

# SOFTWARE ENGINEERING PROCESSES AND TOOLS

## PART B - USER DOCUMENTATION

---

### CONTENTS

<b>Installation and Setup Guide of XAMPP and Database</b>	<b>1</b>
1.0 Prerequisite	1
1.1 Downloading the XAMPP server stack	1
1.2 Installing the XAMPP server stack	1
1.3 Starting the XAMPP server stack	2
1.4 Setting up the web server directory	3
1.5 Setting up the database	3
1.6 How to access Booking Website web page	5
<b>Page Descriptions and Functions</b>	<b>6</b>
2.1 Main Page: index.php	6
2.2 Initial Customer Page: customerPage.php	7
2.3 Customer view/make a Booking: customerBooking.php	8
2.4 Initial Business Owner Page: businessPage.php	10
2.5.0 New Bookings Page: businessPageNewBookings.php	12
2.5.1 Create Booking Page: businessPageCreateBookings.php	13
2.6 Availability: businessPageEmployeeAvailability.php	13
2.7 Add Shift: businessPageEmployeeAddShift.php	16
2.8 Edit/Delete Shift: businessPageEmployeeEditShift.php	17
2.9 Add Worker: businessPageEmployeeAddEmployee.php	19
3.0 Viewing Data logs	19

# 1. Installation and Setup Guide of XAMPP and Database

## 1.0 Prerequisite

**Operating system:** Microsoft Windows

## 1.1 Downloading the XAMPP server stack

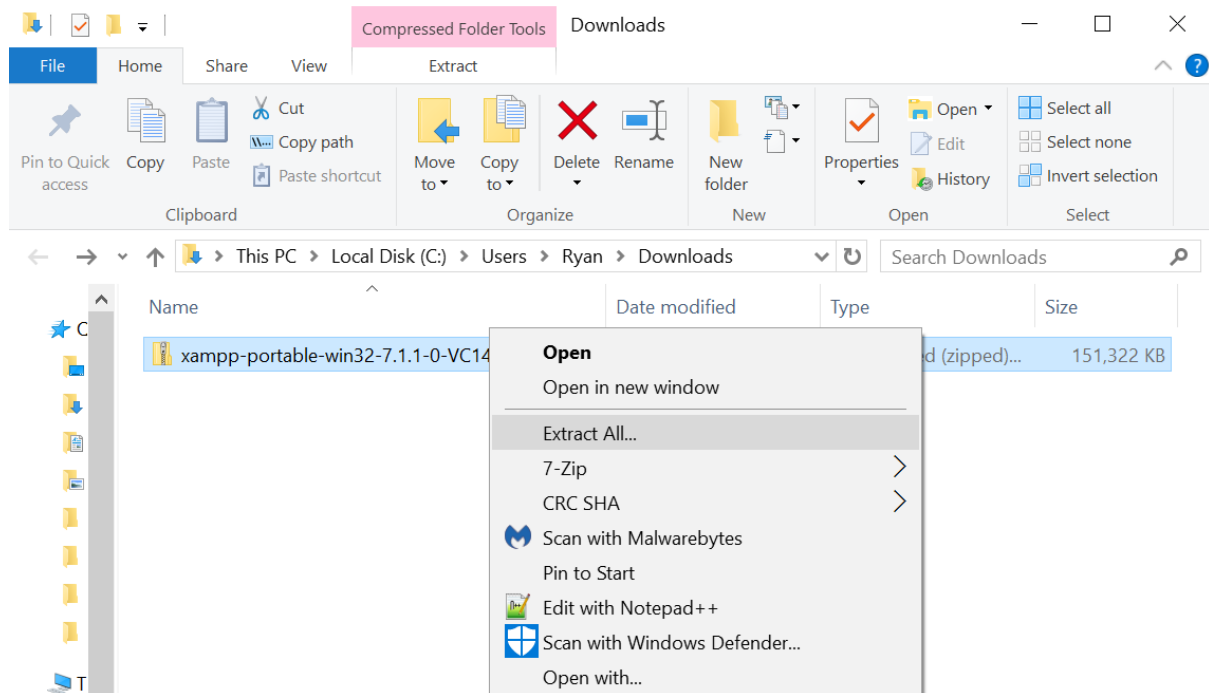
This package is the **XAMPP for Windows 7.1.1**, which includes (**Apache 2.4.25**, **MariaDB 10.1.21(MySQL)**, **PHP 7.1.1**), the services we need.

Download XAMPP portable(Non-installation required) package from the link below:

<https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/7.1.1/xampp-portable-win32-7.1.1-0-VC14.zip/download>

## 1.2 Installing the XAMPP server stack

Extract the downloaded compressed file into the root directory of your partition.





← Extract Compressed (Zipped) Folders

## Select a Destination and Extract Files

Files will be extracted to this folder:

C:\

Browse...

☒ Show extracted files when complete

Extract

Cancel

## 1.3 Starting the XAMPP server stack

Once extracted, go to the extracted package, directory ("C:\xampp", if it was extracted in root directory of partition), and run the program "**xampp-control.exe**", Select the English option and save.

- Start **Apache** and **MySQL** services.

XAMPP Control Panel v3.2.2 [ Compiled: Nov 12th 2015 ]

The XAMPP Control Panel v3.2.2 interface displays a list of modules and their status. The 'Stop' button for MySQL is highlighted. The log window shows the following messages:

Time	Module	Message
3:05:08 PM	[main]	All prerequisites found
3:05:08 PM	[main]	Initializing Modules
3:05:08 PM	[main]	The FileZilla module is disabled
3:05:08 PM	[main]	The Mercury module is disabled
3:05:08 PM	[main]	Starting Check-Timer
3:05:08 PM	[main]	Control Panel Ready
3:05:13 PM	[Apache]	Attempting to start Apache app...
3:05:13 PM	[Apache]	Status change detected: running
3:05:45 PM	[mysql]	Attempting to start MySQL app...
3:05:47 PM	[mysql]	Status change detected: running

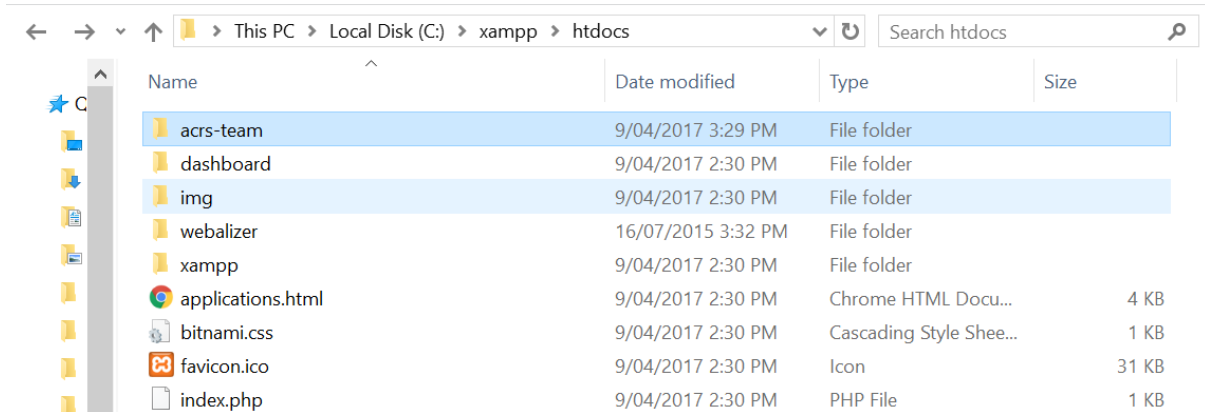
---

## 1.4 Setting up the web server directory

Back to the XAMPP folder, locate the **htdocs** folder, this is where we will be putting the assignment repository.

- Extract or move the entire assignment folder into the **htdocs** folder.

In this demonstration case, the assignment folder is called **acrs-team**.



This PC > Local Disk (C:) > xampp > htdocs					Search htdocs	
	Name	Date modified	Type	Size		
	acrs-team	9/04/2017 3:29 PM	File folder			
	dashboard	9/04/2017 2:30 PM	File folder			
	img	9/04/2017 2:30 PM	File folder			
	webalizer	16/07/2015 3:32 PM	File folder			
	xampp	9/04/2017 2:30 PM	File folder			
	applications.html	9/04/2017 2:30 PM	Chrome HTML Docu...	4 KB		
	bitnami.css	9/04/2017 2:30 PM	Cascading Style Shee...	1 KB		
	favicon.ico	9/04/2017 2:30 PM	Icon	31 KB		
	index.php	9/04/2017 2:30 PM	PHP File	1 KB		

---

## 1.5 Setting up the database

1. Open a Command prompt, and change directory to "**c:\xampp\mysql\bin\**", if xampp was extracted into the root directory, this can be done using the command
  - "**cd c:\xampp\mysql\bin\**", without the quotes.

```
C:\Users\Ryan>cd c:\xampp\mysql\bin\
```

2. Run mysql.exe with the root username, using
  - "**mysql.exe -u root**" without quotes.

The command prompt will return the below message, if successfully logged into the database.

```
c:\xampp\mysql\bin>mysql.exe -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.21-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

3. Locate the sql script called "**createTablesAndInsertDummyRecords.sql**", used to setup the database, tables and insert dummy records into the tables, located in the assignment folder under the "sqlScripts" folder.

("C:\xampp\htdocs\acrs-team\sqlScripts", if it was extracted in root directory of partition)

4. Run the **source** command, used to execute, sql scripts in MySQL, with the absolute file path of the sql script "**createTablesAndInsertDummyRecords.sql**". In our demonstration case, it would be

- **"source**

**C:\xampp\htdocs\acrs-team\sqlScripts\createTablesAndInsertDummyRecords.sql**", without the quotes.

If successful the script will process, create the database, tables, and insert dummy records.

```
MariaDB [(none)]> source C:\xampp\htdocs\acrs-team\sqlScripts\createTablesAndInsertDummyRecords.sql
Query OK, 6 rows affected (0.09 sec)

Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.06 sec)

Query OK, 0 rows affected (0.04 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)
```

## 1.6 How to access Booking Website web page

1. Open a browser of choice, In our case it is Google Chrome.
2. In the URL go to - [http://localhost/<What\\_you\\_named\\_the\\_assignment\\_folder>](http://localhost/<What_you_named_the_assignment_folder>)
  - If XAMPP was extracted into the root directory and the assignment folder was placed in “c:\xampp\htdocs”, enter into the URL:
    - <http://localhost/acrs-team/>
3. The webpage will be redirected to a page similar to below.

The screenshot shows a web browser window with the URL [localhost/acrs-team/](http://localhost/acrs-team/). The page has a light blue background with a cloud pattern. At the top, it says "Appointment Booking System". Below this, there is a message: "Customers - 'Login' to see booking status and to make new bookings. If you do not have an account, please 'Register' and then login. Owners - 'Login' to see more functionalities." There are two main forms: "Login" and "Customer Register". The "Login" form has fields for "Username:" and "Password:" and a "Login" button. The "Customer Register" form has fields for "Username:", "Password:", "Re-enter Password:", "Full name:", "Address:", and "Phone Number:", and a "Register" button.

4. You can test whether the database has successfully been set up by login into the owner account using (Username: admin / Password: admin), and you'll be redirected to the Business owners, administration page.

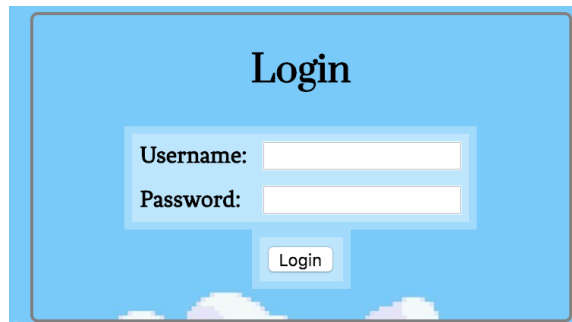
The screenshot shows a web browser window with the URL [localhost/acrs-team/businessPage.php](http://localhost/acrs-team/businessPage.php). The page has a light blue background. At the top, it says "Booking Website". Below this, there is a greeting "Hello, admin." and two dropdown menus labeled "Bookings" and "Employees", followed by a "Sign Out" link. In the center, there is a section titled "Booking Summaries" containing a table with booking data.

Date	Start Time	End Time	Customer Name	Service	Extra Notes
17-April-2017	9:00AM	10:00AM	Chris Hunter		Wash & Dry
17-April-2017	11:00AM	12:00PM	Gary Peterson		Clip, Wash & Dry
18-April-2017	11:30AM	12:30PM	Horton Andersen		Clip & Style

## 2. Page Descriptions and Functions

### 2.1 Main Page: index.php

This page is the first page that should be accessed, it contains the customer and business owner login section, as well as the customer register form.

A screenshot of a login form titled "Login" in a large, bold, black serif font. The form is set against a light blue background with a subtle cloud pattern at the bottom. It contains two input fields: "Username:" and "Password:", both with white text and light blue borders. Below these fields is a "Login" button with a light blue border and a small shadow effect.


**Figure 1 - Login Form**

The login form works for all entries inside the database (including those that have been added through the customer registration form); however the default details that should be used for testing purposes are:

<b>Customer:</b>	username:	customer
	password:	customer

<b>Business Owner:</b>	username:	admin
	password:	admin

Once the user has entered in the username and password they desire, they can either press the "enter" key or click the "login" button to begin processing the login request.

A screenshot of a customer registration form titled "Customer Register" in a large, bold, black serif font. The form is set against a light blue background with a subtle cloud pattern at the bottom. It contains six input fields: "Username:", "Password:", "Re-enter Password:", "Full name:", "Address:", and "Phone Number:", all with white text and light blue borders. Below these fields is a "Register" button with a light blue border and a small shadow effect.

**Figure 2 - Customer Register Form**

For the customer register form accept a new registration.

- All five fields must have entries
- The username must be a unique value
- The username has to consist of letters or digits only and length between 5-10

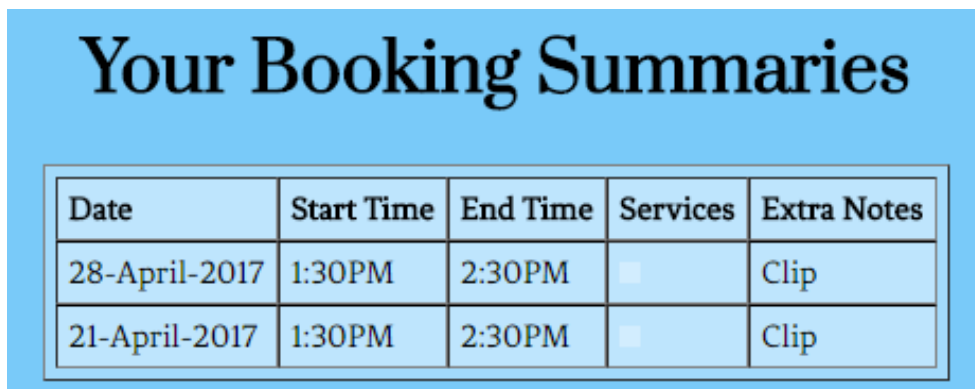
- The password must be at least 8 characters contain at least 1 uppercase letter, 1 lowercase letter and 1 number, it can contain special characters
- The passwords have to match (password <-> re-enter password)
- Full name must contain letters only and length between 5-10
- Phone number must contain 10 digits (acceptable formats are -> 123.456.6789 and 123-456-6789 and 1234567891)

Once the user is happy with their input they must click the “register” button for the form to start processing their input.

## 2.2 Initial Customer Page: customerPage.php

This is the first page that a customer will see after they log in. The central focus of the page is the bookings that have been made by the account that is currently logged in. There is no interaction with this part of the page, however.

(The services column will generally show the actual service name eg: clip, wash, etc. However the test data that was used did not include this column, as such they are not shown here)



Date	Start Time	End Time	Services	Extra Notes
28-April-2017	1:30PM	2:30PM		Clip
21-April-2017	1:30PM	2:30PM		Clip

**Figure 3 - “customer’s” bookings summary**

Along the header of the page there are three buttons: “Show Booking Summaries”, “Make a new Booking”, and “Sign Out”. (Please note, “Hello, customer” is not a button, it just uses the same style as the rest of the navigation bar).



# Booking Website

Hello, customer.   Show Booking Summaries   Make a new Booking   Sign Out

**Figure 4 - Main header of the customer section of the website**

The “Show Booking Summaries” button leads the user to **2.2** “customerPage.php”, “Make a new Booking” takes the user to **2.3** “customerBooking.php”, and “Sign Out” logs the user out of the current session, and takes them back to **2.1** “index.php”. This header is consistent throughout the entire customer section of the website and will not be re-explained for all pages.

## 2.3 Customer view/make a Booking: customerBooking.php

This is one of the most important pages on the website, it allows the customer to actually make a booking into the system.

The screenshot displays the 'Make a booking' page. On the left is a calendar for April 2017. The calendar has a header with 'today' and a right arrow, and 'month' and 'day' buttons. The days of the week are listed as Sun, Mon, Tue, Wed, Thu, Fri, Sat. The dates are arranged in a grid. The date 29 is highlighted in yellow. On the right is a form titled 'Make a booking'. It contains a list of instructions: 1. Select a date in the calendar and view available slots. 2. Enter in booking time below. 3. Click the checkboxes for what tasks you would like done. 4. Choose an employee if you have a preference. 5. Enter in extra requests. 6. Book appointment. The form fields include: Date (text input), Start Time (text input), End Time (text input), Activity (checkboxes for Clip, Wash & Dry, Style), Employee: (optional) (dropdown menu with 'Any Available' selected), Extra notes (text area), and a 'Book Appointment' button.

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

**Make a booking**

1. Select a date in the calendar and view available slots.
2. Enter in booking time below.
3. Click the checkboxes for what tasks you would like done.
4. Choose an employee if you have a preference.
5. Enter in extra requests.
6. Book appointment.

Date:

Start Time:  End Time:

Activity:

Clip ☐

Wash & Dry ☐

Style ☐

Employee: (optional)

Any Available ▼

Extra notes:

Enter any other special requests...

Book Appointment

**Figure 5 - “Making a Booking” page’s main content**

The main content of this page consists of a calendar, that has four buttons on the top (two in the top left corner which are used for view navigation, and two in the top right corner which

change the calendar's view from a monthly overview to a daily overview). The way this page works is somewhat convoluted. Off to the right of the calendar there is a form with four fields that need to be filled in. The "Date" field will be filled in automatically whenever the user clicks on a day on the main calendar, at the same time as this the view of the calendar will shift from a month view to a timeline of the day they have clicked on. This will show all of the appointments that have been made on that day, as well as the times that are free(White).

The screenshot displays a web interface for a booking system. On the left, a calendar view for Saturday, April 29, 2017, shows a timeline from 9am to 5pm. The timeline is currently empty, with white slots indicating available time. To the right of the timeline is a 'Make a booking' form. The form includes a list of instructions: 1. Select a date in the calendar and view available slots. 2. Enter in booking time below. 3. Click the checkboxes for what tasks you would like done. 4. Choose an employee if you have a preference. 5. Enter in extra requests. 6. Book appointment. The form fields are: Date (29/04/2017), Start Time (---:--), End Time (---:--), Activity (Clip, Wash & Dry, Style), Employee (optional) (Any Available), and Extra notes (Enter any other special requests...). A 'Book Appointment' button is at the bottom of the form.

**Figure 6 - Daily Timeline view of the calendar, showing the date has been entered**

Entering in the start time ate; however, it is much more tedious as the user will have to enter in the time specifically. The end time will be automatically updated depending on what activities are selected, as each require a different amount of time. The next drop down menu allows the customer to select the employee that they would like to work with for their booking, if they want. Selecting "any available" will just find any employee available and assign them to the booking.

The last field is simply a text box for any "extra notes" that may need to be submitted with the booking.

A screenshot of a web form for booking an appointment. The form has a light blue background with a subtle cloud pattern. It contains several input fields and checkboxes. The 'Date' field is set to '29/04/2017'. The 'Start Time' and 'End Time' fields are set to '10:00' and '11:00' respectively. The 'Activity' section has three checked checkboxes: 'Clip', 'Wash & Dry', and 'Style'. The 'Employee: (optional)' dropdown menu is set to 'Any Available'. The 'Extra notes' field contains the text 'Enter any other special requests...'. At the bottom, there is a 'Book Appointment' button.

**Date:**  
29/04/2017

**Start Time:** 10:00 **End Time:** 11:00

**Activity:**  
Clip ☒  
Wash & Dry ☒  
Style ☒

**Employee: (optional)**  
Any Available ▼

**Extra notes:**  
Enter any other special requests...

**Book Appointment**

**Figure 7 - Showing the Start time, end time, and extra notes sections being filled in.**

Once the user has finished entering in the details of their desired appointment they can click on the “Book appointment” button to actually book it in (assuming it is a valid appointment).

## **2.4 Initial Business Owner Page: [businessPage.php](#)**

This page is very similar to the initial customer page **2.1** however rather than just showing the bookings for just the currently logged in user, it shows all of the bookings currently in the system for their business (sorted by time, earliest to latest). There is no interaction except for navigation on this page.

Booking Summaries					
Date	Start Time	End Time	Customer Name	Service	Extra Notes
17-April-2017	9:00AM	10:00AM	Chris Hunter		Wash & Dry
17-April-2017	11:00AM	12:00PM	Gary Peterson		Clip, Wash & Dry
18-April-2017	11:30AM	12:30PM	Horton Andersen		Clip & Style
19-April-2017	1:00PM	2:00PM	Bernard Campbell		Clip, Wash & Dry
21-April-2017	10:45AM	11:45AM	Isabelle Barkley		Wash & Dry
21-April-2017	1:30PM	2:30PM	Alica Michaels		Clip
24-April-2017	9:00AM	10:00AM	Chris Hunter		Wash & Dry
24-April-2017	11:00AM	12:00PM	Gary Peterson		Clip, Wash & Dry
25-April-2017	11:30AM	12:30PM	Horton Andersen		Clip & Style
26-April-2017	1:00PM	2:00PM	Bernard Campbell		Clip, Wash & Dry
28-April-2017	10:45AM	11:45AM	Isabelle Barkley		Wash & Dry
28-April-2017	1:30PM	2:30PM	Alica Michaels		Clip

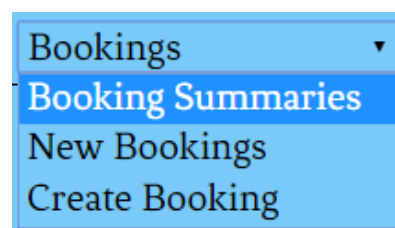
**Figure 8 - Booking Summaries for business Owner**

The main header for the Business Owner section of the website is very similar in appearance to the customer's version but with a few major functionality changes. To begin with, the centre two buttons are now drop down menus with more options.



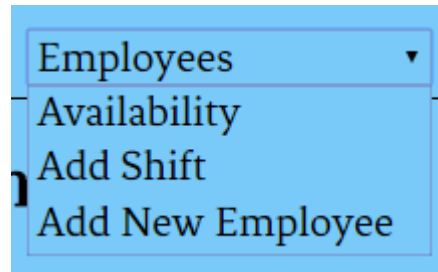
**Figure 9 - Business Owner Section Header**

The “Bookings” drop down menu holds only two items: “Booking Summaries” which leads to the initial “businessPage.php” **2.4**. Whereas the “New Bookings” option yields a new result, leading the user to “businessPageNewBookings.php” **2.5.0**. The newly added “Create Booking” option takes the user to “businessPageCreateBooking” **2.5.1**.



**Figure 10 - “Bookings” Drop down menu options**

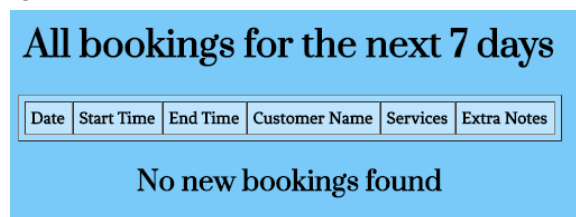
The “Employees” drop down menu holds three completely new options: “Availability” which takes the user to “businessPageEmployeeAvailability.php” **2.6**, “Add Shift” redirects the user to “businessPageEmployeeAddShift.php” **2.7**, and lastly “Add New Employee” leads to “businessPageEmployeeAddEmployee.php” **2.9**.



**Figure 11 - “Employees” Drop down menu options**

### 2.5.0 New Bookings Page: businessPageNewBookings.php

This page is remarkably simple, again having no interaction beyond navigation; however, it serves an important purpose: showing all bookings that have been currently made for the next seven days, showing the earliest first.



**Figure 12.1 - New Bookings format changes, does not have test data inside**

All bookings for the next 7 days					
Date	Start Time	End Time	Customer Name	Services	Extra Notes
30-April-2017	9:00AM	9:28AM	Gary Peterson	Clip Style	
30-April-2017	10:00AM	10:57AM	Isabelle Barkley	Clip Wash & Dry Style	
30-April-2017	11:00AM	11:57AM	Gary Peterson	Clip Wash & Dry Style	asdasda

**Figure 12.2 - All Bookings for the Next Seven Days (old formatting but with test data)**

## 2.5.1 Create businessPageCreateBookings.php

## Booking

## Page:

This page functions identically to 2.3 customerBooking, except that the business owner can select a customer to make the booking for.

today > April 2017 month day

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

**Make a booking**

1. Select Customer.
2. Select a date in the calendar and view available slots.
3. Enter in booking start time.
4. Select services.
5. Enter in extra requests.
6. Book appointment.

**Customer Name:**  
Select Customer ▼

**Date:**  
[Date Input]

**Start Time:** [Time Input] **End Time:** [Time Input]

**Services:**  
Clip ☐  
Wash & Dry ☐  
Style ☐

**Employee: (optional)**  
Any Available ▼

**Extra notes:**  
Enter any other special requests... [Text Area]

**Book Appointment**

Figure 13 - Business Owner Make a Booking for Customer Page

## 2.6 Availability: businessPageEmployeeAvailability.php

This page is another rather complex page with a lot of interactivity. The calendar itself has six buttons:

The top left three are again for date navigation. If you are in month view; the left arrow shifts the view one month backward, and the right arrow shifts the view one month forward; if you are in week view, they each shift the view one week in the same direction; and the same goes for the day view. The middle button “today” just shifts the view to the current date.

The top right three change the view mode to the selected one, either month, week, or day.

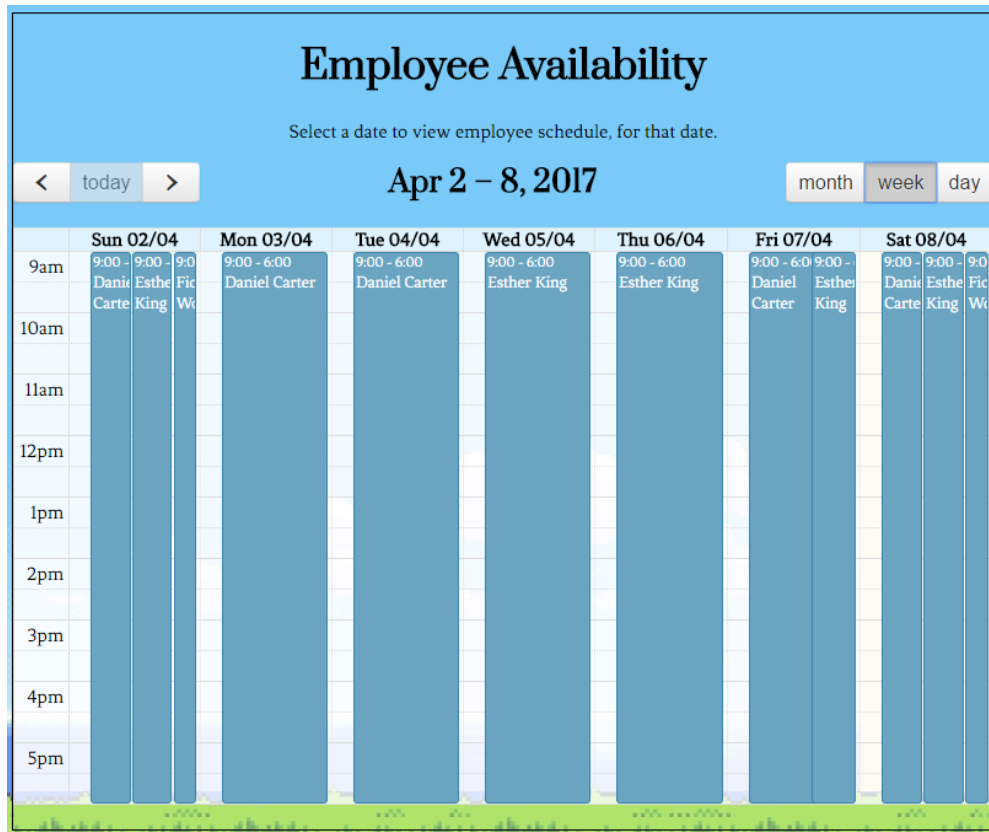


**Figure 14 - Employee Availability Month View**

If a day is clicked in either month view, or week view, then the view is changed to focus only on the selected day, and the actual employee working times are shown. In week view, the working times for each employee are also shown.

In the future it may be wise to have some sort of menu where the user can select which employees they want to the shifts for, as currently it just shows all of them. If there were more than three or four employees rostered on at the same time it may become too cluttered to be useful, and most businesses have more than four employees working at the same time.





**Figure 15 - Weekly view of Employee Availability**



**Figure 16 - Daily View of Employee Availability**



## 2.7 Add Shift: businessPageEmployeeAddShift.php

This page and its main functionality of adding a shift for an employee works nearly identically to the “Make a Booking” 2.3, except you need to select the employee whose shift you are adding from a drop down menu on the right hand side form.

ie: Clicking on a day in the calendar will automatically add it in the form, but the rest must be inputted manually.

This page also takes in functionality from 2.6 in that all of the current shifts (for all employees) can also be seen in both the weekly view and daily view. These are identical to figure 14 and figure 15 so they are not included here as well.

< today > April 2017 month week day

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

**Set a Shift**

1. Select a date in the calendar
2. Enter in shift time below.
3. Set shift.

**Employee Name:**  
Daniel Carter ▼

**Date:**

**Start Time:**  
-- : --

**End Time:**  
-- : --

Figure 17 - Adding a new shift for an employee main content

## 2.8 Edit/Delete Shift: businessPageEmployeeEditShift.php

The screenshot shows a web interface for editing shifts. On the left is a calendar for April 2017. The calendar has columns for days of the week (Sun to Sat) and rows for dates. Each date cell contains a list of shifts for that day, such as '9a Daniel Ca', '9a Esther Kir', and '9a Fiona Wo'. The date 25th April is highlighted in yellow. On the right is a form titled 'Edit a Shift'. It contains a list of instructions: 1. Select employee to show scheduled shifts. OR 2. Select a date in the calendar. 3. Enter in new time below. 4. Edit or Delete shift. The form has fields for 'Employee Name' (a dropdown menu), 'Date' (a text input), 'Start Time' (a time input), and 'End Time' (a time input). At the bottom of the form are two buttons: 'Edit Shift' and 'Delete Shift'.

**Figure 18 - Editing shift for an employee main content**

This page allows the owner to edit or delete the existing employee shift(s). All of the existing shifts can be seen on the page and once one of them is clicked, it grabs the data from that shift and it appears on the right table. (yellow box shows today's date)

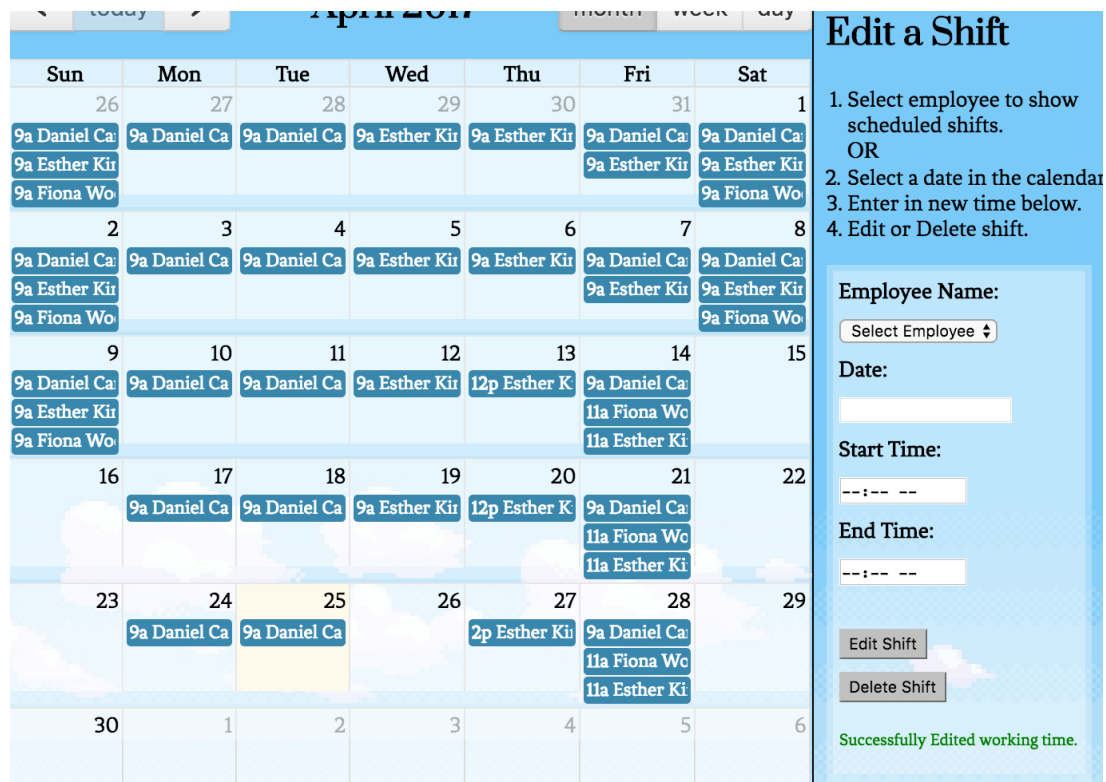
This screenshot shows the 'Edit a Shift' form before a shift has been selected. The 'Employee Name' field is a dropdown menu with 'Select Employee' as the placeholder. The 'Date' field is empty. The 'Start Time' and 'End Time' fields are also empty, showing the default time format '---:-- --'. The 'Edit Shift' and 'Delete Shift' buttons are at the bottom.

**Figure 19 -  
Before clicked  
on a shift**

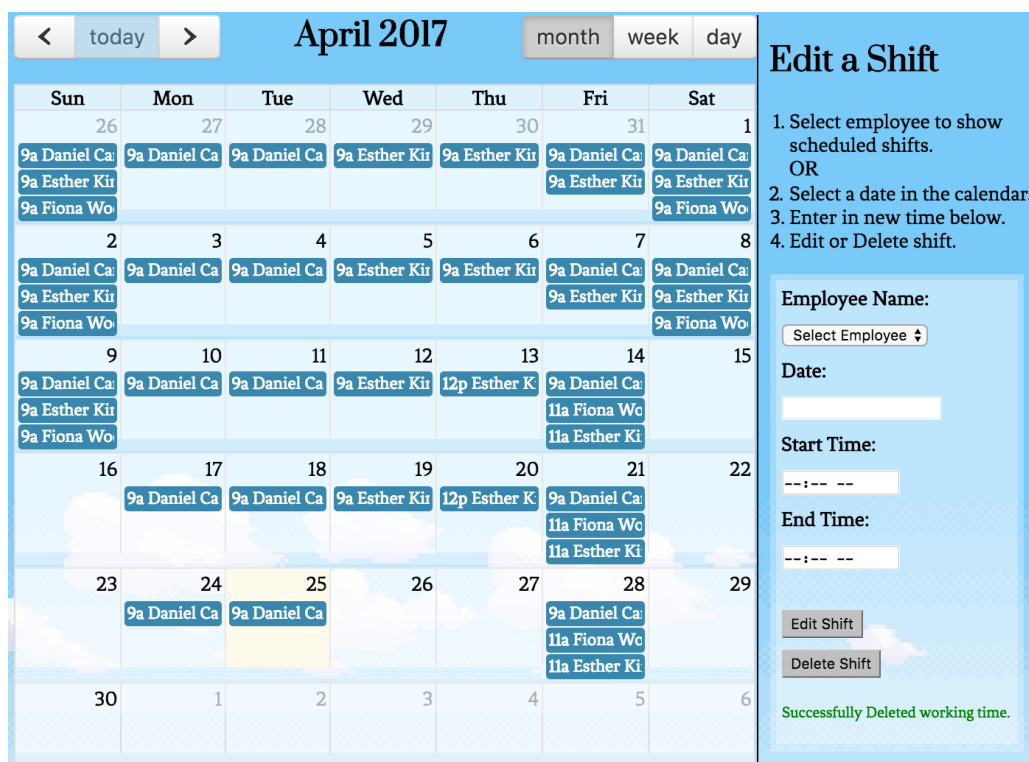
This screenshot shows the 'Edit a Shift' form after a shift has been selected. The 'Employee Name' field is a dropdown menu with 'Select Employee' as the placeholder. The 'Date' field is filled with '27/04/2017'. The 'Start Time' field is filled with '12:00 PM' and the 'End Time' field is filled with '06:00 PM'. The 'Edit Shift' and 'Delete Shift' buttons are at the bottom.

**Figure 20 -  
After clicked  
on a shift**

In this example, 27th April (12p Esther K) is clicked. Therefore, fields below "Employee Name" at the right shows us the date, start and end times of that shift. This can be edited by simply just changing the fields of these boxes followed by clicking "Edit Shift". Once it has been changed, Success message pops up and calendar gets refreshed.



This is how calender looks like after editing a shift. 27th shift which belongs to Esther is now 2pm-5pm.



Similarly this chosen (27th Esther K) shift can be deleted by simply clicking “Delete Shift”. After clicking, success message pops up and calendar gets refreshed.

## 2.9 Add Worker: `businessPageEmployeeAddEmployee.php`

This final page is very simple, just a three text field form that is used to add a new employee to the database. All three fields must have data and the employee number must also be unique before the adding process will succeed. Success will render a message saying it is successful; however it may be useful to see the actual list of employees in **2.7 Add Shift**.



**Figure 23 - Adding a new Employee to the database Form**

## 3.0 Viewing Data logs

For this assignment Log4php is used as a data logger and there are 3 main concepts in Apache log4php: loggers, appenders and layouts. These three types of components work together to enable developers to log messages according to message type and level, and to control at runtime how these messages are formatted and where they are reported.

- **Loggers** : A logger is a component which will take your logging request and log it. Each class in a project can have an individual logger, or they can all use a common logger. Loggers are named entities; it is common to name them after the class which will use it for logging.
- **Appenders** : Logging requests can be sent to multiple destinations and such destinations are called appenders. Appenders exist for console, files, syslog, database, sockets and other output destinations. One or more appenders can be attached to a logger. Each enabled logging request for a given logger will be forwarded to all the appenders in that logger.



**logging.log**

- Layouts : Layouts are components responsible for transforming a logging event into a string. Most appender classes require a layout class to convert the event to a string so that it can be logged.
- Levels : A level describes the severity of a logging message. There are six levels, show here in descending order of severity.

FATAL : Very severe error events that will presumably lead the application to abort.

ERROR : Error events that might still allow the application to continue running.

WARN : Potentially harmful situations which still allow the application to continue running.

INFO : Informational messages that highlight the progress of the application at coarse-grained level.

DEBUG : Fine-grained informational events that are most useful to debug an application.

TRACE : Finest-grained informational events.

However in our logging, only 2 levels are used which are INFO and ERROR.

INFO shows the successful/desired actions which user/owner did.

ERROR shows the unsuccessful/undesired actions which user/owner did.

There is a logging.log file inside acrs-team folder. This text file records the data over time so it allows us to see what user/owner did on the webpage. It keeps adding the loggings. (most recent ones are at the below)

To avoid confusion, it can be deleted because there will be always a logging text file after using webpage so that the text file will have only the most recent actions.

Configuration option is added as a file in our assignment and it is passed to log4php by calling the static method `Logger::(' ../config.xml')` configure before issuing any logging requests.

Configuration path has been added to the .php pages which we want to get the logging from

```
/* Adding logging config path */
include('../datalogging/Logger.php');
Logger::configure('../config.xml');
$logger = Logger::getLogger("main");
```

An example of the logging text file;

