

C868 – Software Capstone Project Summary

Proposal Project Name: _____ C868 - PIBOOKING _____

Student Name: _____ (David) Wayne Massey _____

Table of Contents

Contents

<i>Table of Contents.....</i>	<i>2</i>
<i>Task 2 Part C – C868 Software Development Capstone.....</i>	<i>4</i>
<i>Application Design and Testing.....</i>	<i>4</i>
<i>Design Document.....</i>	<i>4</i>
Design.....	4
Database Design.....	5
.....	6
UI Design.....	6
<i>Introduction.....</i>	<i>7</i>
Purpose.....	7
Default Credentials:.....	8
<i>User Guide.....</i>	<i>9</i>
Introduction.....	9
Installation and Using the Application.....	9
SQL Server.....	9
API Layer.....	11
User Interface.....	12
Login.....	14
Admin Functionality.....	15

PIBOOKING – SECTION C

Create a New Engineer.....15

Reports.....18

Task 2 Part C – C868 Software Development Capstone

Application Design and Testing

Design Document

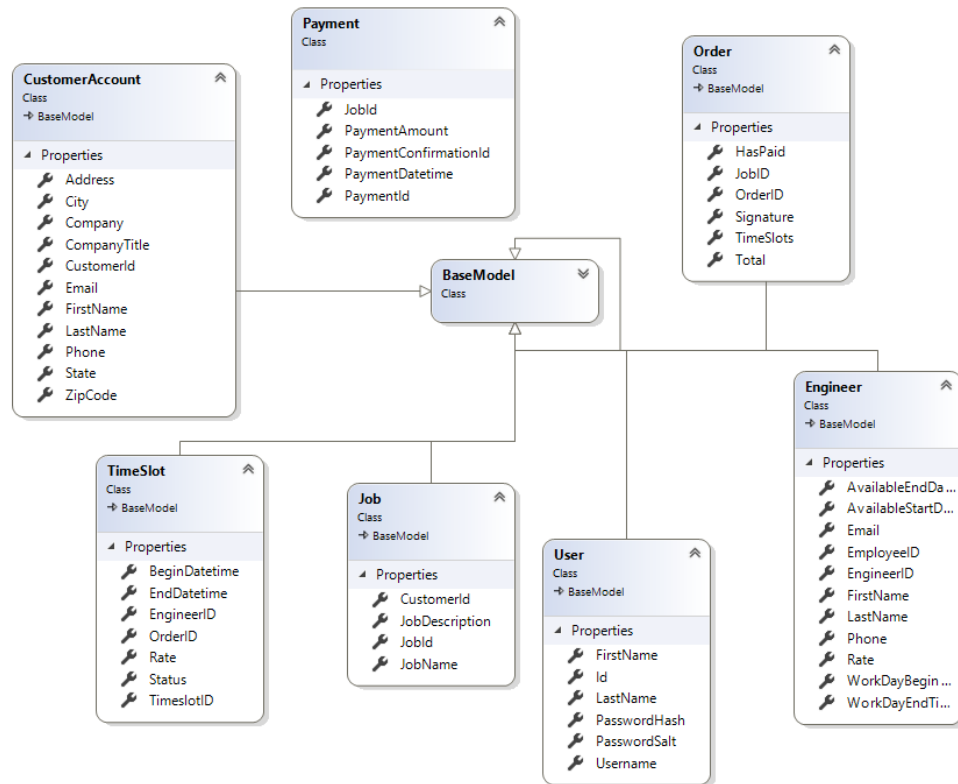
Design

The application consists of the following layers:

Controllers	Acts as a mediator between application and UI. Creates and consumes ViewModels to provide to the UI.
ViewModel	Representation of the data to be consumed or provided by the client UI
Services	Requests the domain models from the repository. Allows scoped business functionality related to said service.
Repository	Hydrates models from persistence store (currently SQL Server)
Models	POCO objects that provide representation of domain objects.

The following class diagram illustrates the classes found in the Models layer.

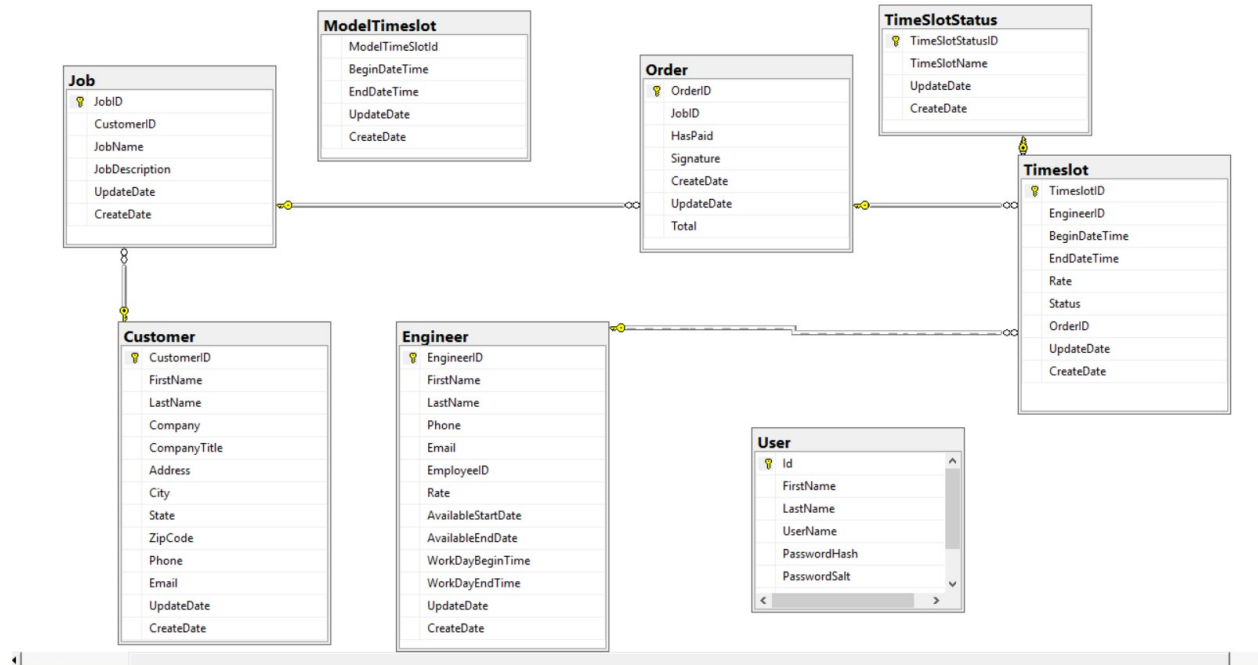
PIBOOKING – SECTION C



Database Design

The application is designed to use SQL Server as a persistence store. Currently, the application makes use of a relational database structure. Future versions of the application may use a NOSQL store for some of the data currently housed in SQL Server. The current database structure is outlined in the below diagram .

PIBOOKING – SECTION C

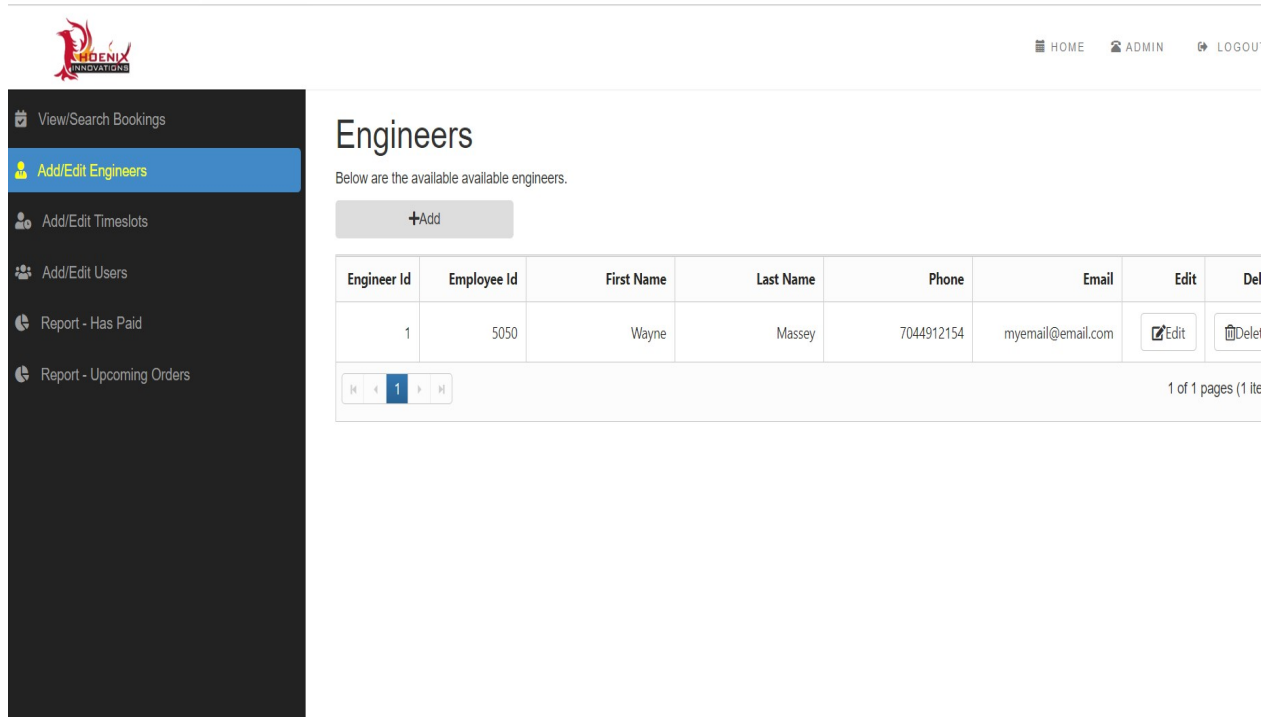


UI Design

The application is designed (for desktops) with a menu at the top and left-hand sections of the page. The top menu allows the user to access either the Home screen or Admin section.

On the home section, the left-hand menu provides a wizard type flow that allows the customer to create a booking (order). Customers are not able to proceed without providing any valid details being asked by the current screen.

On the admin section, the same left-hand menu allows staff to access various admin functionality.



The screenshot displays the PIBOOKING web application interface. On the left is a dark sidebar with a logo at the top and a list of navigation items: 'View/Search Bookings', 'Add/Edit Engineers' (highlighted in blue), 'Add/Edit Timeslots', 'Add/Edit Users', 'Report - Has Paid', and 'Report - Upcoming Orders'. The main content area has a top navigation bar with 'HOME', 'ADMIN', and 'LOGOUT' links. Below this, the 'Engineers' section is titled, followed by the text 'Below are the available available engineers.' and a '+Add' button. A table lists the available engineers with columns for Engineer Id, Employee Id, First Name, Last Name, Phone, Email, Edit, and Delete. The table contains one entry for Wayne Massey. At the bottom of the table is a pagination control showing '1' of 1 pages (1 item).

Engineer Id	Employee Id	First Name	Last Name	Phone	Email	Edit	Delete
1	5050	Wayne	Massey	7044912154	myemail@email.com	<input checked="" type="checkbox"/> Edit	<input type="checkbox"/> Delete

Introduction

Purpose

The application primarily serves to provide lead and order generation for a software development company. The most important data that can be obtained from the application is customer contact details. It is imperative that this is captured and stored in the persistence store.

PIBOOKING – SECTION C

The initial load for the User Interface is approximately 20MB. Depending on bandwidth and other factors, you may notice a slight delay in rendering. Future versions of this application will eliminate some dependencies which will allow the application to render more speedily.

Default Credentials:

The default admin account that can be used to access this application is:

Username: admin

Password: changethispassword

These credentials are also the default created on all new installations.

User Guide

Introduction

The following guide will illustrate how to setup the application, login, and utilize any admin functionality.

Installation and Using the Application

The application was built using .Net Core, Aurelia, and SQL Server. The below will direct on how to install in a new environment.

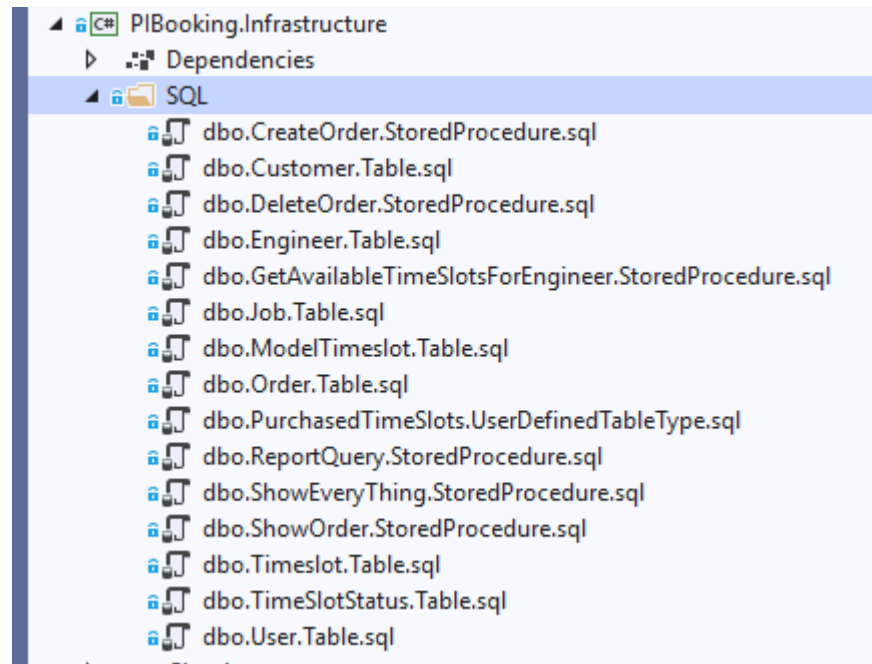
These instructions assume that a host or database server has been appropriated.

SQL Server

Note: There is a model database (PIBooking.Mdf) that runs on SQL Express LocalDB available within the PIBooking.Infrastructure project.

1. Create a new database named pibooking. The database assumes 'SQL_Latin1_General_CP1_CI_AS' collation running on SQL Server version 12.0.
2. Create an application user with appropriate permissions to the pibooking database. The application will need read/write and execute.
3. Within the application code directory, there are SQL scripts found in the PIBooking.Infrastructure project.

PIBOOKING – SECTION C



Execute each of these scripts in the new database. (See below picture) in the following order.

1. [dbo].[User]
2. [dbo].[TimeSlotStatus]
3. [dbo].[Customer]
4. [dbo].[Job]
5. [dbo].[Order]
6. [dbo].[Timeslot]
7. [dbo].[ModelTimeslot]
8. [dbo].[Engineer]
9. [dbo].[DeleteOrder]
10. [dbo].[GetAvailableTimeSlotsForEngineer]
11. [dbo].[ShowOrder]
12. [dbo].[ReportQuery]

13. [dbo].[PurchasedTimeSlots]

14. [dbo].[CreateOrder]

15. InitialDataPopulate

API Layer

1. Create a new web application on a host capable of executing .NET CORE 8.0 code. The applications should be configured to use https only.
2. Restore any NUGET dependencies
3. Build the solution
4. Publish the project PiBooking.API to the target host
5. Configure as appropriate for your environment the following configuration settings expected to be found in AppSettings. E.g. Local environments would use appsettings.json.



```
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Warning"
5     }
6   },
7   "AllowedHosts": "*",
8   "AppSettings": {
9     "PersistenceConnectionString": "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=E:\\Code\\C86
10     "Secret": "Why must this be a secret? WGU ROCKS!",
11     "EmailFromName": "Phoenix Innovations Inc.",
12     "EmailFrom": "",
13     "EmailSMTPUser": "",
14     "EmailSMTPPassword": "",
15     "EmailSMTPHost": "",
16     "EmailSMTPPort": "",
17     "EmailSMTPSubject": "Thank you for your order from Phoenix Innovations, Inc. "
18   }
19 }
20
21 }
```

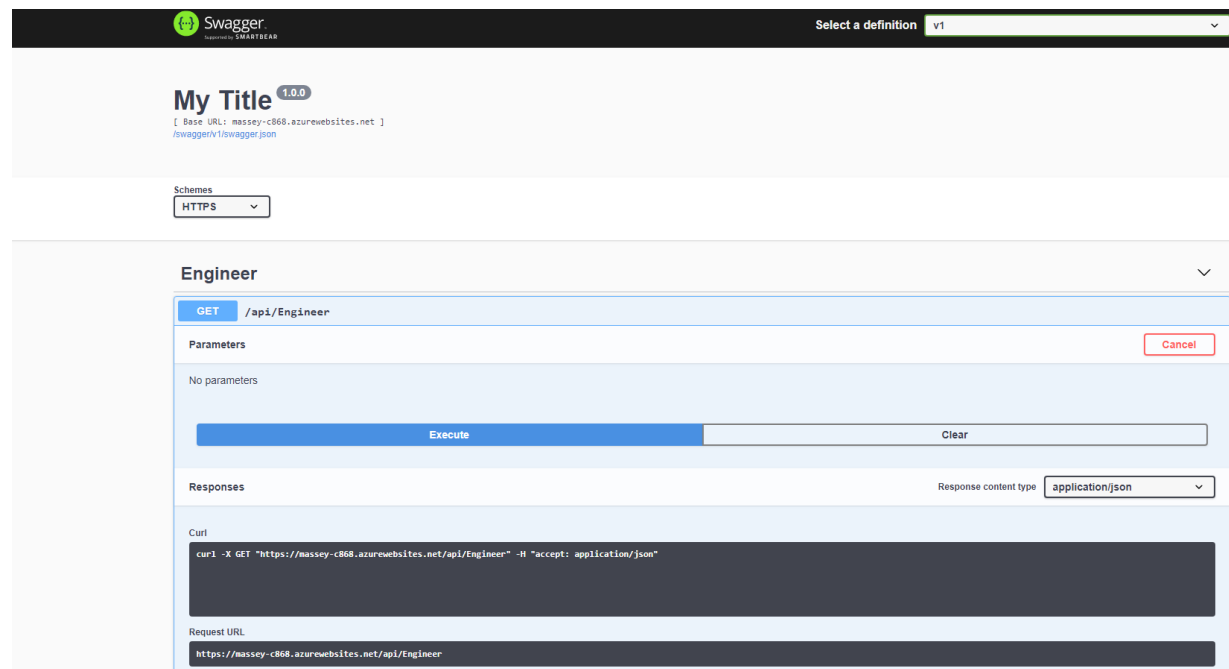
- PersistenceConnectionString - connection to the database
- Secret - a text value secret to use for salting and hashing
- EmailFromName - The name desired to use for **from** when sending invoices.
- EmailFrom - The email address desired to use for **from** when sending invoices.
- EmailSMTPSubject – Subject to use when sending invoices

PIBOOKING – SECTION C

- EmailSMTPUser – SMTP user
- EmailSMTPPassword – SMTP password
- EmailSMTPHost – SMTP host to use
- EmailSMTPPort – SMTP port to use

6. The API Layer should be at this point correctly configured. The API layer uses Swagger, so you should be able to get all ENGINEERS using the swagger URL.

e.g. https://massey-c868.azurewebsites.net/swagger/index.html#/Engineer/Engineer_Get



User Interface

1. The User Interface has been built using the Aurelia framework (<https://aurelia.io/>) , Webpack, and TypeScript. It is assumed that these are installed. The code for the user interface can be found in the project PiBooking.UI. While you can use Visual Studio to open this, it is best to use VS

CODE to work with the UI code. It is assumed that a recent version of node and npm (<https://www.npmjs.com/>) is installed.

2. Modify the application environmental variables found in *PiBooking.UI*
aurelia_project\environments to match the expected environment.

The following variable should be set to the base URL of the API layer. Do include the terminating '/' character.

```
remoteSessionUrl: " https://massey-c868.azurewebsites.net/api/ "
```

Also, change the similar named variable found in *PiBooking.UI\src\app*
environment\environment.ts

3. Open a powershell command window in the root of *PiBooking.UI*
4. Run ***npm i***
5. Run ***npm install --save-dev webpack***
6. Run ***au build*** with the appropriate environmental flag. E.g. ***au build --env:prod***
7. The aurelia transpiler will compile the application and the output will be found at
PiBooking.UI\wwwroot\dist.
8. Copy the output to the target host. It is expected the directory structure remains the same, that the application will be hosted in the root directory, and that the file located at *wwwroot\dist\index.html* is the entry or default path to the application.

Login

Most consumers will not need credentials to utilize the website. However, admin functionality is secured with credentials. To access any admin functionality, you may be challenged with a login screen. The following will illustrate how to successfully login.

1. If you are challenged for credentials:



Phoenix INNOVATIONS

HOME ADMIN

Username

admin

Password


Login

2. Enter in your credentials. The default credentials for a new installation is:

Username: admin

Password: changethispassword

3. Click Login. If the credentials are incorrect, you may be presented with an error message.



Phoenix INNOVATIONS

HOME ADMIN

Username or password is incorrect

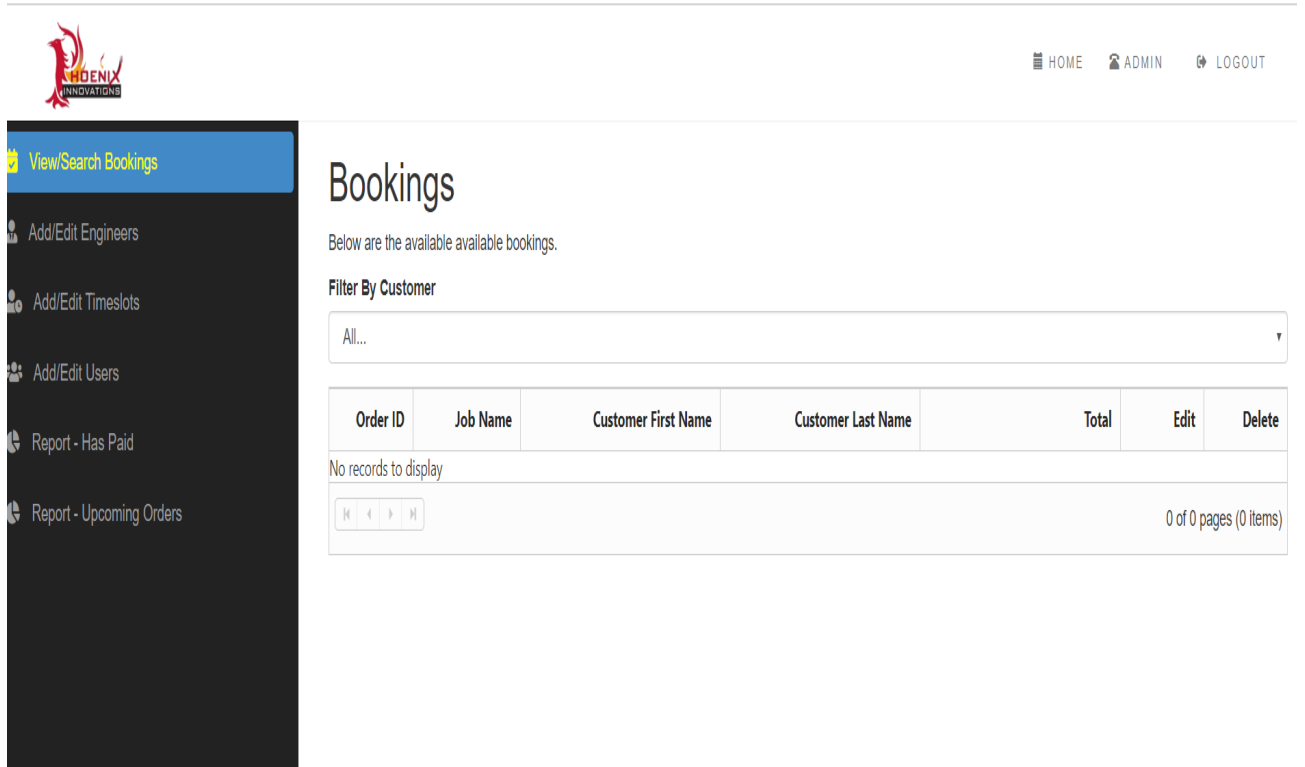
Username

admin2

Password

Login

4. If the login is successful, you will be redirected to the admin section of the website.



The screenshot displays the PIBOOKING Admin interface. On the left is a dark sidebar with a logo at the top and a menu containing: 'View/Search Bookings' (highlighted in blue), 'Add/Edit Engineers', 'Add/Edit Timeslots', 'Add/Edit Users', 'Report - Has Paid', and 'Report - Upcoming Orders'. The main content area is titled 'Bookings' and includes the text 'Below are the available available bookings.' and a 'Filter By Customer' dropdown menu currently set to 'All...'. Below the filter is a table with columns: 'Order ID', 'Job Name', 'Customer First Name', 'Customer Last Name', 'Total', 'Edit', and 'Delete'. The table is currently empty, showing 'No records to display'. At the bottom of the table area, there are pagination controls and the text '0 of 0 pages (0 items)'.

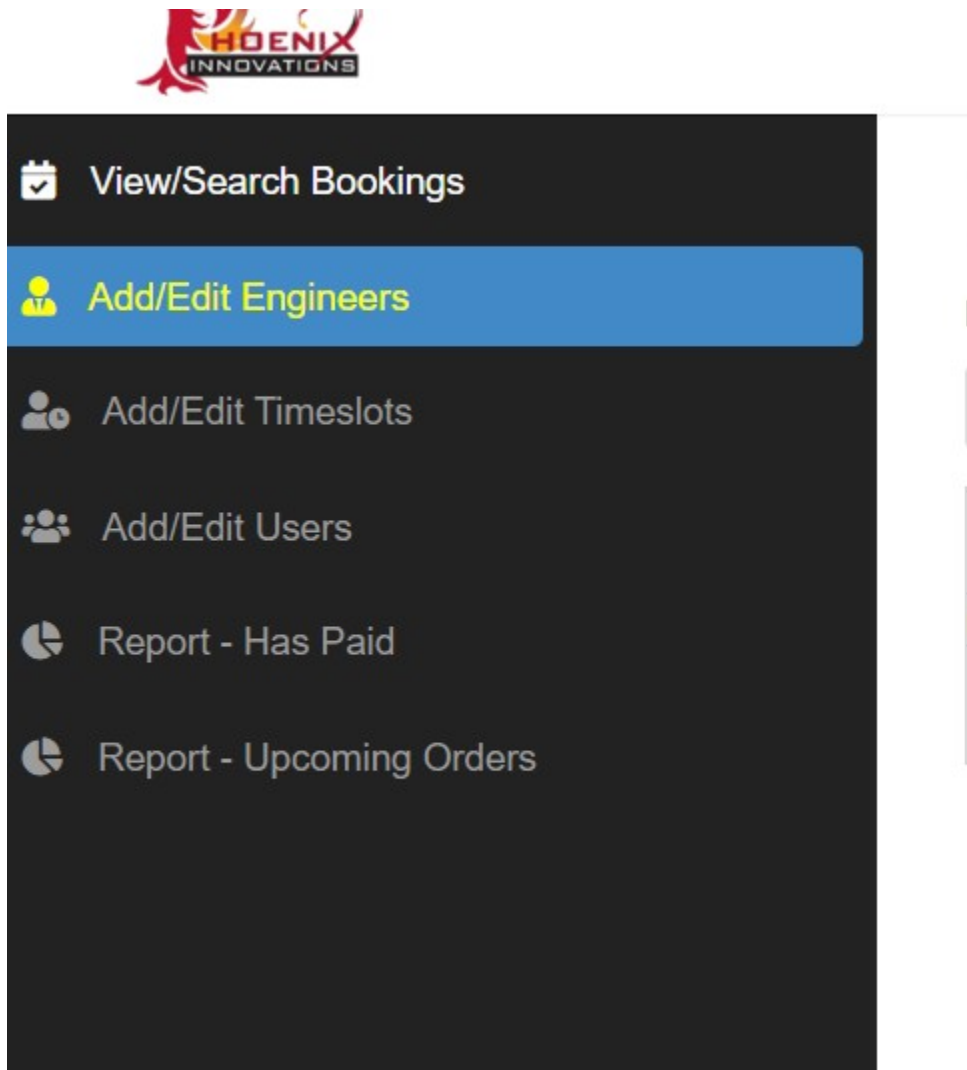
Admin Functionality

Create a New Engineer

On any new install, it is recommended to first create any engineers that will be available for customers. Customers will not be able to use the application if there are no engineers created.

You can create a new engineer as follows:

1. Click on Add/Edit Engineers in the admin section of the website.



2. Click Add

Engineers

Below are the available available engineers.

A grey rectangular button with rounded corners, containing a plus sign icon and the text 'Add'.

--	--	--

3. Enter the engineer details. Note Engineer ID will automatically populate on save.

PIBOOKING – SECTION C

Employee ID	5050
Phone	7044912154
Email address	myemail@email.com
Rate	50
Engineer's Start Date	9/1/2019 12:00 AM
Engineers Termination Date	11/30/2019 12:00 AM
Daily Workday Start Time	04:00
Daily Workday End Time	22:00

4. Click Save

 Save

5. The engineer will be saved, and Engineer ID will be populated. The application will note that the “Engineer has been saved.”.

Please enter the engineer information below.

Engineer has been saved.

Engineer ID

1

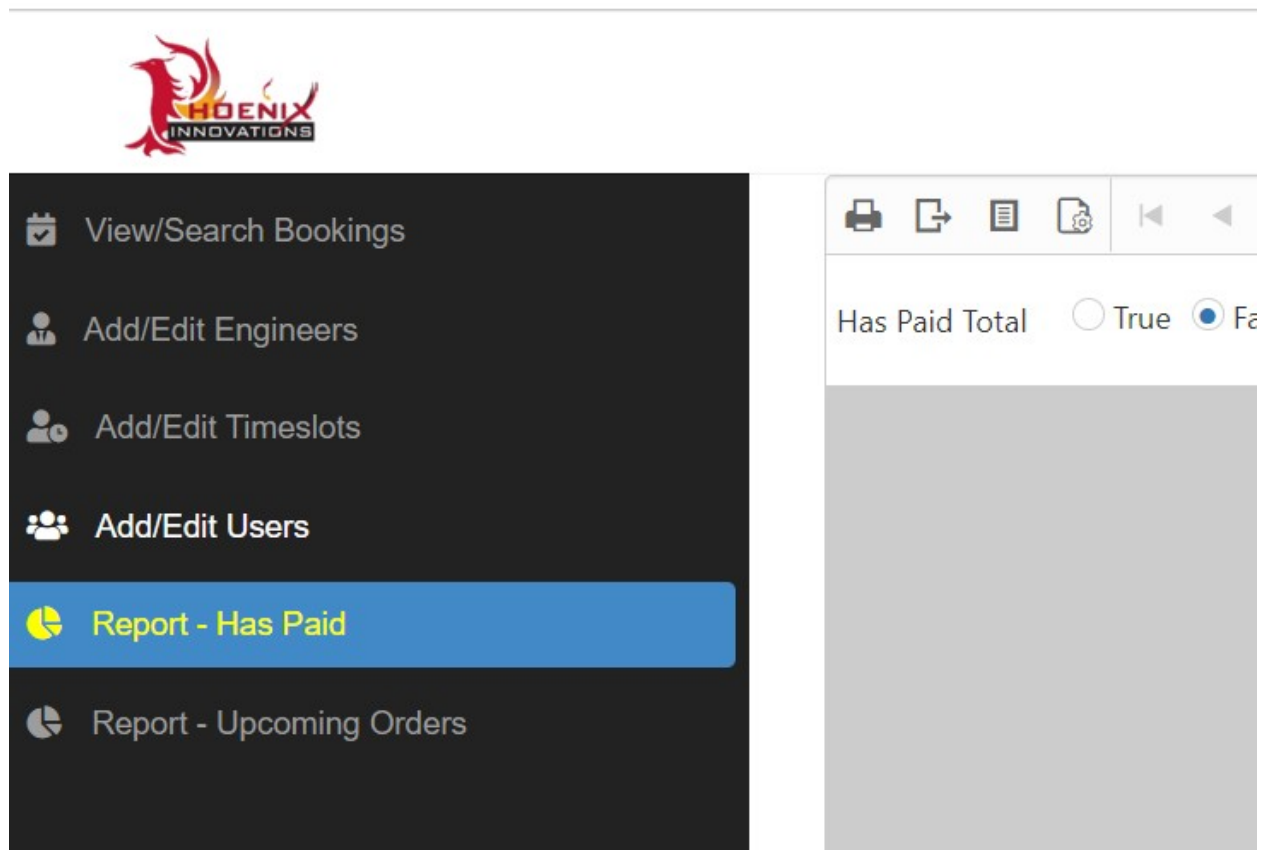
First Name

Wayne

Reports

The application consists of several reports designed to allow administrators to determine any orders (aka bookings) that have been paid or currently remain unpaid. In addition, reports are available to allow staff to determine which orders (bookings) are upcoming in a given date range.

Reports can be found in the admin section of the website.



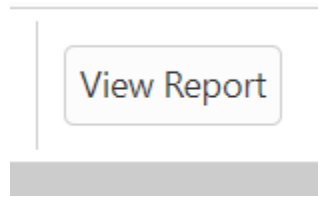
The following will illustrate how to print the “Has Paid” report.

1. Click “Report- Has Paid” in the admin menu.
2. Click on the appropriate value for the report parameter.

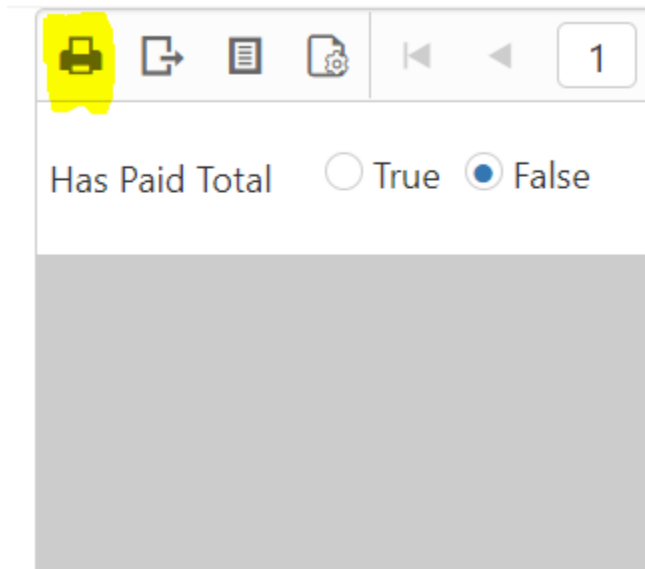
Parameter Value	Effect
True	Show those orders that have been paid
False	Show those orders that remain unpaid

3. Click the “View Report” button in the top-right. This button will need to be clicked anytime the parameters change.

PIBOOKING – SECTION C



4. The report will render.
5. Click the “Print report” icon directly above the label “Has Paid Total” in the top-left are of the report menu.



6. The browser will show the Print window. This window will be browser specific.
Click Print.

