The Bridge Inn website

Richmond School

Web application and database interface

George Smith (Student)

2019

Table of Contents

[Analysis 3](#_Toc14081031)

[Background to/identification of the Problem 3](#_Toc14081032)

[Research Methods 3](#_Toc14081033)

[Identification of prospective user(s)? 3](#_Toc14081034)

[User Needs 4](#_Toc14081035)

[Acceptable Limitations 4](#_Toc14081036)

[Description of the current system 4](#_Toc14081037)

[Objectives 5](#_Toc14081038)

[Data Dictionary 5](#_Toc14081039)

[Data Flow Diagrams 5](#_Toc14081040)

[Entity Relationship Diagrams (for databases) 5](#_Toc14081041)

[Object Orientation planning (for Object Orientation) 5](#_Toc14081042)

[Design 6](#_Toc14081043)

[Modular Design 6](#_Toc14081044)

[Definition of data requirements 6](#_Toc14081045)

[File organisation and processing (if appropriate) 6](#_Toc14081046)

[Database design including normalised relations 6](#_Toc14081047)

[Sample of planned SQL queries 6](#_Toc14081048)

[Identification of suitable algorithms for data transformation, pseudocode of these algorithms 6](#_Toc14081049)

[UML Diagrams 6](#_Toc14081050)

[Class definitions(diagrams) 6](#_Toc14081051)

[Use case Diagram 6](#_Toc14081052)

[Sequence Diagram 6](#_Toc14081053)

[(https://www.websequencediagrams.com/ ) 6](#_Toc14081054)

[User interface design (HCI) rationale 6](#_Toc14081055)

[Identification of storage media 6](#_Toc14081056)

[Technical solution 7](#_Toc14081057)

[Overview guide 7](#_Toc14081058)

[the names of entities such as executables 7](#_Toc14081059)

[data filenames/urls, database names, pathnames so that a third party can, if they so desire, run the solution/investigation 7](#_Toc14081060)

[Detailed Algorithm Design 7](#_Toc14081061)

[Testing 8](#_Toc14081062)

[Introduction and overview of testing 8](#_Toc14081063)

[(Black Box and White Box testing) 8](#_Toc14081064)

[Test Table 8](#_Toc14081065)

[(Test Number, Descrption of test, Test Data Type, Expected Result, Pass/Fail, Crossreference) 8](#_Toc14081066)

[Each screenshot should have a short annotation describing what is shown if not obvious 8](#_Toc14081067)

[Evaluation 9](#_Toc14081068)

[Did you meet your objectives? 9](#_Toc14081069)

[Feedback on requirements 9](#_Toc14081070)

[Analyse feedback 9](#_Toc14081071)

[Extensions 9](#_Toc14081072)

# Analysis

## Background to/identification of the Problem

The Bridge Inn

Grinton, Richmond, North Yorkshire DL11 6HH

A former Coaching Inn dating back to the 15th century, The Bridge Inn sits in the picturesque village of Grinton opposite the 12th century Church of St. Andrew, which is also known as the 'Cathedral of the Dale'. The boss and landlord Andrew Atkin manages the pub and leads a team of chefs, waiters, pot washers, and a few other odd jobs.

Unfortunately, the rustic appeal and old fashioned means the Bridge’s online presence is lacking. It does currently have a website but it is years out of date and very simplistic with no real purpose. I plan to create a website that is far more attractive and will bring in customers that would not normally come to the pub due to its rural location. An online booking system for tables and rooms would also improve accessibility and replace the old pen and paper system that is often misplaced and cannot be used remotely. As well as booking, a scheduling system that could organise the many employees would be very helpful and also replace its outdated paper predecessor.

## Research Methods

I have worked at the Bridge for well over a year and learnt these problems by extension. I will also create and use a survey with a range of workers (e.g. Chefs and waiters) about problems they encounter that I may not have seen.

* Current systems. Paper based booking and paper rota sheet.
* Physical systems can be easily lost or misplaced. Paper cannot be accessed remotely or by more than one person at a time.
* The booking pen and paper is only written by the barman or the boss but is read by all members working in the kitchen
* A scheduling system would be edited by the boss or head chefs but everyone member could read their schedule.
* The new system will allow online booking, automatic employee scheduling on a weekly basis but also for larger events. On the front it is also a website that customers can view to check the pub out.
* It will be a web application

## Identification of prospective user(s)?

Anyone can access the website in some form but some may access more depending on their authorisation. IT skills among users will be limited so it must be user friendly

* Customers can only access the website front and online booking
* Higher level members like the boss can access all parts in order to control the business. This includes functions such as manual edits to schedules or removing a booking.
* Lower level workers such as a pot washer or waiter can only view the rota and bookings
* The website will be primarily access by mobile phones but will be compatible with as many devices as possible.

## User Needs

* Advertisement for the pub, links to social media
* Menu viewing
* Online booking
* Login system for employees of the pub
* Automatic scheduling
* Databases
  + Bookings
  + Tables to book
  + Rooms to book
  + Employees
  + Menu

## Acceptable Limitations

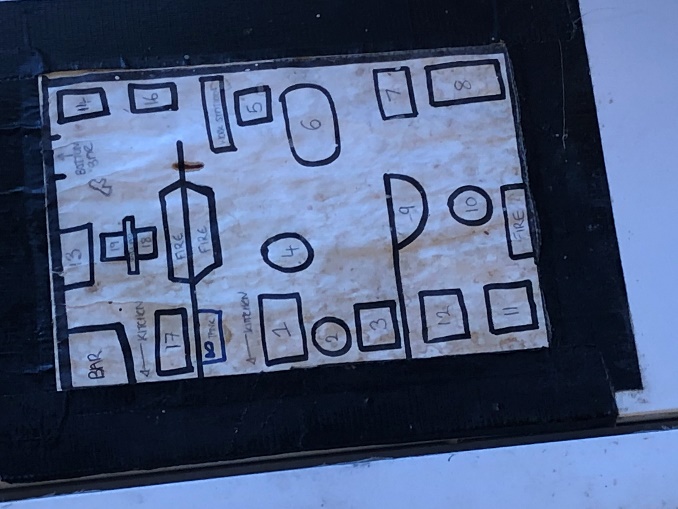
The most important request for the website is advertisement. So at the very least the website needs to show off the pub with social media links, nice pictures, and a smooth interface. Systems like booking have a tried and true fall back if the system fails.

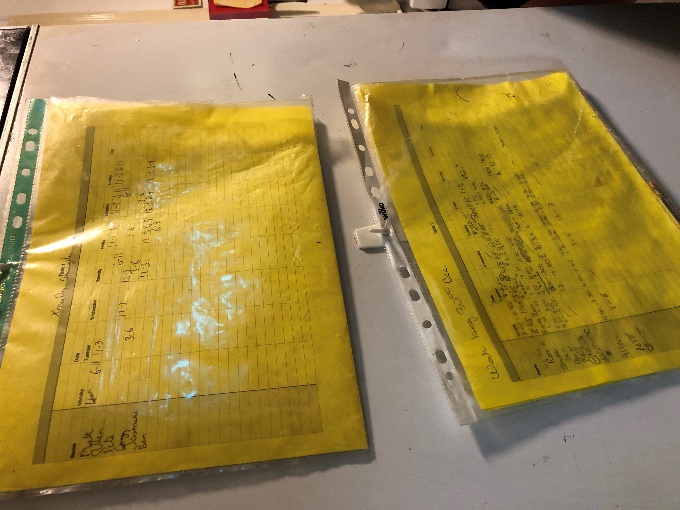
## Description of the current system

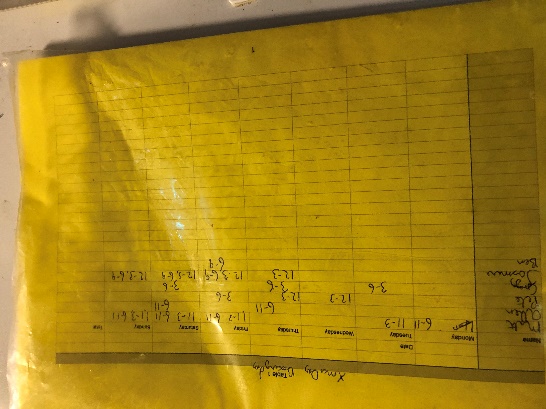
[Current website](http://www.bridgeinn-grinton.co.uk/)



* No social media Bridge specific links
* Simple descriptions
* Simple share on social media links
* Years out of date
* No online booking
* No menu
* Static
* Not very appealing
* Current systems. Paper based booking and paper rota sheet. Accessed behind the bar or booked by phone or in person
* Physical systems can be easily lost or misplaced. Paper cannot be accessed remotely or by more than one person at a time.
* The booking pen and paper is only written by the barman or the boss but is read by all members working in the kitchen
* A scheduling system would be edited by the boss or head chefs but everyone member could read their schedule. Currently it is messy and unofficial.
* The new system will allow online booking, automatic employee scheduling on a weekly basis but also for larger events. On the front it is also a website that customers can view to check the pub out.
* It will be a web application

 The current system for tables and organisation

The rota and future schedule

 Closer look at how employees’ shifts are recorded

## Objectives

1. The website’s aesthetic will be designed according to the boss’ input
2. The home page will include the booking and login systems
3. The booking system will allow customers to view available rooms and tables
4. Customers will be able to book an available room and/or a table
5. The contact us page will contain social media links, phone number, and email
6. The menu page will show food available with respective prices
7. Customers will be able to search by term or order
8. Staff will be able to login to the admin pages to access extra features
9. Some staff will have priority access
10. Priority access will dictate the features staff can access
11. All staff see the rota how many people are booked in
12. Priority staff can add bookings manually that haven’t been made online
13. Shifts will be automatically scheduled and displayed in the rota

## Data Dictionary

Employees:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Name | To identify the employee | String | 70 | John Doe | Not blank |
| Job title | To identify their job at the pub and show what they can and can’t access | String | 70 | Waiter | Not blank |
| Password | To allow a secure login system | String | 70 | \_Pa$$word123 | Not blank, strong password |
| Email | Used to contact with rota | String | 70 | Johndoe123@hotmail.com | Email format |

Shifts:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Day of the week | Identify time of shift | String | 9 | Friday | Not blank, valid day of the week |
| Time of day | Identify time of shift | Date/Time | 4 | 18:00 | Date/Time format |
| Date | Identify time of shift | Date/Time | 6 | DD/MM/YY | Date/Time format |
| People booked in | Used to see how busy the shift will be | Integer | 100 | 30 | Integer |

Menu:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Name of food/drink | Identify what it is | String | 70 | Sunday lunch | Not blank |
| Price | How much does it cost? | Float | 5 | £09.95 | Price format |
| Type of course | Determine whether its starter, main, desert, or drink | String | 7 | Starter | Not blank,  Either starter, main, dessert, drink |
| Dietary | Whether food meets dietary requirements | String | 70 | Vegetarian | Not blank |

Table bookings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Surname | Identify customer | String | 70 | Smith | Not blank |
| Number of people | Identify the amount of people that will need to be served | Integer | 80 | 5 | Not blank |
| Time | Identify time of booking | Date/Time | 4 | 18:00 | Not blank |
| Date | Identify time of booking | Date/Time | 6 | DD/MM/YY | Not blank |

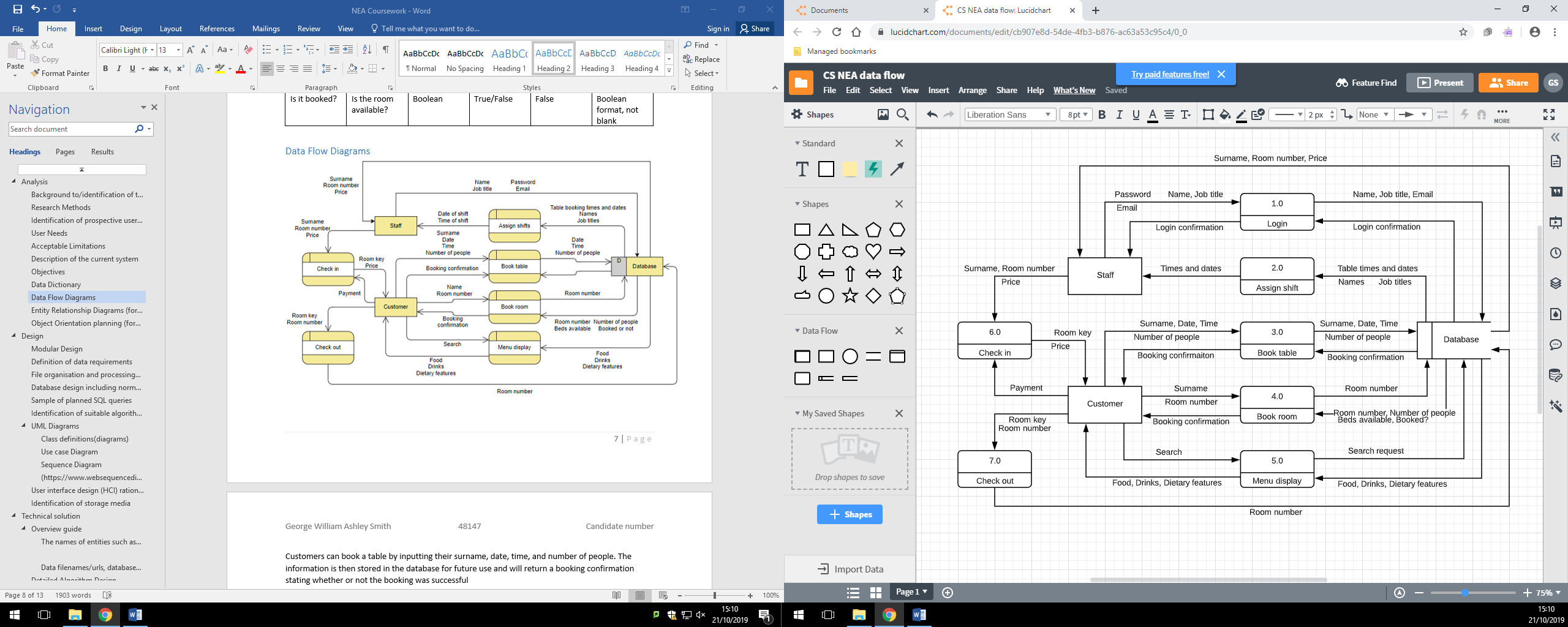
Room bookings:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Surname | Identify the customer | String | 70 | Smith | Not blank |
| Room number | Identify the room | Integer | 1 | 2 | Not blank |
| Numbers of people | Identify number of customers | Integer | 5 | 2 | Not blank |
| Date | Identify time of booking | Date/Time | 6 | DD/MM/YY | Date/time format |

Rooms:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Room number | Identify the room | Integer | 1 | 2 | Not blank |
| Beds available | What kind of beds are available | String | 70 | Two single beds | Not blank |
| Number of people that can fit | Number value for size of room | Integer | 5 | 2 | Not blank |
| Is it booked? | Is the room available? | Boolean | True/False | False | Boolean format, not blank |

## Data Flow Diagrams



Customers can book a table by inputting their surname, date, time, and number of people. The information is then stored in the database for future use and will return a booking confirmation stating whether or not the booking was successful

Customers can book a room by specifying their name and the room number according to available rooms. The database uses the room number to return the information about this room such as number of beds and if the room is not already booked then a booking confirmation will be sent to the customer.

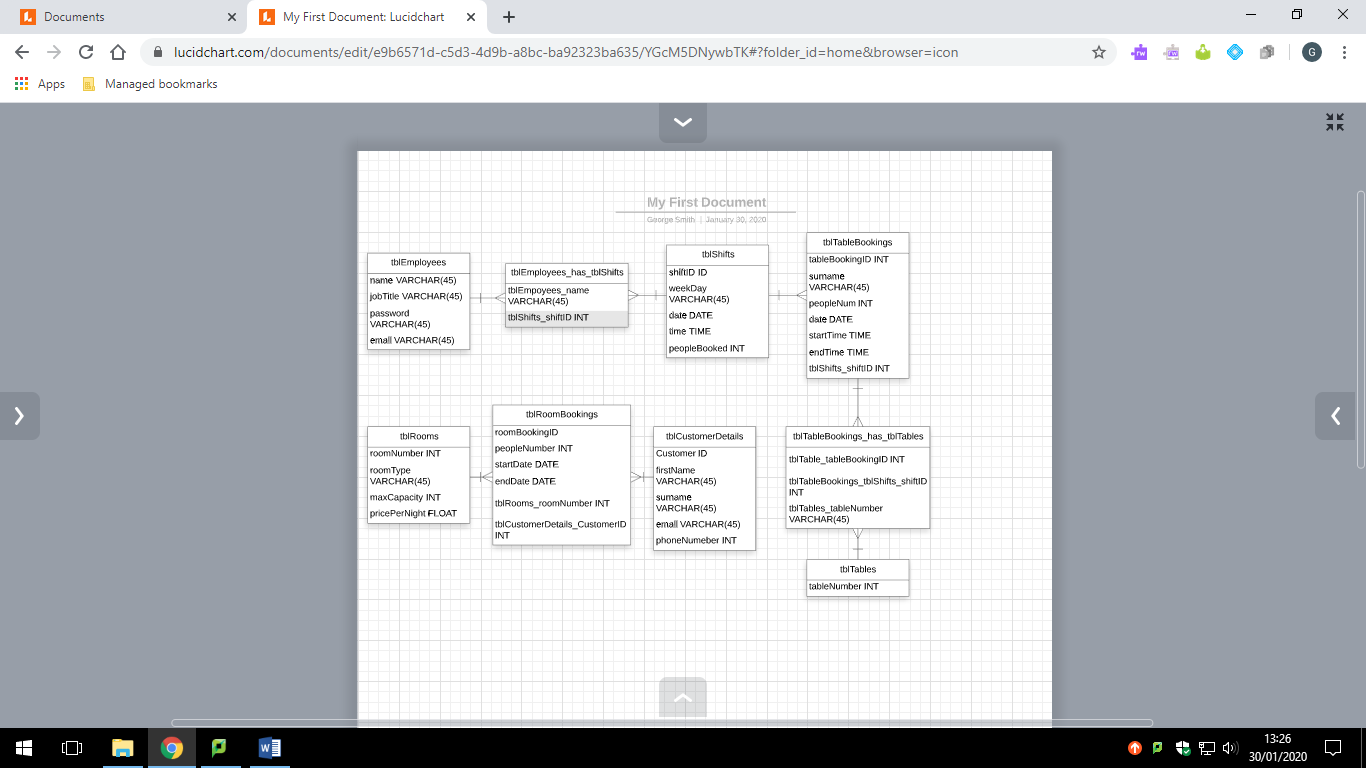
Customers can view the menu and input things to search by. The database then returns the information according to the search for displaying

Customers who have booked a room have to check in and check out. Checking in requires the database to send reference information to the staff to confirm the identity and booking with the customer as well as then hand over the room key. The costumer will respond with payment that once the database has returned price. Checking out will include customers returning the room key and the room number must be returned to the database in order to reset the room to available

Staff can register and login to the system with their account details which are then stored in the database.

Using all booking data in the database the system will assign shifts to the staff automatically. Staff then receive the date and time of their shift.

## Entity Relationship Diagrams (for databases)



# Design

## Modular Design

Menu page

Main page

Contact us page

Web application

Search or order by term

Login

Social media access

Booking

Table booking

Room booking

Access rota

Edit rota

Edit bookings

* Menu page
  + Search or order by term
* Main page
  + Booking
    - Table booking
      * Book a table
    - Room booking
      * Book a room
  + Login
    - Access rota
    - Edit rota (priority staff)
    - Edit booking
* Contact us
  + Access social media

## Definition of data requirements

Employees:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Name | To identify the employee | String | 70 | John Doe | Not blank |
| Job title | To identify their job at the pub and show what they can and can’t access | String | 70 | Waiter | Not blank |
| Password | To allow a secure login system | String | 100 | $2y$10$/Gd2flYQvTGX3SoZs  84UoOCYa4NnIpKEDwbXde8  J2kGeMv/bH3UJW | Not blank, strong password |
| Email | Used to contact with rota | String | 70 | Johndoe123@hotmail.com | Email format |

Shifts:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Day of the week | Identify time of shift | String | 9 | Friday | Not blank, valid day of the week |
| Time of day | Identify time of shift | Date/Time | 4 | 18:00 | Date/Time format |
| Date | Identify time of shift | Date/Time | 6 | DD/MM/YY | Date/Time format |
| People booked in | Used to see how busy the shift will be | Integer | 100 | 30 | Integer |

Menu:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Name of food/drink | Identify what it is | String | 45 | Rump steak | Not blank |
| Description | Brief description of the meal for menu display | String | 255 | Local beef steak cooked to your liking and served with grilled tomato, onion rings, mushroom, rocket salad and chips |  |
| Price | How much does it cost? | Float | Float | £16.00 | Price format |
| Type of course | Determine whether its starter, main, desert, or drink | String | 45 | Main | Not blank,  Either starter, main, dessert, drink |
| Dietary | Whether meal is vegetarian | Boolean | 1 | True | Not blank |

Table bookings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Surname | Identify customer | String | 70 | Smith | Not blank |
| Number of people | Identify the amount of people that will need to be served | Integer | 80 | 5 | Not blank |
| Start time | Identify start of booking | Date/Time | 4 | 18:00 | Not blank |
| End time | Identify end of booking | Date/Time | 4 | 18:00 | Not blank |
| Date | Identify time of booking | Date/Time | 6 | DD/MM/YY | Not blank |

Room bookings:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Surname | Identify the customer | String | 70 | Smith | Not blank |
| Room number | Identify the room | Integer | 1 | 2 | Not blank |
| Numbers of people | Identify number of customers | Integer | 5 | 2 | Not blank |
| Start date | Identify start of booking | Date/Time | 6 | DD/MM/YY | Date/time format |
| End date | Identify end of booking | Date/Time | 6 | DD/MM/YY | Date/time format |

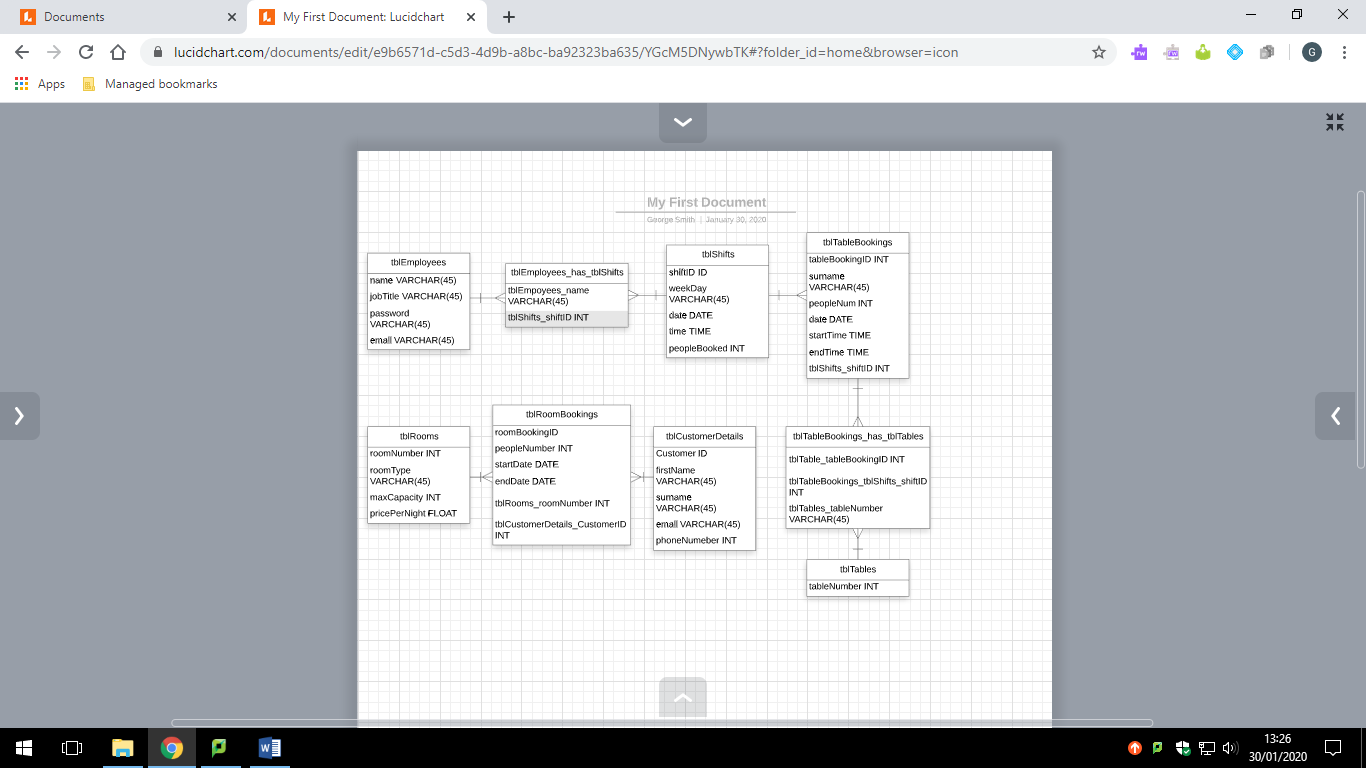
Rooms:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Room number | Identify the room | Integer | 1 | 2 | Not blank |
| Room type | What kind of beds are available | String | 70 | Twin | Not blank |
| Number of people that can fit | Number value for size of room | Integer | 5 | 2 | Not blank |
| Price per night | Price | Float |  | 80.00 | Not blank |

Tables:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field name | Field purpose | Field type | Field size | Example data | Validation |
| Table number | Identify the table | Integer | 2 | 14 | Not blank |

## Database design including normalised relations



## Sample of planned SQL queries

“SELECT itemName, description, price, vegetarian FROM tblMenu;”

* Retrieves all columns from the tblMenu table for a menu display

“INSERT INTO tblEmployees VALUES (:name, :jobTitle, :password, :email);”

* Inserts a record into tblEmployees in order to register an employee

“SELECT name, password FROM tblEmployees WHERE name = :name;”

* Retrieves necessary information from tblEmployees in order to check and accept a login

“SELECT name, jobTitle FROM tblEmployees WHERE name = :name;”

* Retrieves necessary data from tblEmployees for accurate display of rota and information as lower level employees shouldn’t be able to edit shifts

"SELECT roomNumber FROM tblRooms WHERE startDate > :startDate AND startDate < :endDate OR endDate > :startDate AND endDate < :endDate AND roomNumber = :roomNumber"

* Finds any rooms that are booked with the preferred dates that the customer has inputted

CREATE TABLE IF NOT EXISTS `rscomput\_gsmith`.`tblTableBookings` (  
  `tableBookingID` INT NOT NULL,  
  `surname` VARCHAR(45) NOT NULL,  
  `peopleNum` INT NOT NULL,  
  `date` DATE NOT NULL,  
  `startTime` TIME(0) NOT NULL,  
  `endTime` TIME(0) NOT NULL,  
  `tblShifts\_shiftID` INT NOT NULL,  
  PRIMARY KEY (`tableBookingID`, `tblShifts\_shiftID`),  
  INDEX `fk\_tblTableBookings\_tblShifts1\_idx` (`tblShifts\_shiftID` ASC),  
  CONSTRAINT `fk\_tblTableBookings\_tblShifts1`  
    FOREIGN KEY (`tblShifts\_shiftID`)  
    REFERENCES `rscomput\_gsmith`.`tblShifts` (`shiftID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;

* Creates the tblTableBookings table which is connected with tblTable and tblShifts

## Identification of suitable algorithms for data transformation, pseudocode of these algorithms

Logging in and registering

* For registering the user inputs data into fields which the code then processes and adds to the database
* Validation should include a valid email address (with an @)
* The password is hashed upon creation and stored hashed
* For logging in the user must input data into a field which is then checked against retrieved items from the database
* This includes the hashed password so an unhashed password must be compared with it

Scheduling

* Each shift has different requirements
  + Weekdays need 2 chefs and a waiter
  + Friday and Saturday nights, and Sunday afternoons need 2 chefs, 2 waiters, and a pot wash
* The algorithm must process a table of employees, a table of shifts, and also how many bookings there are in order to automatically assign the correct amount of available employees

Displaying and search bars

* Specifically in the menu section
* There will be a search bar for different ways of sorting the menu
* Eg. Search by course such as starter

## UML Diagrams

### Class definitions(diagrams)

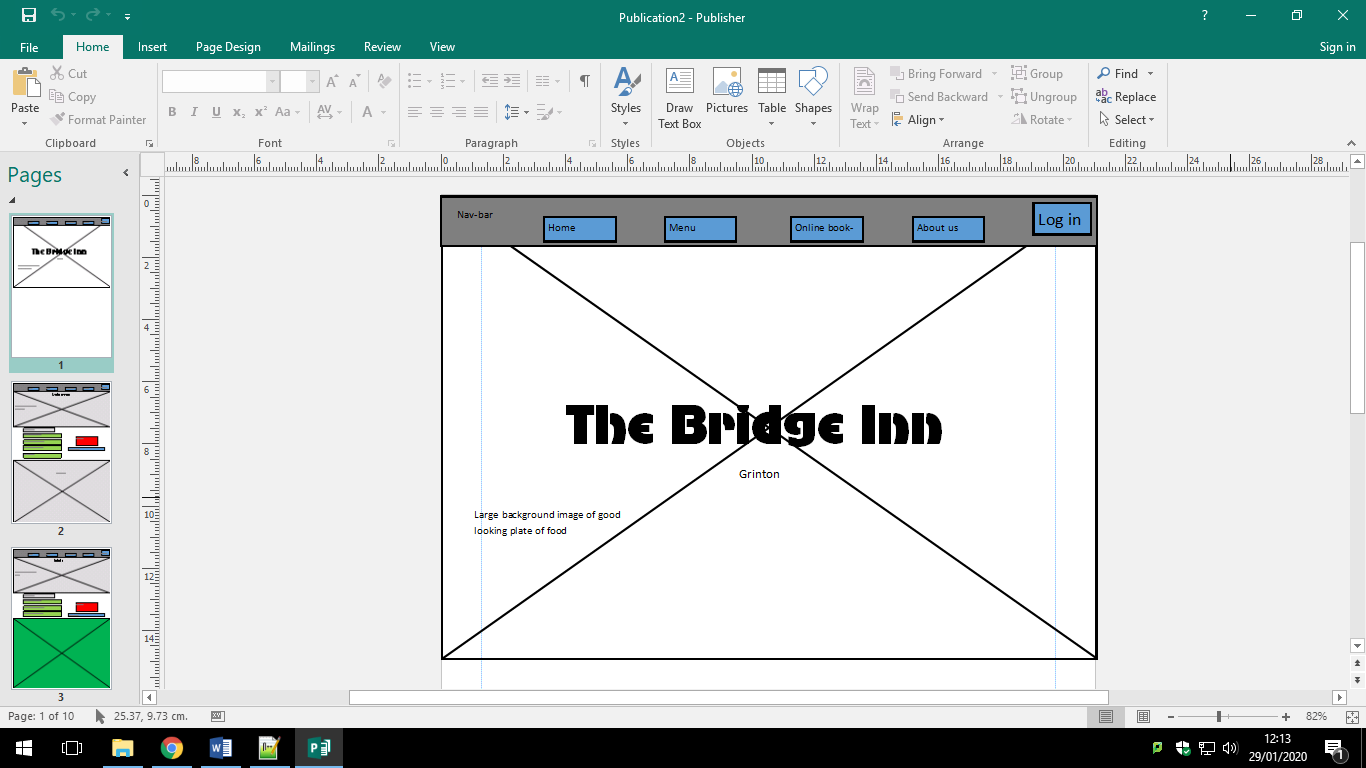
### Use case Diagram

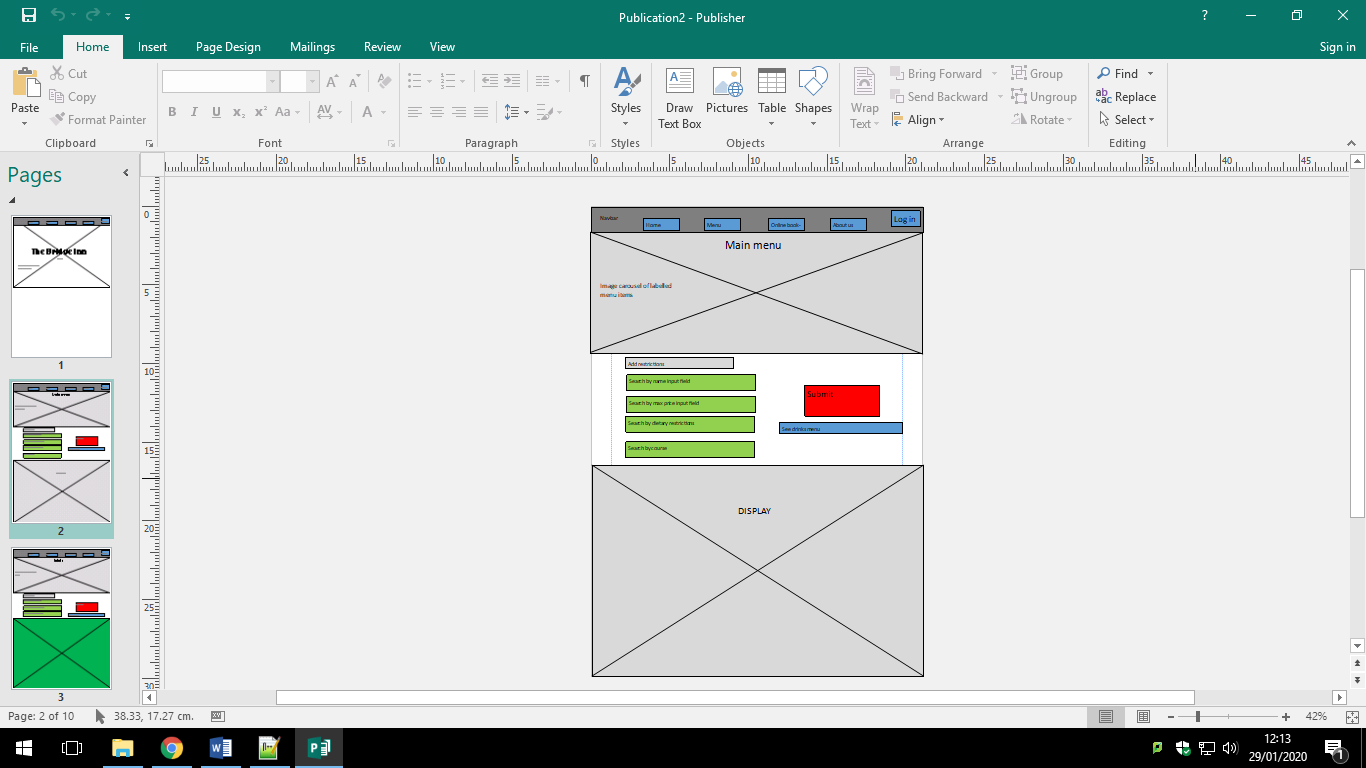
### Sequence Diagram

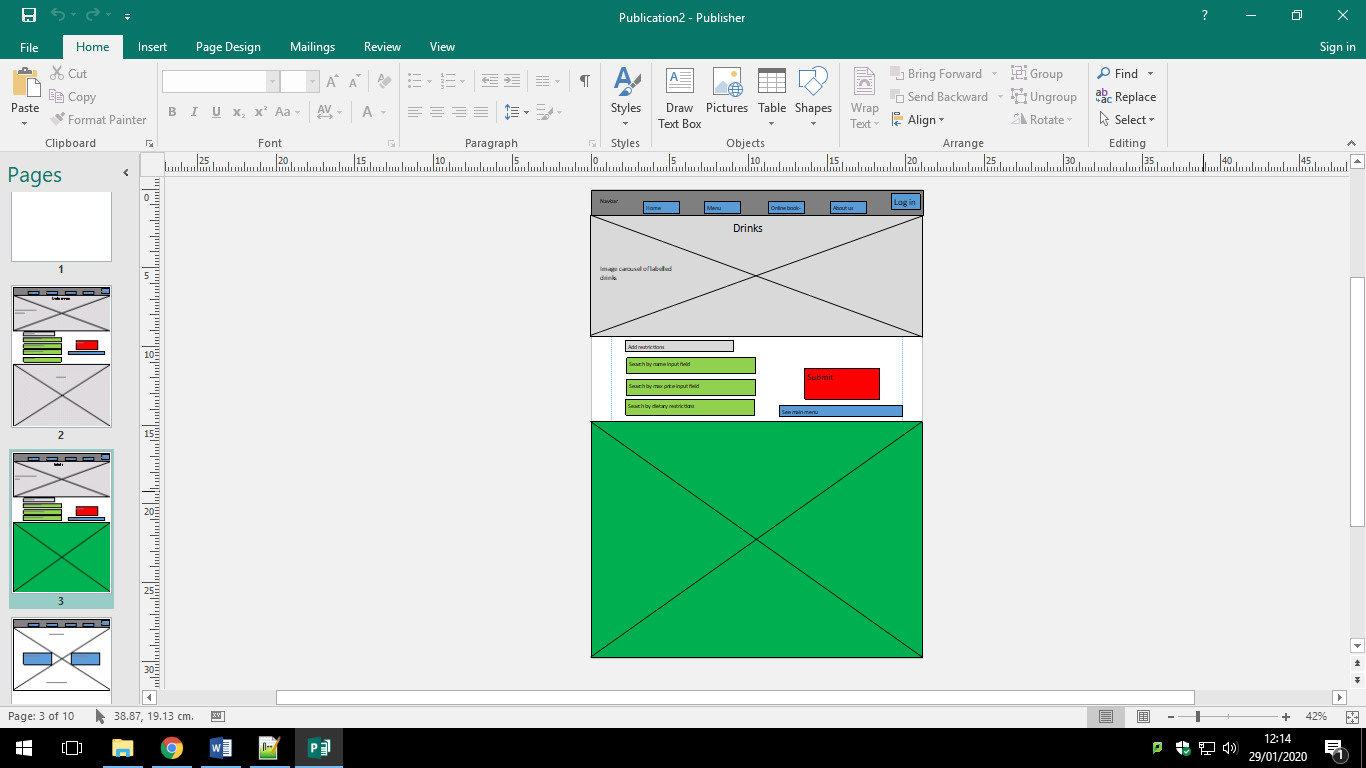
### (https://www.websequencediagrams.com/ )

## User interface design (HCI) rationale

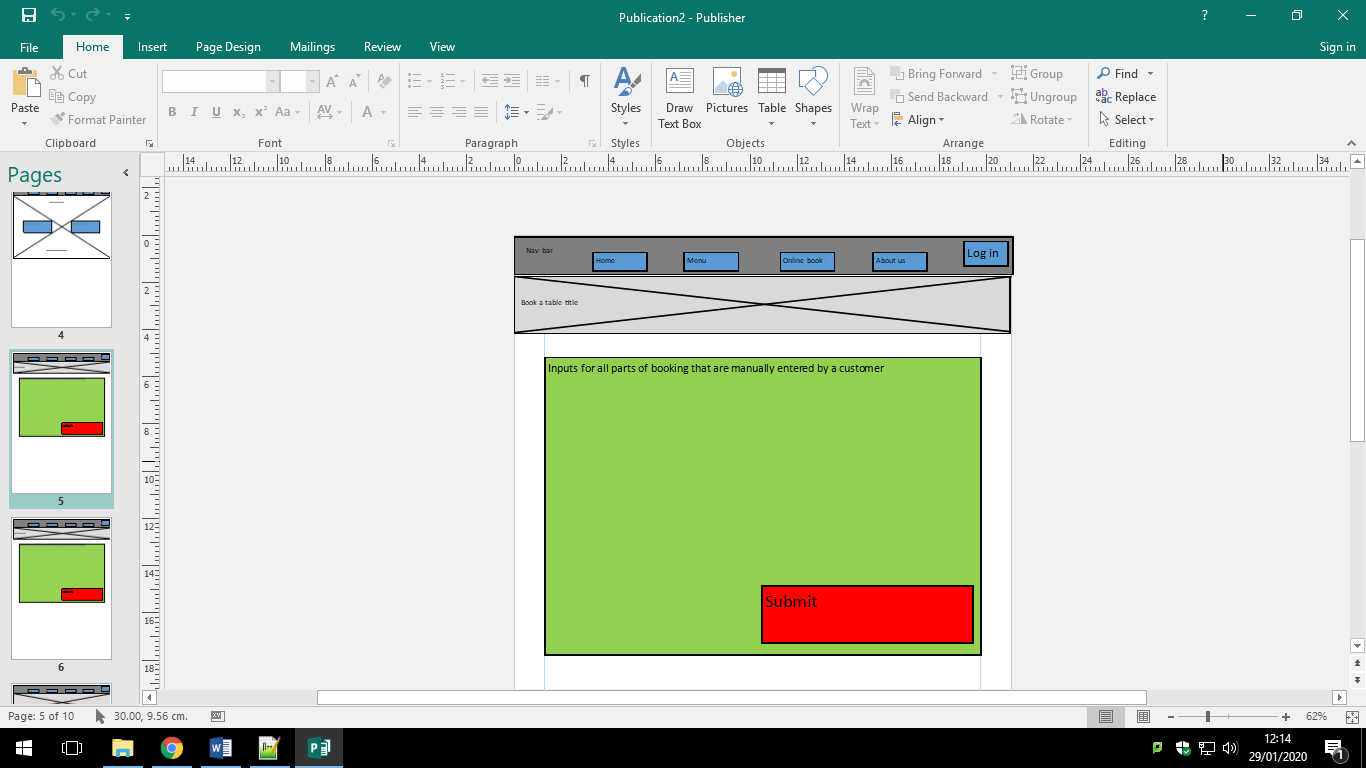
These images are all designs for each pages of the website

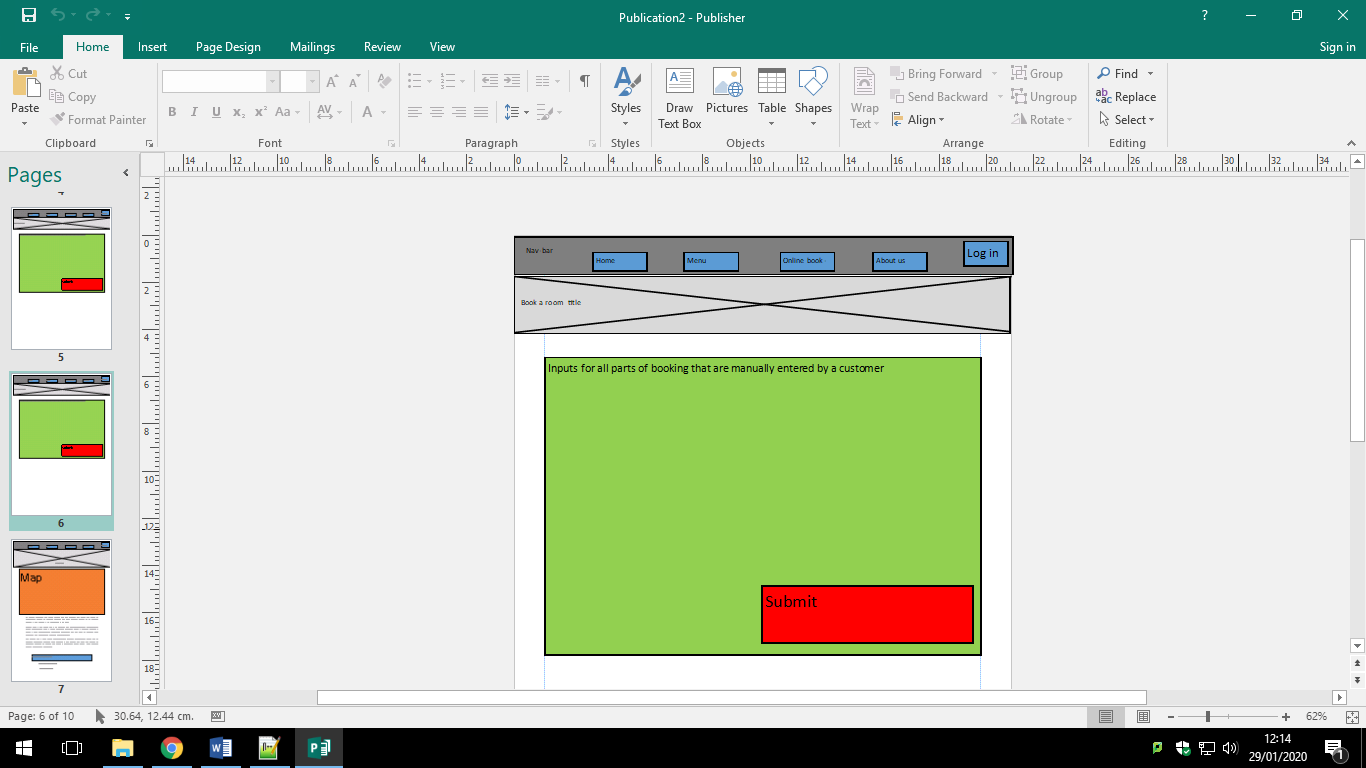
Index

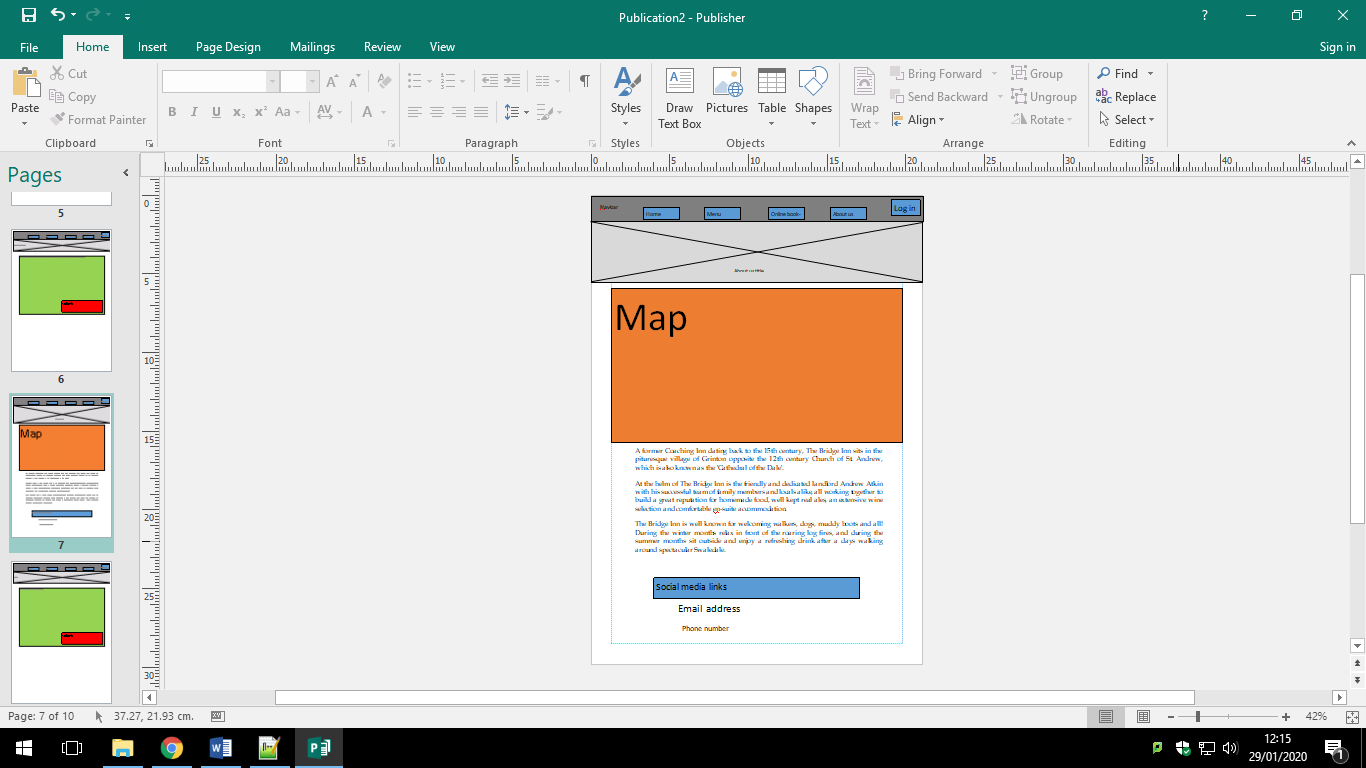
Food menu

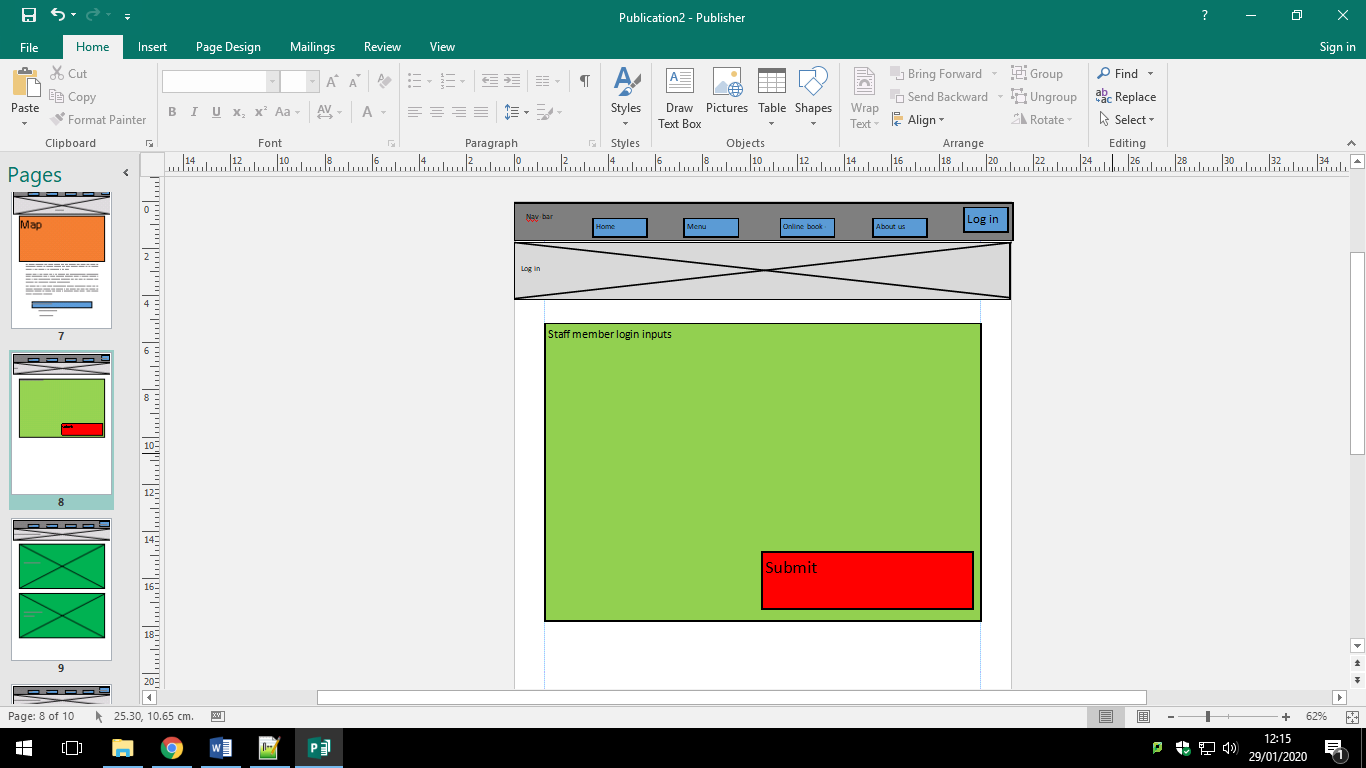
Drinks menu

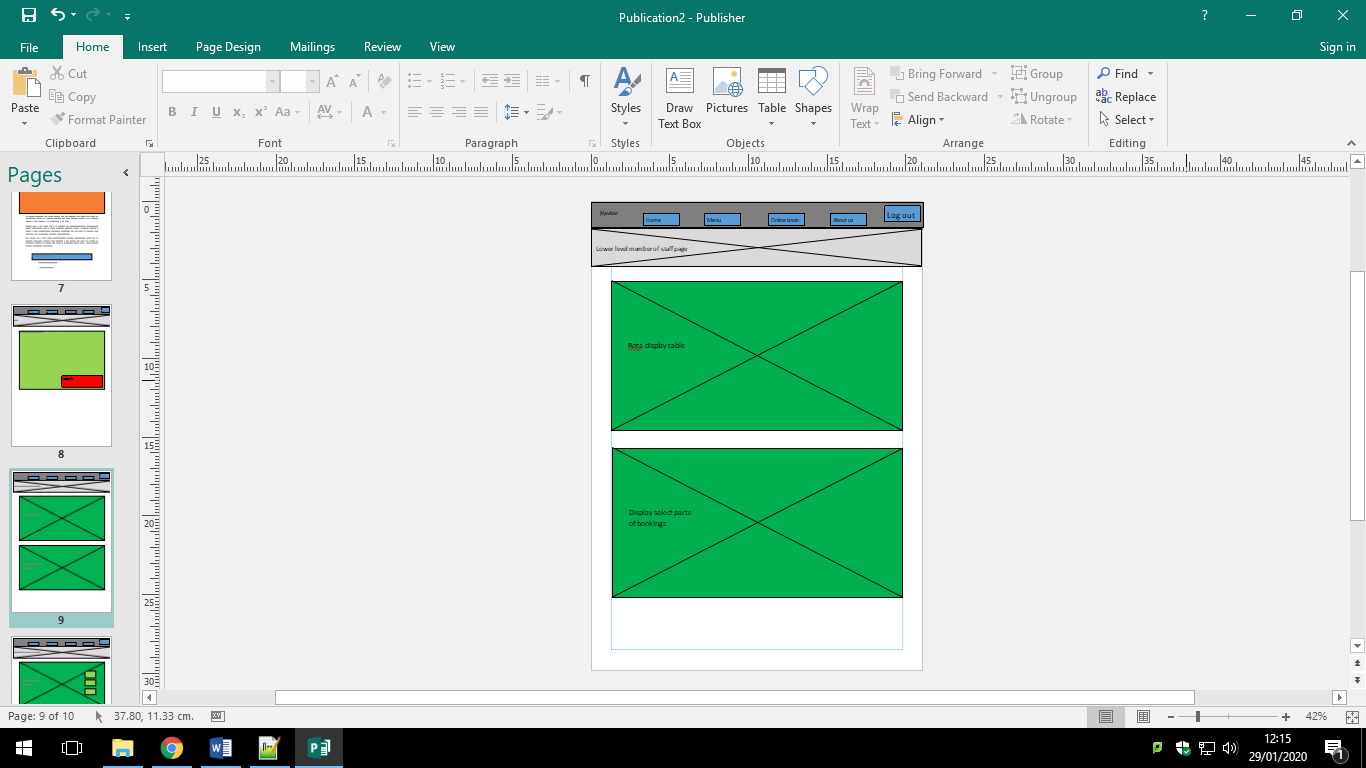
Bookings

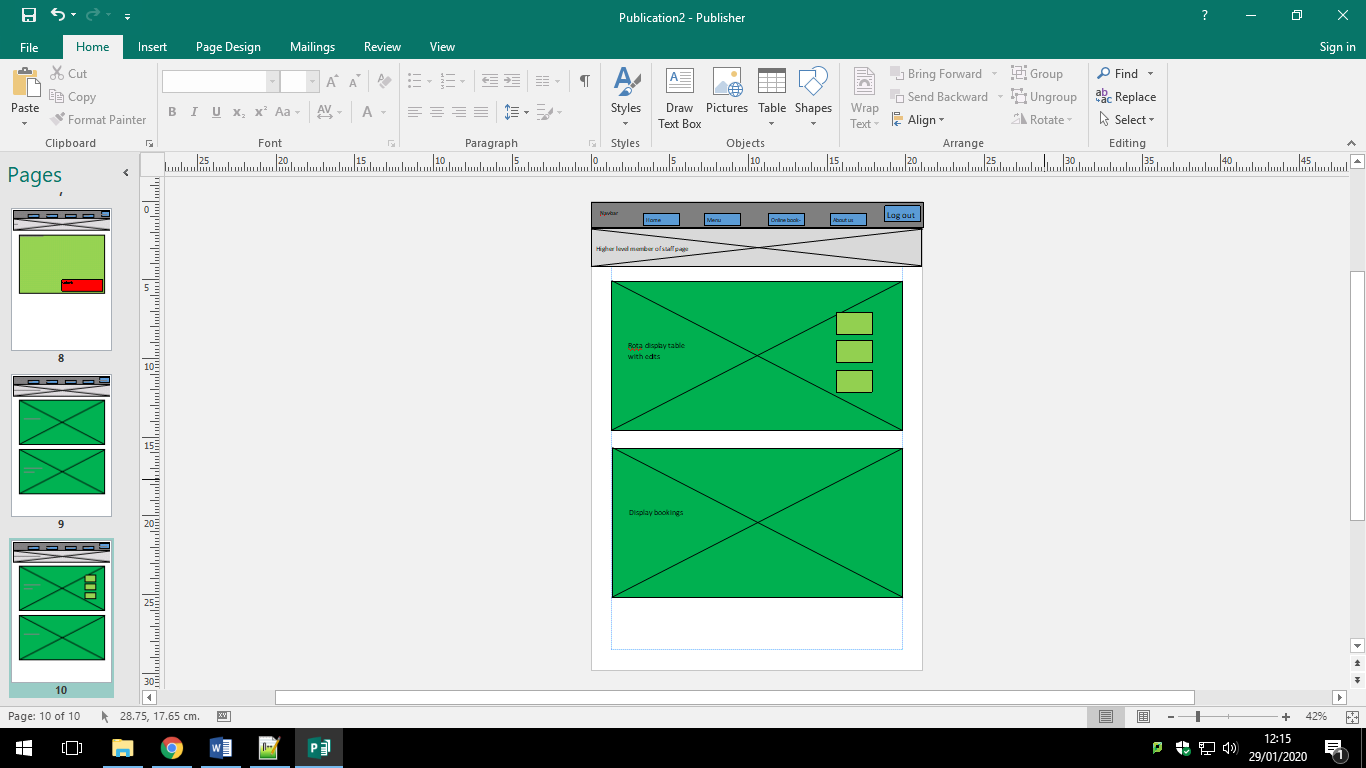
Booking a table

Booking a room

About us page

Login page

Display of rota and bookings for employees

Better display showing extra info for high level employees

## Identification of storage media

# Technical solution

## Overview guide

### The names of entities such as executables

### Data filenames/urls, database names, pathnames so that a third party can, if they so desire, run the solution/investigation

## Detailed Algorithm Design

# Testing

## Introduction and overview of testing

### (Black Box and White Box testing)

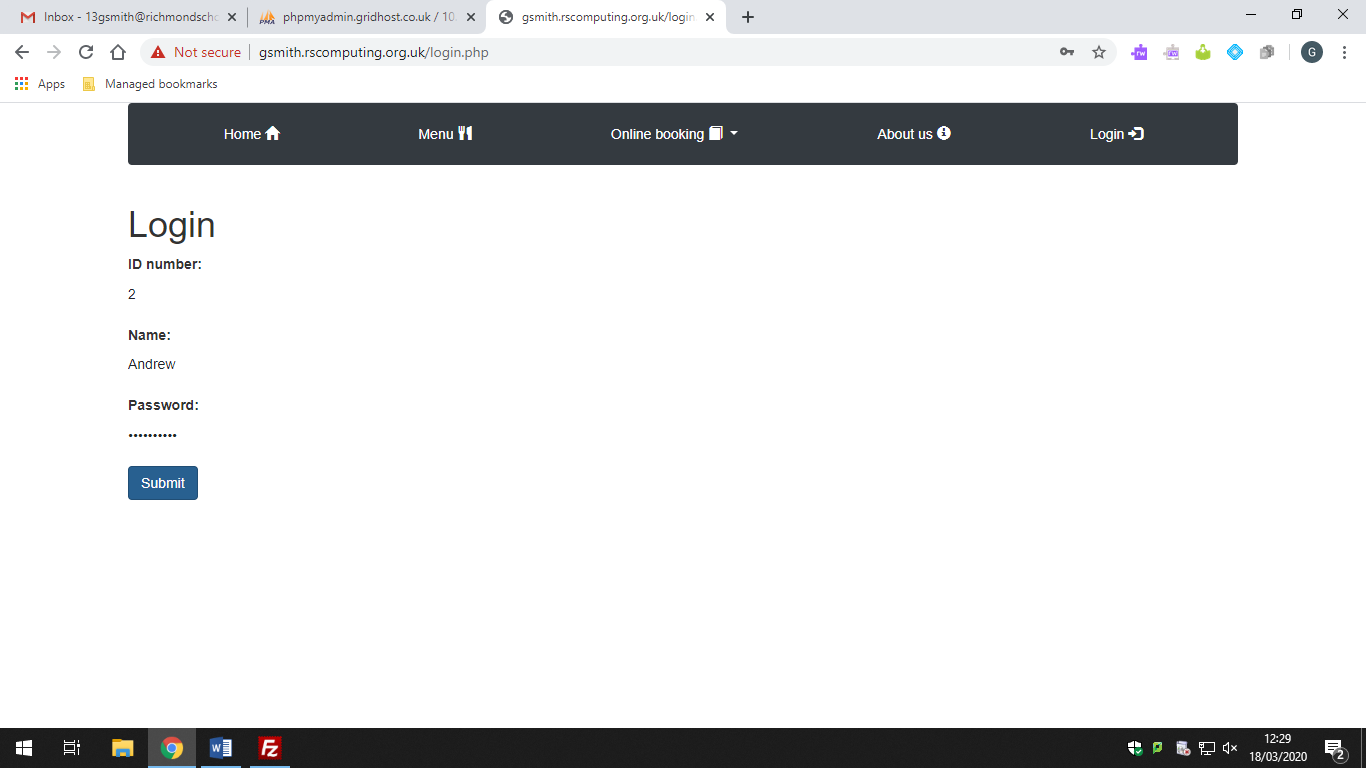
## Test Table

### (Test Number, Description of test, Test Data Type, Expected Result, Pass/Fail, Crossreference)

### Each screenshot should have a short annotation describing what is shown if not obvious

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test number | Description of test | Test data type | Expected result | Pass/fail | Cross reference |
|  | Logging in as a boss | Typical | Accepted login opening timetable.php with high level privileges | Pass | 1,2,3 |
|  | Logging in as a chef | Typical | Accepted login opening timetable.php with medium level privileges | Pass | 4,5,6 |
|  | Logging in as a pot wash | Typical | Accepted login opening timetable.php with lowest level privileges | Pass | 7,8 |
|  | Logging in with incorrect information | Erroneous | Pop-up box warning “Invalid login” | Pass | 9,10, 11 |

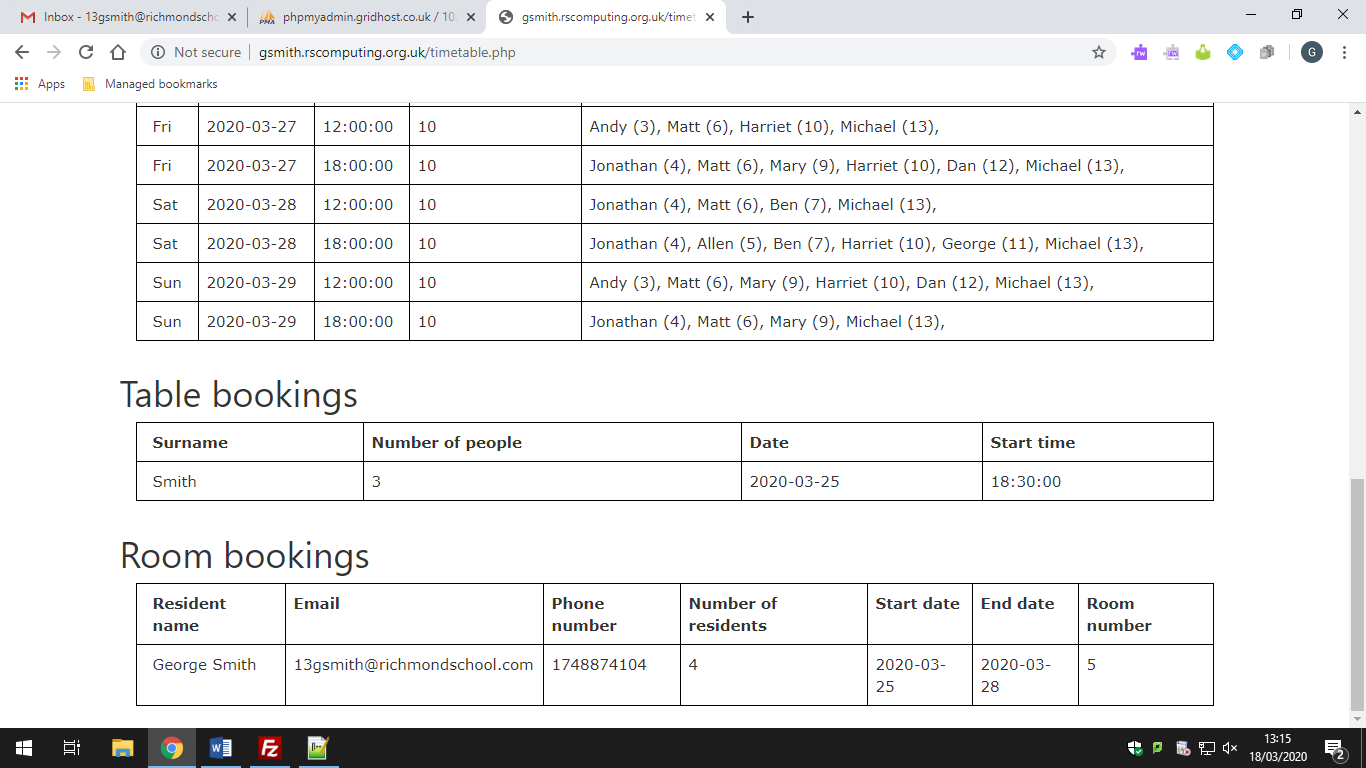
1. Logging in with Andrew’s, the boss, login information.



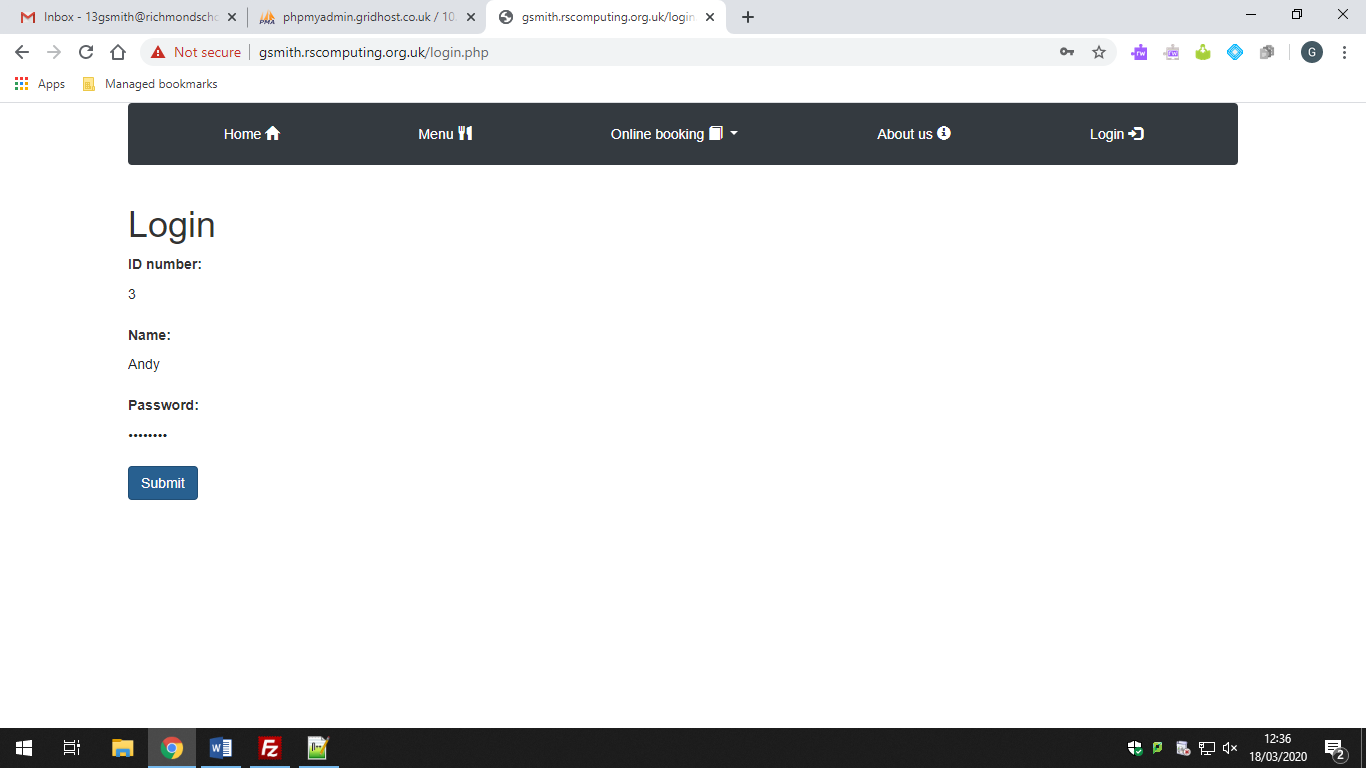
2. Showing timetable.php with all shifts viewable and buttons for registering a new employee and generating shifts.



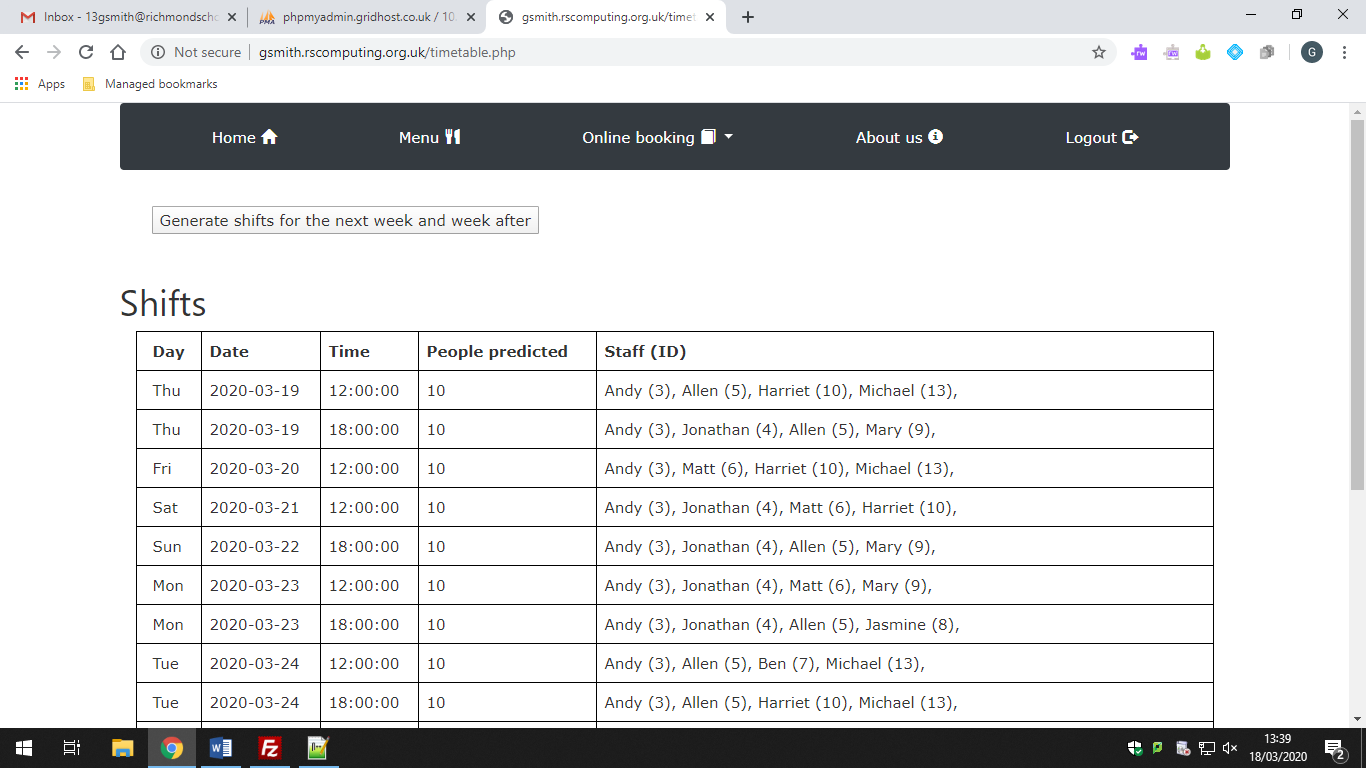
3. Showing all table and room bookings to the boss



4. Logging in with Andy’s, a chef, login information



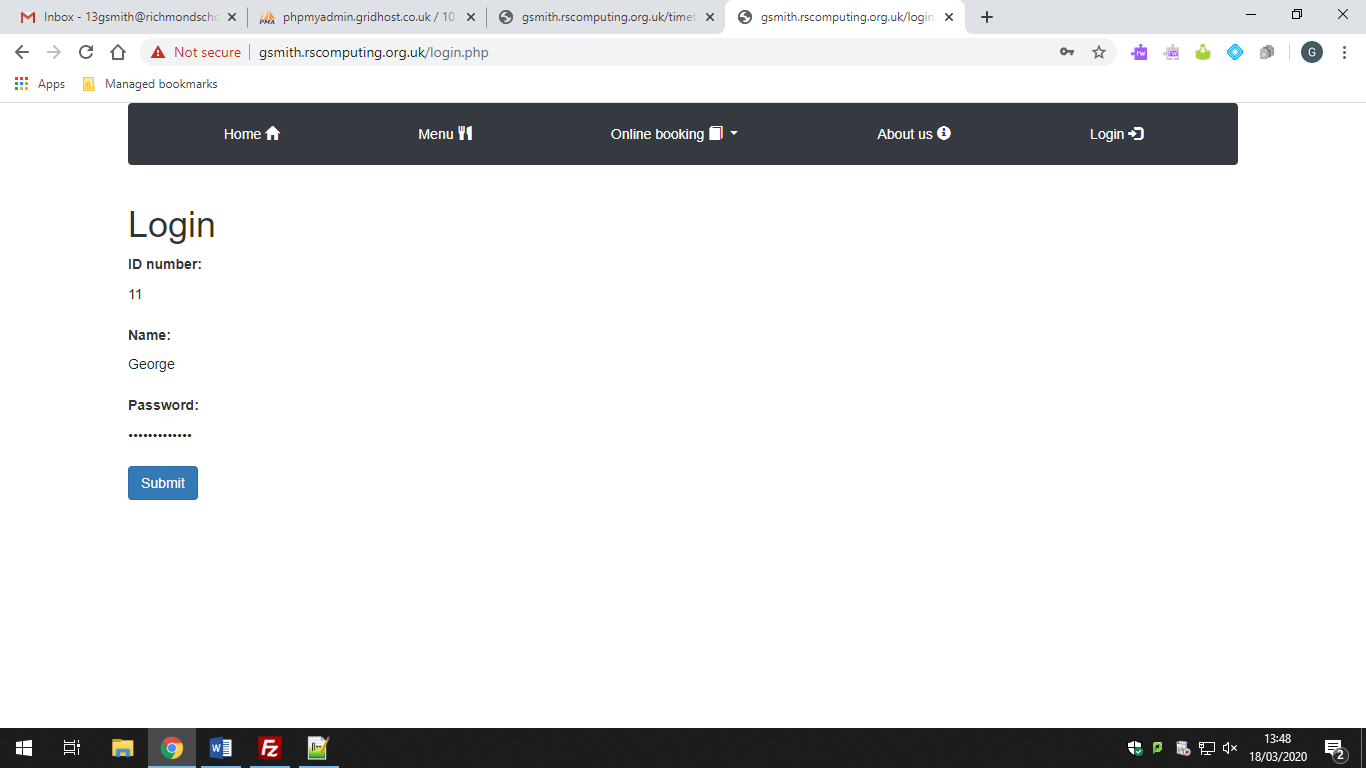
5. Showing timetable.php with all shifts viewable and a button for generating shifts.



6. Showing all table bookings to the chefs.



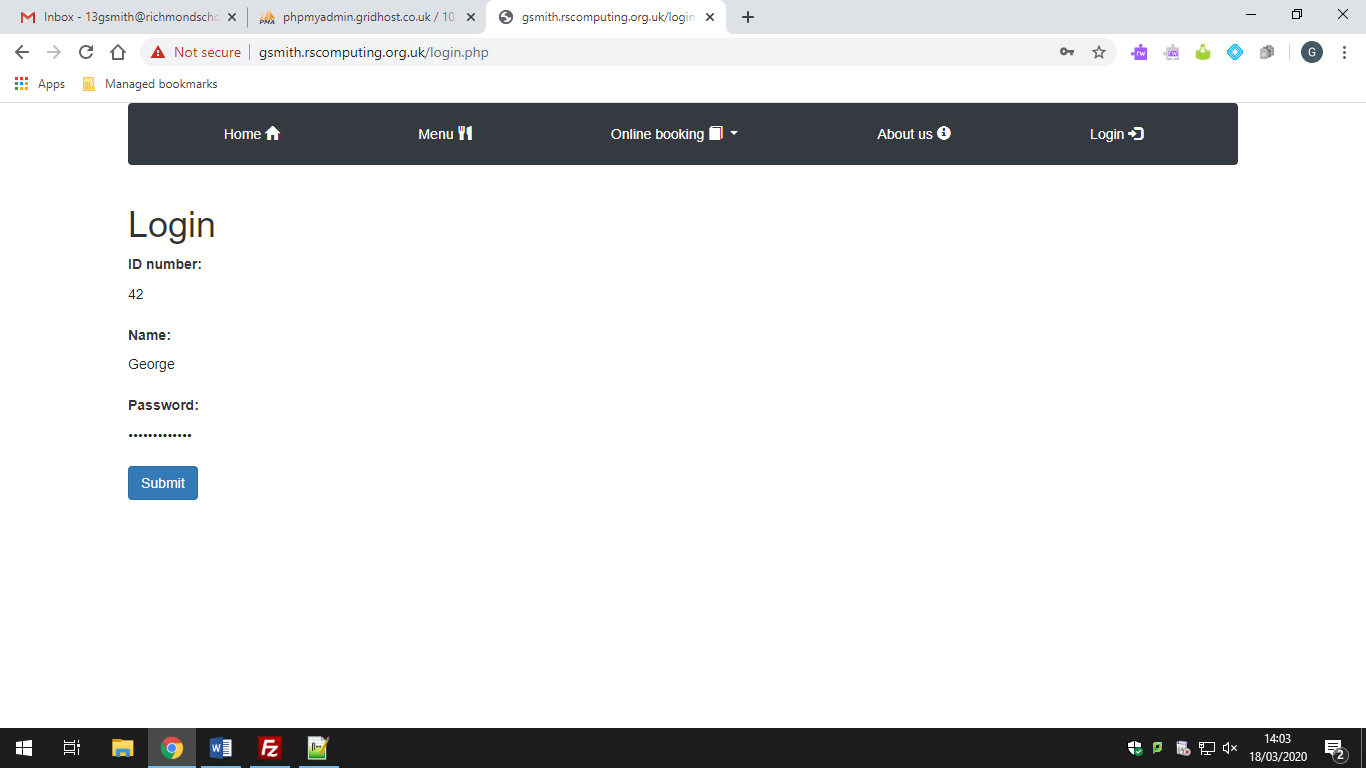
7. Logging with my, pot wash, login information.



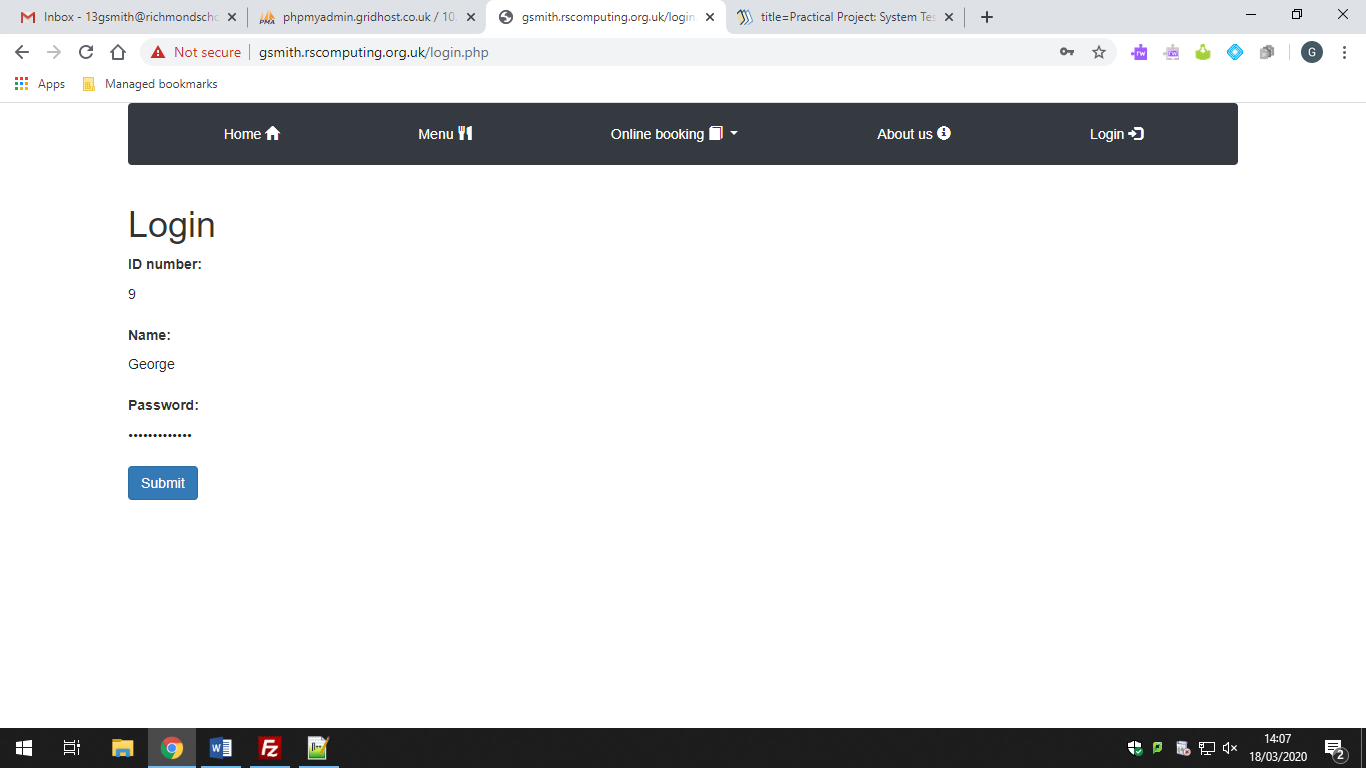
8. showing all employee specific shifts for this pot wash.



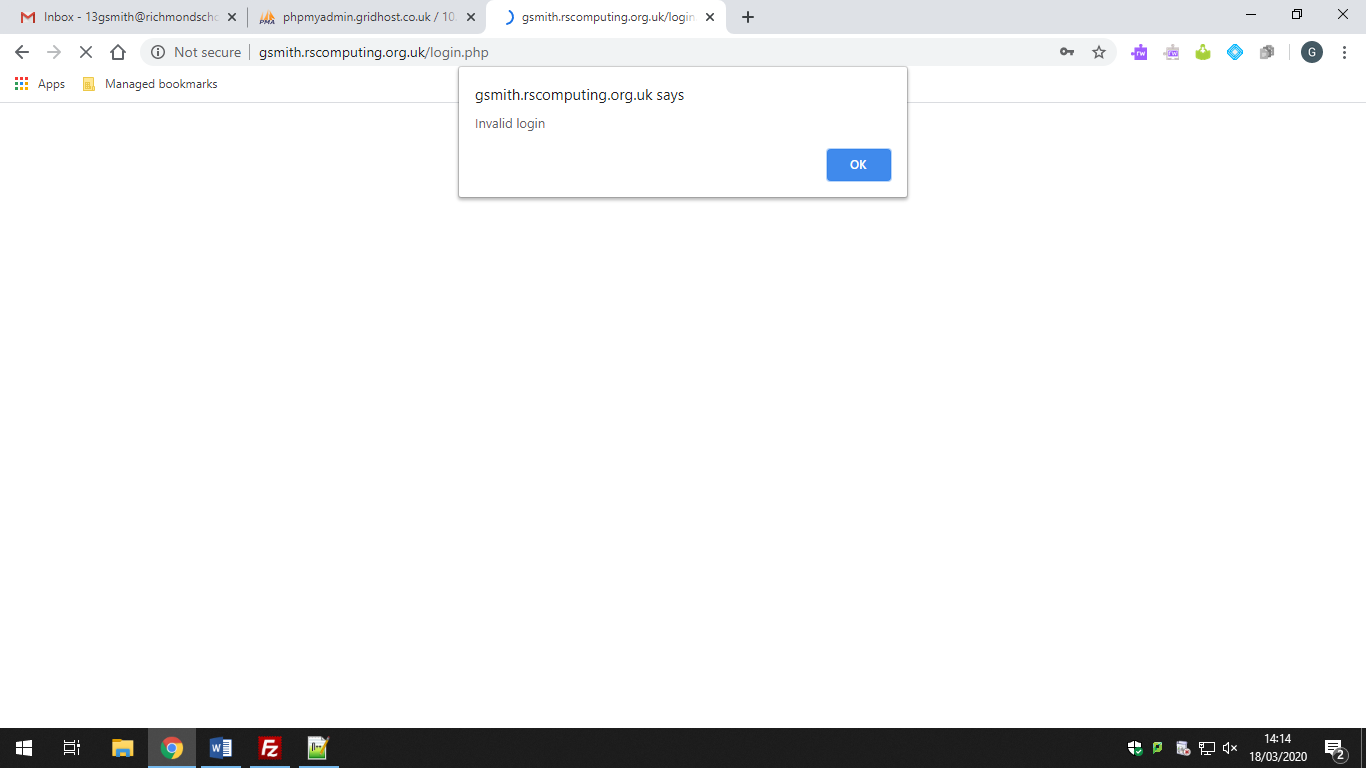
9. Incorrect login information. ID number too high



10. Incorrect login information. Using someone else’s employee ID number



11. Login error pop-up box



# Evaluation

## Did you meet your objectives?

## Feedback on requirements

### Analyse feedback

## Extensions