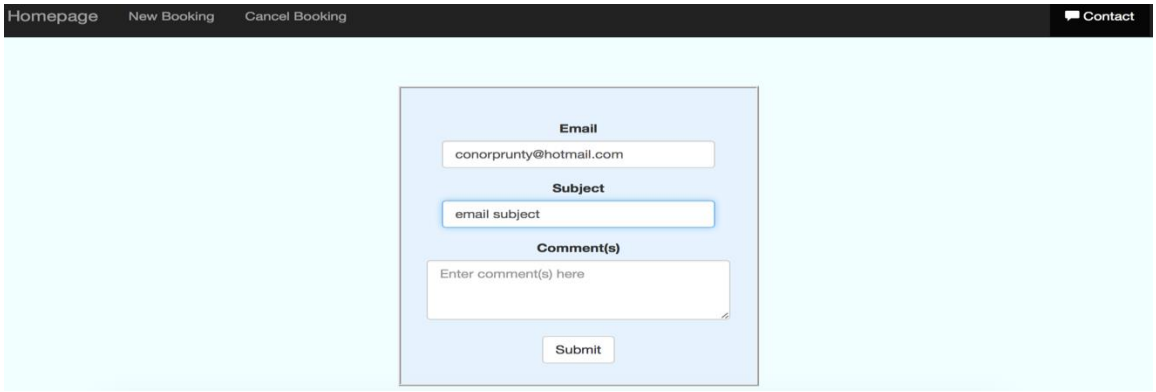


Test CCF3: Enter only a valid email address.

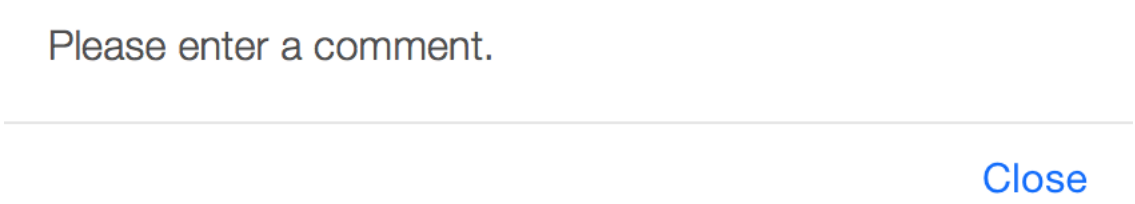


Result: The system asks for the subject to be completed.

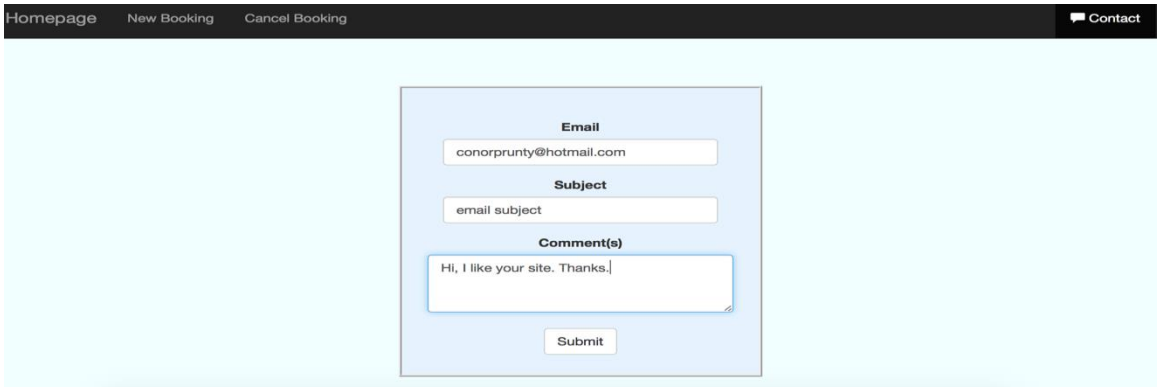
Test CCF4: Enter a subject and a valid email address.



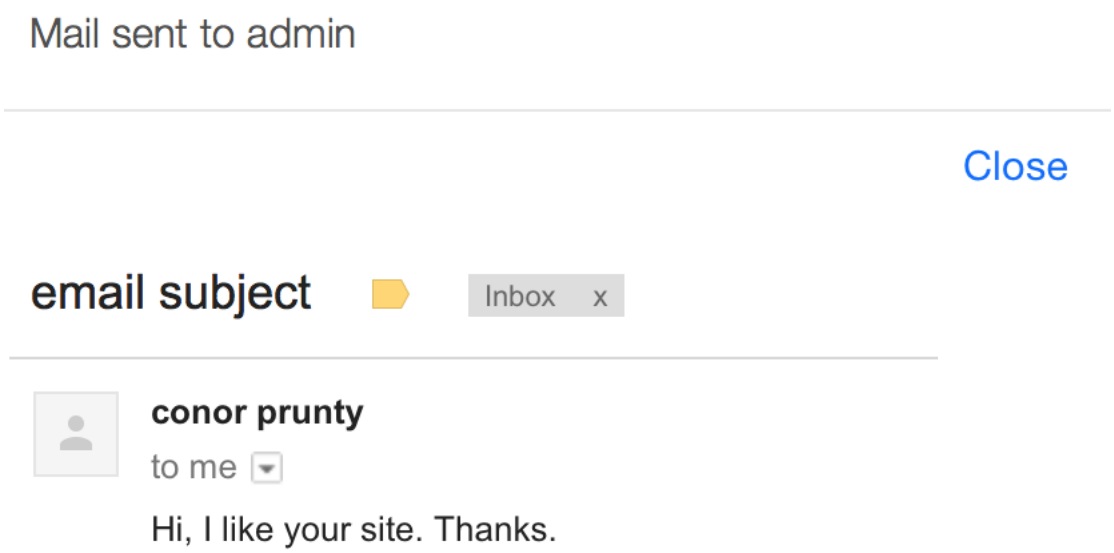
Result: The system asks for the comments to be completed.



Test CCF5: Complete the form with valid information.



Result: The system sends an email to the admin and a pop advises the user that an email has been sent.



2.10 Evaluation

To evaluate the project, I chose a well-known technique; heuristic evaluation. A heuristic evaluation is a usability inspection method for computer software that helps to identify usability problems in the user interface design (Uxness.in, 2017). I asked two colleagues to perform the analysis, with one being the admin user, and the other performing the role of the external customer. There are ten general principles to achieve these evaluations, which are explained below along with the results.

- ***Visibility of system status***

  Description: The site should show the user where they are, how they got there, and where they can go next.

  Admin: Easy to use and obvious where I am on the site. There is no 'how I got here' information, but it's not strictly necessary for this site I don't think.

  Customer: The navigation bar highlights what section I am on, so no issues here.

- ***Match between system and the real world***

  Description: The content of the text and phrases used should be what the user expects from a particular site, and not unrecognisable terminology or system-oriented terms.

  Admin: There isn't a great deal of text, which I like, and it makes it obvious what everything is meant to do.

  Customer: Nice simple text used throughout.

- ***User control and freedom***

  Description: Regularly a user can click on a link or page unintentionally, and the system should provide an easy-to-find and available undo button so the user doesn't need to navigate extensively to get back to where they wanted to be.

Admin: There are links to each section at the top of the page, and on some pages a second time so it's easy to find your way around if the wrong button is clicked.

Customer: There aren't many pages so it's not hard to find your way back. The navigation bar at the top also helps this.

- ***Consistency and standards***

  Description: The words and terminology used should not differ across the site, and the user should not have to try figure out if different actions do the same thing or not. A much higher success rate in usability terms is achieved if they navigation is based on what user's think, as opposed to what the company thinks.

  Admin: The setup feature has its own heading, but is also a main heading in the admin page. Not sure both are needed, but generally no issues here.

  Customer: I see consistency across all sections of the customer page.

- ***Error prevention***

  Description: Errors can occur. However, the software behind the site should cater for as many errors as possible and display a message to the user in plain terms.

  Admin: Simple messages were displayed when trying to do certain parts – such as registering with an invalid email address.

  Customer: I got an error when clicking the back button on the browser after I selected an incorrect value. It asked me to resubmit the form, which I clicked yes, and it seemed to continue working. I started the booking process again however, just to be sure.

- ***Recognition rather than recall***

  Description: The options and actions on the page should be visible, and try to reduce how much the user actually needs to memorise.

  Admin: The options are laid out nicely and visibility is not an issue.

  Customer: No issues with figuring out the actions on any of the pages.

- ***Flexibility and efficiency of use***

  Description: Accelerators may aid the user in displaying content on a certain section of the site that they have already requested. For example, if the user needs to register and login, the site should remember their interactions and display accordingly.

  Admin: I closed the browser and could return to the admin page without having to re-login.

  Customer: No login for me, so the system didn't need to remember any details.

- ***Aesthetic and minimalist design***

  Description: The design should be simple to use and easy on the eye. Irrelevant information should not be presented to the user. Non-important or rarely used information should be hidden from the user and only displayed when necessary.

  Admin: Very simple to use, big icons and generally quite pleasing on the eye.

  Customer: The large icons make it easy to use and I found no problems with the general design.

- ***Help users recognize, diagnose, and recover from errors***

  Description: The site should display user-friendly messages and not what the server or code may display in the background. The message should give the user a valid solution to recover from the error or retry what they were doing. For example, if the user enters a wrong password, the message should be "Password incorrect", and not "com.obs.securityException.getMethod.invoke".

  Admin: I didn't find any 'funny' exceptions when testing out any of the fields.

  Customer: I left some sections blank and the system returned user-friendly messages.

- ***Help and documentation***

  Description: Despite this not being required in an ideal world, the site should have a section to aid the user should they require it. It should be simple to use and provide a relevant answer to the user's query.

  Admin: There isn't a help section; the only help I could see was tooltips over the icons on the admin page.

  Customer: No help section visible, not fully necessary on this particular application, but would be nice nonetheless.

3 Conclusions

In conclusion, the project came together nicely in the end. All the functionality that was initially scoped, along with what feedback was provided at the half-way point, was implemented and works well with the application as a whole.

The main advantage is how quickly it is now for an admin user to set up the area(s) they have available, and then how quick a customer can book their chosen timeslot.

The application is somewhat mobile-friendly, to the extent that the page reacts well to different screen sizes, but it's not a mobile site, per se. In the future, I would like to create a similar app, specifically tailored to suit mobile devices. A lot of the code will be the same, but there would obviously be a few required changes to be made.

Currently, the only times available on the site are hourly slots. This was due to researching sports clubs websites, and noticing that this was what the majority had for what they were renting. In the future, I would like to add the option of half hour slots, or better yet, let the admin user define their own timeframes during the initial setup.

With more time and resources, this application could potentially go global and be used all across the world for sports clubs that take bookings. There could also be the option of non-sports clubs who take bookings making use of the site for whatever they may need it for.

Overall, I felt the project went well, and I am very happy with the end result. You can find the site online (see links below), but note that this is the layout for no specific club, whereas each club using the site would have their own unique URLs.

Admin: http://obs1.byethost14.com/index.php

Customer: http://obs1.byethost14.com/welcome.php

4 References

Designshack.net. (2017). *How & Why Prototypes Are Mandatory for Good Design | Design Shack*. [online] Available at: https://designshack.net/articles/graphics/how-why-prototypes-are-mandatory-for-good-design/ [Accessed 21 Apr. 2017].

Level, I., Tutorial, A., Dates, 2., Tests, I., Us, C., Policy, P., Use, T. and Us, A. (2017). *Why is software testing necessary?*. [online] Istqbexamcertification.com. Available at: http://istqbexamcertification.com/why-is-testing-necessary/ [Accessed 27 Apr. 2017].

Jetpeppers.com. (2017). *The importance of attractive website layout | Jet Peppers*. [online] Available at: http://jetpeppers.com/the-importance-of-attractive-website-layout/ [Accessed 14 Apr. 2017].

Perrigon, M. (2017). *Benefits of the requirements gathering process*. [online] Cyber Corp Software. Available at: https://cybercorpsoftware.com/blog/what-are-the-benefits-of-a-high-quality-requirements-gathering-process/ [Accessed 1 May 2017].

Tech-faq.com. (2017). *What Are the Benefits of PHP?*. [online] Available at: http://www.tech-faq.com/pros-of-php.html [Accessed 17 Apr. 2017].

Uxness.in. (2017). 10 Heuristic Principles – Jakob Nielsen's (Usability Heuristics). [online] Available at: http://www.uxness.in/2015/02/10-heuristic-principles-jakob-nielsens.html [Accessed 30 Apr. 2017].

5 Appendix

5.1 Declaration

SECTION 1

Name: Conor Prunty
Student ID: 13102311
Supervisor: Manuel Tova-Izquierdo

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism and that the work I have submitted for assessment is entirely my own work.

Signature:___*Conor Prunty*_____Date:___*02/05/2017*___

5.2 Project Proposal

This project will be to create an online booking system for use with sports clubs.

The system will be used primarily for booking any room, area, pitch or hall that a particular sports club has to hire. This can easily be expanded or reduced at the club's request to add or remove any booking functions needed.

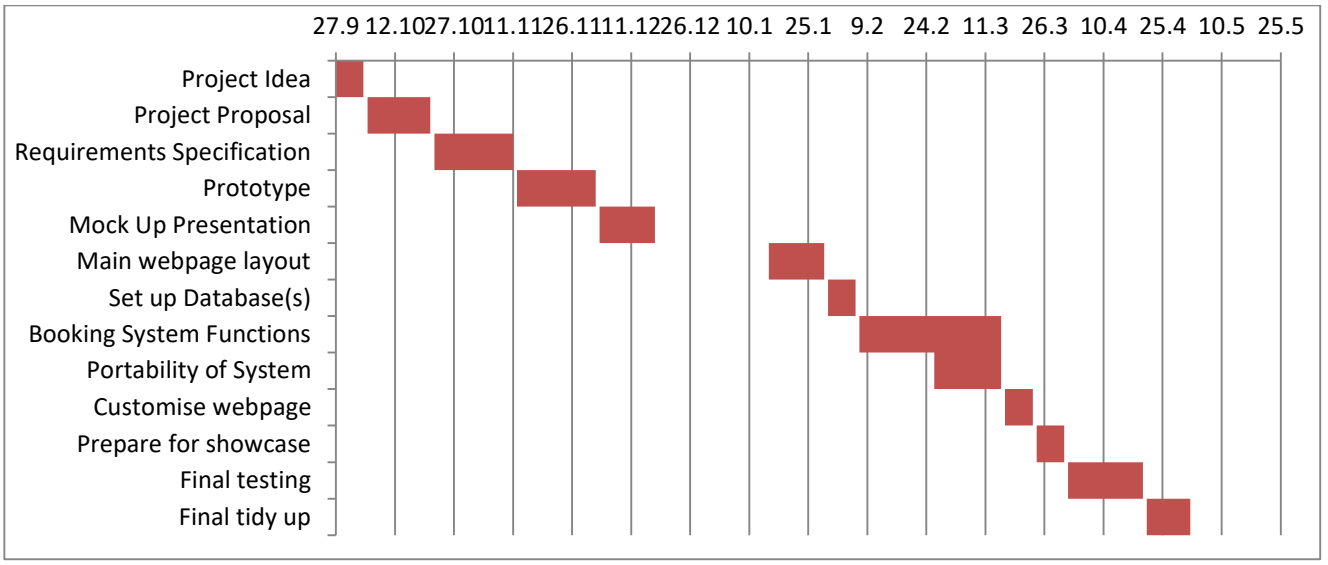
The system will be based around easy usability for the club member in charge. It will be kept in mind throughout that not everyone has technical capabilities.

The club has to register once, and the system will grant admin rights to that user for their particular club only. They will select from a list which sections their club has to offer, and also select the times and dates they are available for hire. They can view or cancel bookings as needed. The customer will be able to create bookings and cancel their own booking also. To ensure the customer can only cancel their own booking, a unique code will be generated and sent to the customer upon booking, along with their notification, and this will be needed to cancel.

The system will be portable so as it can be set up quite easily for any clubs looking for such functionality. The design will be neutral, but can be tailored to individual clubs' choice by adding their own banner, should they desire.

The customer will select their section of choice and a time, check availability, and then can book it. They will get a notification by email, sent once the booking is confirmed. As mentioned above, they will receive a unique code upon booking confirmation, which can also be used in the event of cancellation.

5.3 Project Plan



The project idea was formed, and the proposal was completed by the 21st of October.

The next deliverable was the Requirements Specification document. This gathered all the requirements and formed a plan of everything that is needed to be implemented.

This then ran into creating a prototype to show the audience what the project will look like upon completion. Although sometimes the design can change a little, this was a fairly accurate representation of the end goal. A presentation of this was done in mid-December, and showed the customer how the project is progressing to date.

There was a gap in the timeline around Christmas time, but this is due to studying and exams at the end of the semester.

From mid-January to early February, I have the webpage design combined with the required database(s) set up, and the link between the two for setting up users. Once completed, the system itself was created and implemented into the webpage. As this is the major part of the project, I gave it a 5 week timeframe for completion, but this is overlapping with setting the system up to be portable, as this is other major functionality required for the system to work as desired.

I had put in a week to go over the design of the webpage and system itself, in order to set it up for the showcase and final presentation. Although it looks like testing only occurs at the end, it was in fact throughout the whole project. I had put it in the chart initially from when coding began until the end, but I felt that this made the chart look ‘clunky’, so I only am displaying the final testing phase.

There was also a final tidy up at the end. This incorporated any minor bug fixes that were left, edits to the website layout and ensured everything was ready for the final presentation.

5.4 Monthly Journals

5.4.1 September

20/09

So, back in college. It's been a long time since my January exams but here we go for fourth year. First class was a surprisingly large class; to the extent that one latecomer didn't even have a chair or desk.

Eamon is the lecturer for our main project. We got a general update on the project – what it is, timeframes, marking schemes etc. Time to start mulling over ideas, despite having had a great plan all summer to come up with one…

I like the idea Eamon put forward of meeting with supervisors; this kind of thing is always good to get opinions and feedback from people who have done it before.

The class ends at half 7 so I run down for a cup of tea and back up for the next class. No sign of the lecturer yet… 10 to 8, still no sign… 10 past 8, still no sign…the lads are slowly starting to leave. At half 8, I've had enough, as did the majority, and it is time to go home. I had to collect the wife after I got home, so had a quick chat in her friend's house with 3 others about my project. Just for some general ideas, and more just to get a feel for how other people think in this kind of situation.

21/09

Had a quick chat before work began with one of my colleagues; fishing for project ideas – as many different opinions the better. I didn't get much this morning, I asked my brothers also and one came up with unmanageable ideas, the other with unrealistic stuff. I will think over it for the next few days and ask the lads in college on Thursday or Saturday how they're getting on with their ideas.

22/09

Back in class again today for 2 subjects – Strategic Management and Web Services / API Dev. Had come up with an idea to create a website for my local GAA club; it would have all the usual results, fixtures, history etc., but also members can log in and pay, book the function room for private functions, book the astro pitch, check availability, etc. I had a quick pitch of this idea to the lads but

didn’t really get much more than a grunt, so not sure if its good, bad, or that they just haven’t come up with anything yet so they don’t care about mine…

Ron’s class went off grand, much less numbers than day 1, and from experience will dwindle slowly into the bare bones. Ron knows his stuff but theory based business subjects aren’t for me so I’ll have to dig deep to do well. 2nd class was with a new lecturer, Manuel. He went through the history of computers and I kind of zoned out and started to think of other ideas for my project or whether my existing idea will be complex enough. I’ve decided I’ll give it another 3 or 4 days before I start putting more effort into this specific idea, if I’ve come up with nothing much else.

24/09

Spent last night chatting with the in-laws as they were home from their 3-week holiday so didn’t do any college work. Up early this morning for more new classes, and they’ve changed our timetable which now gives us a 3-hour break. I’m unsure whether this is good or bad – I’d like to get out earlier on a Saturday but the break may give me time for projects and other course work. We started with Data App Dev – I was thinking how I could get incorporate data analytics somehow into my main project, but haven’t given it enough thought yet. Big break now before the afternoon class, so I went to Lagoona to watch the football before the next class – Multimedia and Mobile App Dev.

Another new lecturer, Swapnil, for this subject. We just did some intro and a ‘hello world’ program on Android Studio to get us used to it. Fairly low attendance with several leaving during the class. Lecturer didn’t seem to care. He also said that this is an intro class but could cover more advanced topics if necessary – personally as we’ve many other projects, I’d be happy with just an intro if it’s going to give good marks overall. Looks like we only have 3 Xmas exams and not 4 as the timetable changed, which is great.

Home time, and I'm wrecked, time to put the feet up and not think about the upcoming semester for a night.

27/09

Joined the gym today so all I did was a try out a few machines, and a short run on the treadmill. I'll try building it up, but the hardest part is getting into a routine of going.

Anyway, I started researching more into my project. Had a look at some GAA clubs, and some sports complexes and not one has an online booking system (well ALSAA do but it's 'temporarily' out of order), so you either ring or email them. From working in a bar in a GAA club, I know how infrequent they answer the phone as first of all, it's closed during the day usually, and second, if it's any way busy in the bar, they won't have time to answer. With email, you've no idea how long they may take to respond, so what I propose is to create an online booking system for a sports club, with an immediate response. I will research more into ones and see if any exist and if they're any good first.

29/09

So it's getting closer to having to put our ideas to a panel, so I've decided I will go ahead with my idea. I'll create a website for a GAA/sports club (not fully decided which), which will have an online booking system for their facilities e.g. astro pitch, function room etc. It will also be a full website where members can log in and pay for their upcoming membership, have general results/fixtures etc. (for a GAA club), and I'll try incorporate some of our other modules this year, such as calling APIs and possibly data analytics. I think this whole idea could work, and would be complex enough to make, especially if I'm writing my own API (if that's feasible in the timeframe). I have done more research online and still cannot find a place where you can book pitches online, bar the ALSAA which I mentioned previously

does not work. Over the weekend I will practice a quick spiel and be ready to present to whoever is on the panel.

5.4.2 October

01/10

Saturday morning – and for some reason I'm in at 8:15am for a 9 o clock start. We've a rescheduled class instead of our regular class. A few of the lads starting filtering in and they seem to have come up with a few ideas for themselves, but I'm glad I've my own idea and not feeling left behind. I've no idea how the panel will react on Tuesday when I pitch the idea. One of the lads hasn't really come up with anything but Eamon told us the other day that they have about 40 ideas if you have nothing so he doesn't seem too bothered.

We've a three-hour gap between classes every Saturday. We've requested to change the time but the college isn't budging so I will use that time for project work, or anything else I need to catch up on.

I am happy enough still with my idea so on Monday night I will go through what I will say during the pitch, as tomorrow I've other work to contend with and I've a feeling soon I'll wish there was more hours in the day.

04/10

Pitched the idea to Eamon, Anu and Dominic. The idea got the green light, thankfully, although with a minor change that I probably should have thought myself. The online booking system should be created with the idea that it could be easily implemented for any potential users to set up, especially those with no technical knowledge. They also suggested I should get an actual club to set it up with initially instead of my idea of a fictitious one.

I have in mind two potential clubs to go speak to, and will do when I get the chance. I'm hoping that this is all manageable but I guess the majority of this semester is

the documentation before the coding begins after my January exams. This is probably good as we have 2 less subjects next semester, and it ends in April. But, we have a few weeks to finish coding this main project after those exams are complete.

06/10

Got stuck in work as a big release happening this weekend so didn't get in to college until late and missed the first class. The second class I went to but was fairly exhausted so didn't take in as much as I would have liked. I'm away now with the family-in-law for the weekend so won't be in Saturday but I have the lads informed so they can update me with any important issues.

11/10

Back in college after a nice weekend down in Waterford. Eamon has said we won't be getting our supervisors until after the project proposal has been submitted. Yesterday, I reviewed his template and set it out how I would like it to work – although didn't really put in much content. I won't get a chance tomorrow or Thursday to look at it, so Friday after work seems like when I'll put the majority of the work into it.

13/10

I started on my project proposal today. For now, using the proposal template on Moodle, I just wanted more to scribble down a few points in each section first before I write anything properly. More like a brainstorming session with myself. I did this for the first few sections and I think I've enough material (for those parts anyway) to start my write up properly. I also need to look into creating a Gantt chart too, as I'm not terribly familiar with them.

15/10

I put together a rough draft of each section, but knowing that I have to tidy them up and provide more information. I like to break problems down into smaller, manageable parts so that you don't feel like there's far too much work to do in one go. At this stage, I reckon I've probably about 50% of the proposal done, and will

continue again next week. I haven't looked into the Gantt chart as of yet, but it's on my mind to do next.

18/10

Finished with my proposal, both over the weekend and before college today. Submitted to Moodle. I know it's early but, along with other subjects, feels good to start getting things submitted (and on time). I'm doing my best to keep on top of everything and so far, so good. Eamon couldn't make class tonight so that hour helped to tidy up the last few bits. Eagerly awaiting my supervisor now.

21/10

Found out my supervisor – Manuel. I had him for class yesterday but didn't know until today. He seems like a nice man so I will mail him when I get a chance, but right now I've two exams tomorrow in college so need to put all my effort into that.

25/10

First exam was an essay style question to be answered, and I gave it my best shot. No idea how it's going to be graded. Second exam was to make a mobile app with specific requirements; didn't seem too bad and got about 90% of it working. I do get impatient though and would like immediate results! Haven't got results for anything yet this year, so hopefully the lecturers don't all wait until Christmas to give them to us. Emailed Manuel this morning also, so I'm hoping for a quick response, but I have him in class on Thursday so I can re-ask him then if necessary.

Eamon went through the requirements documentation that's required. It seems like it's going to be a heavy enough document, so will break it down into sections and not rush into it. It's reading week coming up so after Thursday's lectures, we're off college until Saturday week. Also, I'm off work Monday as it's a bank holiday, so I've 3 full days to start the document, along with the 75 other things due over the next two weeks.

27/10

Hadn't received a response from my supervisor so I sent another mail, both of them through Moodle as I didn't have the email address. Got an instant response this time, so maybe the first didn't send for some reason. My first meeting is the Thursday after the reading week, so hopefully it's not too late for the first meeting, but I'm sure it's grand. My 2nd lecturer today, who so happens to be my supervisor, can't attend tonight so I get to go home early, which is nice. Also means a slightly longer reading week!

30/10

Sunday, and I had great plans to begin my requirements document, but of course it was great timing to get sick! I have a pretty bad cold (manflu), but I'm finding it hard to concentrate on anything so I'll just have to rest until it goes away. At least I have Monday off work for the Halloween bank holiday. I'll see how I feel tomorrow but not getting my hopes up to do any work yet.

5.4.3 November

01/11

Back in work today after the long weekend, but still a little 'flu-ey'. I can't leave it much longer though to start the requirements document, so once I get home tonight, I'll put as much effort as I can in to get a start on it. If I was feeling a little better, I'd probably go to the library (assuming the college is open during reading week), and I wouldn't have any of the distractions that I would at home (such as Champions League!), but for now, I'll do it at home and perhaps go in to the library on Thursday after work. Would really like this to be done as soon as possible, as I've more CAs coming up and don't want it hanging over me.

04/11

I've been off work since yesterday with a chest infection – think it's possibly the first one I've ever had. So I used the time to get a good start at this document. I

got as far as the first two sections, the use case diagram and the functional requirements over the last 2 or 3 days. Silver lining and all that.

Saturday tomorrow but I'm on antibiotics and won't be going into college so I'll try continue and get this document done.

05/11

I got some of the later sections done today, like the system architecture. I used a mix of Creately and Gliffy for the diagrams; those two free (trial at least) online tools are very handy. I hadn't logged in to either of them in quite a while, but still had some diagrams from this time last year on them so they were a good base to start with. I put together a rough draft of the non-functional requirements today too, but I've other college work to do, including an exam next Thursday so have to prepare for that too.

06/11

It's Sunday and I usually like to take some days away from college work but it seems that there is an absolute ton to be done over the next 6 weeks including several CAs, 3 projects, and more work on this project. Oh, and a handful of reports also. Fun times. I think there are not enough hours in the day.

Having said that, I just completed the non-functional requirements today and did a little more preparation for the CA on Thursday, and that's enough for today.

8/11

Took yesterday off completely. Was back in work and still not fully over the chest infection so just put the feet up after I got home. I did a run through of the interface requirements, which is the last section I have left. I'll complete it tomorrow.

09/11

I proof-read and went over all the sections I've already done. This document is taking way longer than I'd hoped. I tidied up and spelling errors etc. and sent it to my classmate Kevin to get a second pair of eyes. He replied by the end of the day,

giving a few tips and pointers, and some other potential requirements I may attempt to incorporate. I'll sleep on them and have one final look tomorrow.

10/11

Made a few small changes following the peer review and am pretty happy with the overall document. It's like 20+ pages and over 3500 words, but I guess that reflects the work put in. I know I said it's the last time I'd look at it, but I will have one final look tomorrow before I submit.

11/11

Met my supervisor last night, and he suggested to make some changes to the document. Nothing major, but try sell it more on the proposal and scope in the first section as this is where the reader would read first. There has been an extension on the deadline for this, so I do have until next weekend, but will try getting it done as soon as possible.

14/11

Made some changes to the document – tried to sell it more in the proposal and scope sections. I will leave it alone for a few days and have a last and final look on Wednesday or Thursday before submitting. I also rechecked the submission date and it's not until the 18th of the month, so will submit once I do have my final read (or edit).

15/11

I submitted my requirements document tonight; I had enough of looking at it for now and just made sure it was submitted before the deadline. I know it can be changed if necessary, but for now I'm happy with it. My next meeting with Manuel is Thursday week and he's to come back to me with what exactly is needed for the mid-point presentation. I've two exams this week and another next week so the main project is going to take the back foot for the next few days.

23/11

Haven't looked at anything to do with the main project since the requirements doc was submitted. Eamon said last night we've to submit a technical report which includes the project proposal and the requirements specification that we've already done, plus any changes that have been made. I haven't changed anything yet, but am meeting my supervisor again tomorrow night so I will hold off until after that, just in case. Trying to juggle the other 5 subjects along with this project is difficult so sometimes certain ones have to take a back seat. I'll find out tomorrow how much coding is needed for the presentation (mine has actually been changed to the 13th December at 6.30pm – but will wait for official confirmation), as I'll need to find the time, as I currently have 4 projects which need a lot of work to be done on each of them.

25/11

Met my supervisor last night and got his train of thought in to what I'll need for my presentation. The UI is not terribly important for it; he does want the main functionality working. I'll need to get the main booking page working from what the user selects on the main page. He also wants the notification email working. I think I can get it done in time. My presentation is on the 13th of December, a week before everyone else, so a little less time to prepare.

30/11

Haven't touched the project for a few days – have 3 other projects still ongoing and finding the time for everything can be tough. I'll finish out this week doing those other projects, and then begin early next week with what I need for this presentation. I do have an idea of how I'm going to code it, but it remains to be seen how it goes. Two and a half weeks left.

5.4.4 December

05/12

This is the week I've allocated to do a decent prototype. I started tonight but I didn't make much progress, not sure if I was tired or didn't prepare enough for what I

wanted, but it did not go well. I'll have to give it another crack tomorrow evening, as it annoyed me that the initial part didn't work.

06/12

I had to make the decision to skip both classes tonight, and put the head down in the library in order to get some of this done. And, lo and behold, my decision paid off. I made some excellent progress, albeit still a good bit of work to do before I'm ready for the presentation. I got some of the initial steps done, which make it easier to build on and progress further. I think my presentation is 10 minutes of me talking, 10 for them to ask questions, and that's all. Manuel said he doesn't mind about the UI itself, so I'm not going to spend long on that. That's for the new year.

07/12

I did a bit more work on the prototype this evening, but I don't think I have enough ready to demo to Manuel during our meeting tomorrow. I've documented on paper what I hope to achieve before Tuesday, so that will be my target. Hopefully, it will be enough.

08/12

Met Manuel this evening. I showed him what I had written last night as my plan for the presentation. He seemed pretty happy with it, bar he wanted one bit of extra functionality; some sort of validation when a customer tries to book a slot that's already booked. At this stage, I'm not sure I'll get that far, as I have to do all my other bits before I get to that point. I'll try though – but I have an exam on Saturday morning and a technical report due by Sunday night, so I'll see how much I can squeeze in. We were initially told that no coding would be involved this side of Christmas, but here we are and I need to do quite a bit, so it's pretty tough at the moment. Saturday week cannot come around quick enough.

11/12

Spent a lot of time today working on my prototype; it's been a while since I used PHP so it took a lot of messing around to get it working. I had to work on my technical report too today, so couldn't dedicate the whole day to it. I have a chunk

of the core functionality working at this stage, albeit a bit messy on the coding side, but that's ok for now. I've to prepare slides and the presentation tomorrow so I hope the two lecturers will be happy with what I present.

12/12

I prepared my slides and went over what I was going to say in my presentation. It feels like it'll only last about 10 minutes, even though everyone is allocated 20 minutes to speak, before 10 minutes of questions and feedback. I tried to tidy up my code a little (not sure why as no-one will be looking at it yet), but it actually didn't go too well so I left it as it was. I added some more functionality, such as a login page, and some validation behind the scenes on email addresses or if the user leaves one of the mandatory fields blank. Overall, it looks decent enough so I'm happy with my progress on it.

13/12

Presentation day. I didn't get a lot done in work as I was going over what I wanted to say. I'm not nervous about these types of things, although I do like to know quite well everything I will be saying. I think I'll be pretty well prepared.

14/12

The presentation went pretty well I think! I talked about my ideas and what I've done, and demonstrated some of the core functionality I had written over the last few days. I got some feedback from the two lecturers, but none of it was negative, just more ideas to add to the project e.g. social media interaction, payment option on the page, change email notification to an SMS, reverse the calendar (i.e. have all times selected so the user only has to uncheck the ones they don't want), and to allow the user to style the page themselves. Bar the last one, I think I can manage them all, as I do have a good bit of time now to do it. When I was leaving, I glanced at the clock and it took the half an hour. That went fast. I only have one project left to complete, and to submit three others that are already completed, so I'm amazed I actually got this far to be honest. I'm still on track for a first, but

obviously need a pretty good score in all three January exams. That's this log done until the new year as I will be concentrating on my exams for the next 4 weeks.

5.4.5 January

24/01

The exams over Christmas went well, I think. 2 out of 3 of them had changed the format without telling us, so that was an unwelcome curveball, but still pretty sure I managed them alright. We also have one project result due at the same time as the exams; next Friday.

First night back in tonight was grand. We had our project class and no deliverables (bar these journals) until the end of the semester. I have to organise to meet my supervisor again, so I'll email him as soon as I can get on to my laptop. The second class was Usability Design and after one lecture, it seems like it's a manageable class. Two less subjects this semester, so hopefully that will make it a little easier than the last one, as that was very intense.

It's time to start working on the project now though, and not leave it until the last minute. I might just get through this week of lecturers and see what all the new subjects are like first.

29/01

Finally started work on the project. I started with the first requirement from my documentation; registration. I spent a bit of time and the functionality is almost there, barring a few minor issues but ones which won't affect it enough for now. I also emailed Manuel yesterday for our next meeting, and am awaiting a response. I won't get time tomorrow after work for any project stuff, so will get an hour or two before college on Tuesday hopefully.

31/01

Finished work earlier than expected so headed into college to get more work on the project. After a little while, I got the login page working, and correctly calling the users credentials from the database. I also put in some links between pages and just set a few dummy pages for now.

I have been emailing Manuel back and forth trying to sort out when our next meeting will be. He offered lunch time Monday, in person or by conference call, or else Wednesday evening. I will get back to him tomorrow once I figure out which suits best.

5.4.6 February

02/02

I'm currently working on the admin user's set up page, as per the requirements. As PHP is a self-taught language, I do hit a lot of speedbumps at times. Initial tests done were just to update what the user selects, and see will it add to a test database. After a bit of messing around with variables, it is working so I'll need to amend it to actually do what I want with it now.

06/02

More work done tonight on the setup page. At the moment, the user selects their area which will set up a table for that in the database. Then the user needs to select the days available. Whichever days they choose, a column in the database updates to 'true', so that these will only be the days displayed on the booking page for the customer. Next step is to work on the times available.

09/02

Worked on the PHP setup again after college. Two things; as there are two setup pages, I forced the user to have to go through the first page, in order to reach the second page. If they try navigating to the second page via the browser, it will redirect them back to the first page. The second thing was I tidied up the code for

the days chosen – it's now in a nice loop rather than making individual calls for each day chosen. Not sure if I will get time over the weekend to add more, but I'll see how college goes on Saturday.

12/02

I have completed a lot of the functionality for the setup page. The admin user can successfully choose the area, day and time available and the system will display the areas in a nice table. I have put in code to allow the admin to delete the entries if needed, and the next step is to allow them to edit their selections. They can also add in additional areas entirely, but no code there yet to display the multiple tables, based on the selection. I have a list of things I've completed for the admin user and will bring to my meeting with Manuel tomorrow to see if I'm missing anything.

14/02

I worked on the update section tonight. Basically the admin user can edit their selections if they change their mind. Currently, they can only add or remove a time previously selected but adding or removing a day is proving a bit trickier – I'll work on that again on Thursday. I also need to work on the admin page, i.e. update email address etc., and then that's the majority of the admin functionality in place. There is still a good bit to do though, even on those pages, such as making the proper links between pages, and styling the page to look how I wish it to look.

16/02

I have put together some code for the updating of the chosen selections – it's not perfect yet so will need some fine-tuning. I have also put together a survey for different parts of the site, to see what the public think and I will base certain aspects from that survey.

20/02

Worked on the customer booking page. It's only partially done but has given me an insight on how I'll proceed with it. I met Manuel today and he didn't have the same setup as me so couldn't view what I've done on localhost, so I'll need to bring my laptop to the next meeting. He also made a point which surprisingly I hadn't

thought of (although I'm not at that part of the project yet), which is a page where the admin user can see the bookings submitted, and by whom.

21/02

I spent the evening looking into APIs that will allow my site to send SMS, but generally they cost money to set up. I have also started looking at allowing my site to send emails, as they will be free. I think I'll let my survey do the talking before I make a final decision.

23/02

I put 5 questions in the survey, and set the maximum answers to 25. I sent it out and pretty much got the answers within a few hours so I'm all set for decisions on how I'll code certain parts of the application. Some surprising results, including email responses being much more popular than SMS.

26/02

For the past few days, I've been chipping away at getting my site to send emails. I have gotten it to send an email alright (on localhost, which was proving a bit tricky), but just need to sort out the content and ensure the booking reference number gets sent correctly. I hope to finish this pretty soon, and ensure the booking details are saved for the admin user to view and delete if necessary. I also need to bring in PayPal functionality to allow the users to pay for their booking in advance.

5.4.7 March

05/03

I made lots of changes over the last week to the functionality. The email functionality now sends as expected and stores a random booking reference number. I tidied up the navigation bar on the top of each page, but this still has a little work to be done on it. As per the survey results, only two users can register, otherwise an error will display. I added in the option to select the cost of the area

per hour, as a start for when I implement the ability to pay for what is hired. The user can only enter numeric values in this field also. I added a new account, i.e. for a different club and I put this online for an initial test to see if the web hosting site works. I removed the MySQL references and changed them to MySQLi or PDO, depending on what I needed them for, as MySQL is deprecated in PHP7 and higher.

I emailed the link to the temporary online site to Manuel for review, so I'll wait to see what he comes back with.

13/03

I haven't heard back from Manuel regarding feedback of what I've done so far. In the meantime, I've put more work into the page. I've added a PayPal button (only a test account for now), and customers can now pay for their booking which was one of the changes I was asked to complete at the mid-point presentation.

I also began tidying up the code – although I will leave most of the tidy up until the last few weeks as it's on the lower end of the importance list right now. I fixed a few minor bugs also as it's getting near the end. I put in a small footer too, as the webhost account I'm using puts an ad right at the bottom which covers a bit of the page including some of the buttons. This will get sorted once the styling is updated.

I'm not sure if I'll have a meeting with Manuel today as it is reading week, but I've sent him an email to double check.

15/03

Manuel couldn't make the meeting this week as he is busy grading CAs and setting up summer exams. I haven't really made any progress in the last few days – bar making a checklist of remaining items. I feel the progress has gone well so far, just need to knuckle down for the last 7 or so weeks. I have other projects which are due much sooner, so I have to concentrate on them also.

I haven't received feedback on the progress just yet, so waiting on Manuel to get back to me this week.

20/03

Over the long weekend, I put in a lot of work for some more changes to the application. I added in the option for users to cancel their bookings. This took a lot of work but is working as expected now. I changed around the navigation bar, depending on whether a user is logged in, or if it's just a customer who doesn't need to log in. I added a share button for Facebook and for Twitter – and they work nicely.

I requested feedback from a few people and got some interesting insight. I have a put a few ideas together and will try implementing them over the next few weeks.

22/03

I got some feedback off Manuel and he had a few valid points to make. I have implemented the following over the last two days:

- Add/Remove banner – to allow the company to somewhat personalise it
- Added 20 new areas
- Fixed several navbar bugs
- Added prompt when attempting to logout
- Added contact information on booking email
- Added tooltips for additional help
- Added new images to admin and welcome page

Not long to go now, and I feel I'm in a good place with the project. I have a few more weeks and still have a good bit of work but I feel I've made some excellent progress.

23/03

I added some nice features today. The page of all bookings has a new filter so the admin user can quickly type a partial name and all matches in the table will be the only ones displayed. This will come in handy when there are plenty of bookings and they need to search for a specific one. I also added a 'select/deselect' all button when setting up an area. Coupled with this, I added a second button that will only select the early times, and another that will only select the late times. This could save the admin user a bit of time. I also fixed a couple of small bugs that I'd noticed.

26/03

I changed the search around to a nicer version which searches all part of the table, and filters it nicely. I also separated all the JavaScript into its own files – instead of inline. Something went wrong however, and I had to revert these changes. But it turned out I made a mess of something else, so I reverted all the reverts! There's not much more functionality I'll be adding at this stage as I need to do my testing and technical report.

29/03

From here until submission date, I will be just concentrating on the testing and the technical report. The only changes made will be styling changes and any bug fixes as required during the testing phase. If (big if) there is any more time, I will try implement any functionality that I may have missed, but for now I don't foresee much free time.