Prototype

Computer Science NEA

21HeffernaJN42

Contents

Features included in the prototype

Identify the different aspects of the system you will be including in your Prototype and justification as to why you have included these.

Login Page

- In order to input the user's passcode, I have created a function that will output a series of buttons in the layout of a keypad. This essentially removes the problem of invalid/erroneous data entries as only integers can be entered through the pushing of buttons.
- Once the user has entered their password and clicked confirm, the system will then execute a SQL query to determine whether there is a record/account linked to the passcode provided. If the SQL query provides a result, the system will then store the AccountID of the record, in order to implement user access levels.
- If the SQL query has no result, an error message will appear stating that the passcode entered is incorrect and the to retry.

These features have been added as they provide the essential security needed in the system, such as user access levels and passcodes for entry of the system.

Booking Page

- **Date Selection**
  - During the initialisation of the system, the minimum date that can be selected of the calendar widget on this page is set to the current day. This is done by obtaining the current date, then set the minimum date of the calendar widget to that date and set the selected date to the current date to prevent the system from automatically selecting a past date.
  - Once a date has been clicked, a function is run which will obtain the selected date and return that value in python format. This returned in python format in order to perform calculations and commands on this value.
  - In addition to this, once a date is selected, the Party Size button will be able to be selected.

  These features have been implemented to allow ease of date selection while also removing the possibility of invalid data entries such as past dates, etc.

- **Party Size Selection**
  - Once a party size has been selected, the Time Selection button will be able to be selected
  - In order to get the party size, a function is run to determine this. In particular, if the custom party size checkbox is selected, the party size will be the value in the spin box below the checkbox. However, if the checkbox isn't checked, the party size will be determined by the index of the selected item in the list widget.

- Another function included clears the values of both list widget and checkbox. For example, the function will set the selected value of the list widget to -1, and it will uncheck the custom size checkbox.

These features have been implemented to successfully obtain the correct party size based on the user's inputs.

- **Time Selection**
  - Similarly to the Party Size Selection page, data is inputted through the use of a list widget. Once an item in the list widget has been selected, a function is run which will determine what the time they have selected is.
  - The function determines what time they have selected by firstly obtaining the index of the item that have selected in the list widget. After that, it will then get the text placed into the item of the selected item.
  - Another function included converts the text of the selected time into hours and minutes.
  - Another function, which is activated when tabbing out or when the booking has been complete, clears the list widget input by setting the selected index to -1.
  - Once a time has been selected, the table selection will then be able to be pressed, which allows the user to select the tables for the reservation.
- **Table Selection**
  - In order to display the tables in the restaurant, a function is run which loops the creation of checkboxes, which represent the tables. A total number of 15 tables are outputted with the function.
  - In addition to this, to avoid overbooking, tables are disabled. The algorithm performs an SQL query which selects the tables that have been booked within a 2-hour radius of the booking time. This is completed using loops which increment the SQL query time parameter by half an hour each iteration.
  - Another function I have added is to obtain an array of the tables that have been selected by the user. This function is run when any checkbox is clicked, whether it be to select or deselect. Essentially, this function determines whether the checkbox has been selected or deselected, then it will append or remove the table number from the array depending on if the checkbox has been selected or deselected.
- **Guest Details**
  - In order to store the details of customers in the database, the system must ensure there are no duplications of data to avoid errors and incorrect data being outputted. I have included a function which is used to obtain the CustomerID, which is the unique identifier for the table, for the new customer. This is done by first identifying whether there is already a customer registered with the email provided. If there is an email registered, the system will use the CustomerID of the record with the corresponding

email. However, if there is no record linked to the entered email address the system will execute an SQL query which will return the maximum value of CustomerID in the table Customer. The CustomerID of the guest to be added will be the value returned from the query + 1.

- Once the Confirm button has been pressed, the system will begin to create the booking. Firstly, the system will obtain all of the reservation details by storing the results of the functions used in previous pages. Then the system will execute a series of SQL queries which will insert the reservation details into the correct tables. This is crucial as it ensures that the correct data can be extracted from the correct tables. This will ensure the whole system is functioning as the majority of the system depends on the extraction of data as operations are completed on this data.

Seating Page

- During initialisation, a function is run which outputs a series of push buttons through the use of loops. These push buttons represent the tables in the restaurant. Running the function will result in 15 push buttons being outputted onto the screen.
- In order to identify which table is which, a function is run which determines the table's capacity, then it assigns the corresponding table image. This helps the staff members identify the tables easier, instead of just looking at plain boxes.
- In addition to this, the system runs an SQL query to obtain the array of tables that are currently seated. Once these tables have been obtained, a command is executed which sets the image of the push button to a certain shade of the default colour to help with identification.
- Once a button is pressed, a function is run which will change the index of the widget to the order food page. This page is used to order food based on the table clicked.

Calendar Page

- Once a date is selected on the calendar widget, a function is run which will create an SQL query to obtain all of the bookings for the selected date. This query is then passed into another function.
- Once the query has been passed into the function, the function will then execute the query and populate the table with the results of the query. This is an essential part of the system as staff members need this to prepare for upcoming reservations.

Order Food Screen

- Once a button representing a menu item is clicked, a function is run which obtains the item details of the item, such as the OrderID and price. The OrderID is then used to insert a record into the table TableOrder. In addition to this, a row is added to the table which includes the items name, quantity, and price.
- If a row on the table containing the orders is double clicked, a pop up will appear. This pop up will allow the user to alter the quantity of the item that have ordered through the use of a spin box.

- If the Close Table button is clicked, the system will execute an SQL query to update the record of the corresponding SittingID to set 'Active' to False. This notifies the system that the table is no longer in use and can be used by new customers.
- This page has a collection of subpages which contains the buttons of the items in each category of the menu. The user can toggle between these subpages by clicking the corresponding button which represents the category of the subpage.

Add Item Page

- When this page is opened, a function is run which will display all of the current menu items in the system. These will be outputted in a table which displays the OrderID, Name, Category, and Price of the items.
- Once the 'Confirm' button is pressed, the inputs of the user and stored as variables in the system. A confirmation page will appear which will confirm to the user the data that they have entered. If this is all correct, the user will then select the confirm button, which executes a series of SQL statements that will result in the item being added onto the system.
- Once the item has been added to the system, or the back button has been clicked, the system will clear the input fields in order to provide the same experience when adding another item.

Remove Item Page

- Once this page is opened, a function will run which outputs all the current menu items in a table. This table displays the OrderID, name, category, and price of the items.
- In order to select the item you want to remove from the system, the user must click on a row in the table. Once the item is selected, the user will then press the remove button, which will display a confirmation pop-up.
- This popup will contain the details of the item that the user has selected. If the user then clicks the remove button, a series of SQL queries will execute that will remove the item from the system. However, if the user clicks the cancel button, the popup will close, and the item will remain in the system.

Alter Item Page

- Once this page is opened, a function is executed that will fill the table with all of the menu items currently in the database. This includes the OrderID, name, category, and price of the items.
- In order to alter an item, the user must double click an item on the table. Another window will then appear with a series of input fields. The input field will automatically be filled with the information of the item selected previously. The user can then alter this information to the correct values.

- Once the correct values have been inputted, the user can then press the 'Alter Item' button. This will then execute a SQL query that will update the chosen record to the values inputted.

Features not included in the prototype

Identify the different aspects of the system you will be including in your Prototype and justification as to why you have not included these.

As this project provides the user with an eclectic supply of features, it was difficult to provide all these features to the highest extent.

In particular, the features that I have failed to include in the prototype include:

Different Access Levels

- In my opinion, the main purpose of a prototype is to develop and demonstrate the functionality of the program. In particular, the key points of this program are the Reservation Booking process and the food ordering process. As I have already developed an algorithm that obtains the accountid of the logged in user, it will be simple to add different access levels.
- Different Access Levels can be added by only allowing the function to be run if the accountid is the one of the admin.

Alteration/Deletion of reservations

- As the ability to alter or delete records is present throughout the program. In particular, this is seen where the user can alter or delete menu items. I believe that it would not be essential to include the alteration of reservations as the process would be similar to the process stated before.

Validation

- As validation is critical when using databases, I wanted to leave validation as one of my last developments. This is because I wanted my program to be in its final stages so I can complete the validation without making major alterations.
- I plan on adding validation on all entry fields in order to ensure that all data is sensible and follows the guidelines of the database

Statistics/Graphs

- In my prototype, I did not include any statistics for the restaurant, such as the total revenue, most popular items, etc. The main reason I did not include this in my prototype is that I believe that it is not essential in the performance of the program. This means that the user will still be able to complete the tasks the system was programmed to do, even with the exclusion of this feature.
- In addition to this, I have had multiple past projects that include some sort of statistics or graphs. This means that it will be less of a struggle to program this feature due to the prior experience.

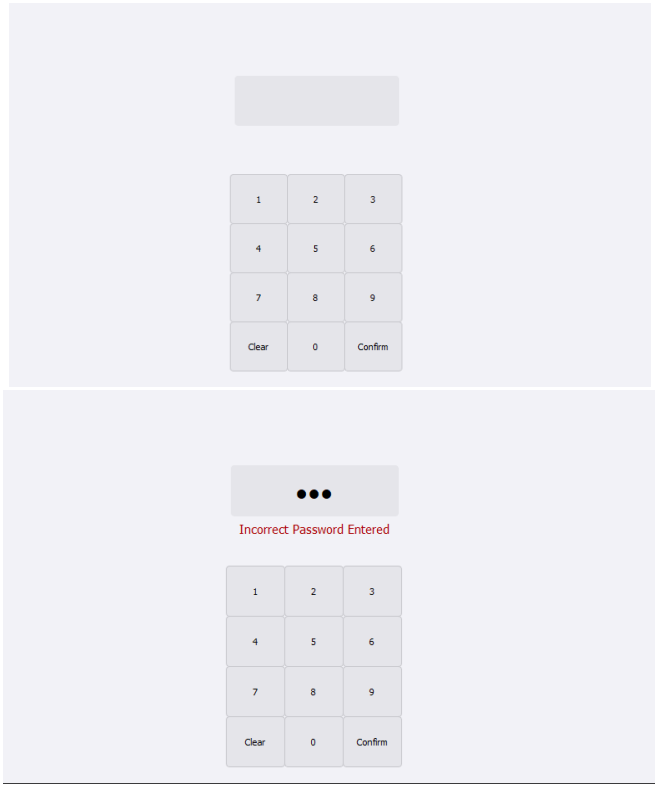
Display number of customers

- Similar to the reasoning behind the statistics/graphs, I have not included the ability to display the number of customers seated at a certain time. This is not included in the prototype as I believe that it is not essential for the performance of the system, as it is only seen as an add-on.

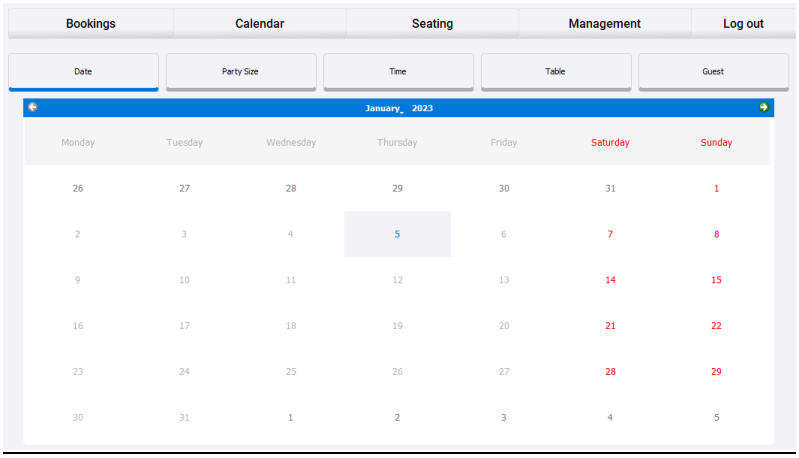
Prototype

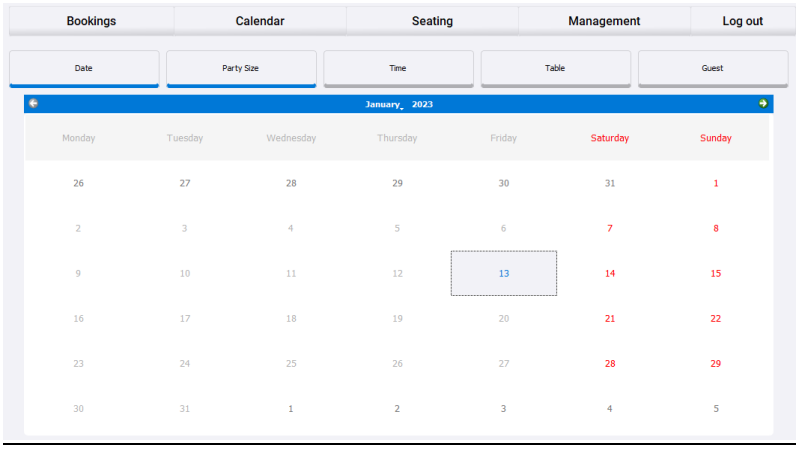
Include print screens of every screen of your system you will be including in the prototype. This should be in a logical order, generally how the user would see them. Make sure you show these with realistic data and any error messages/validation this will generate.

Login Page

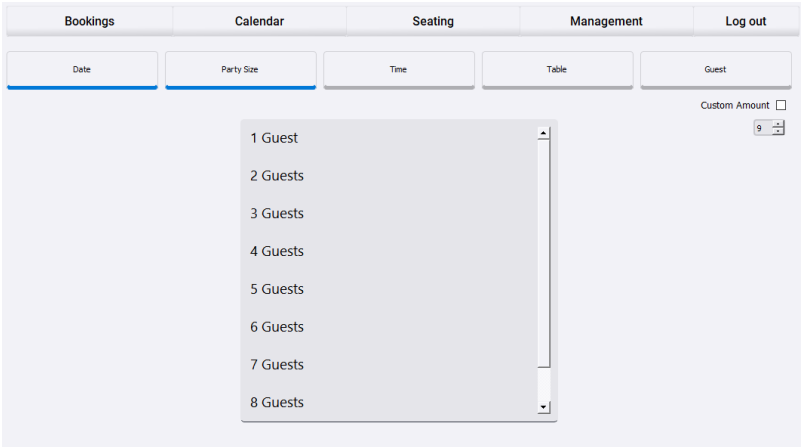


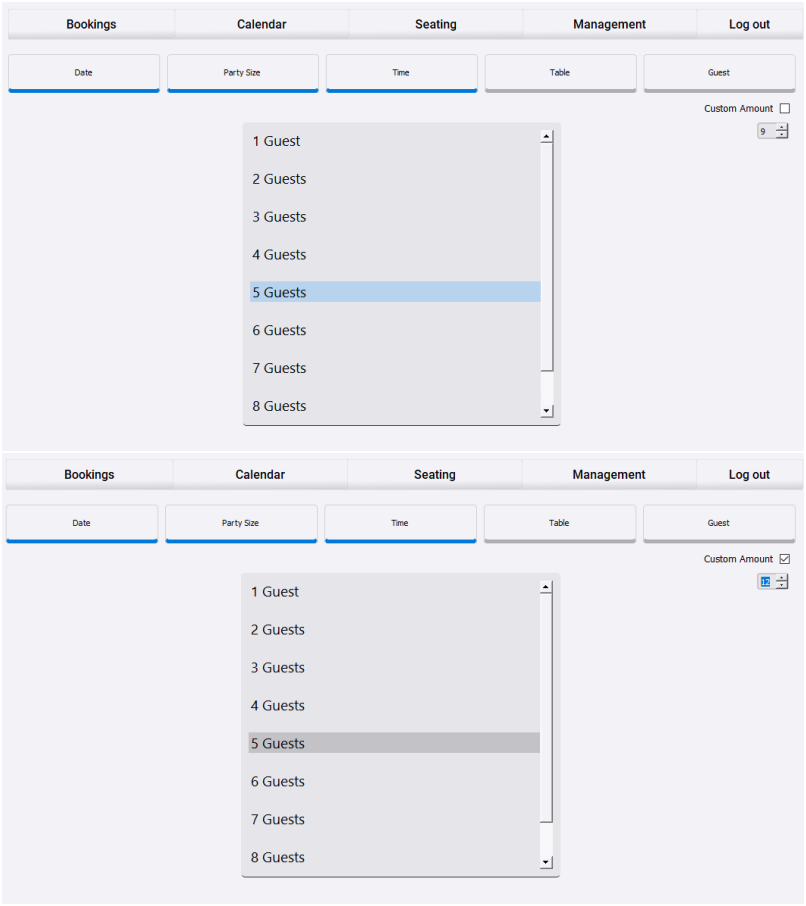
Booking Page – Date Selection



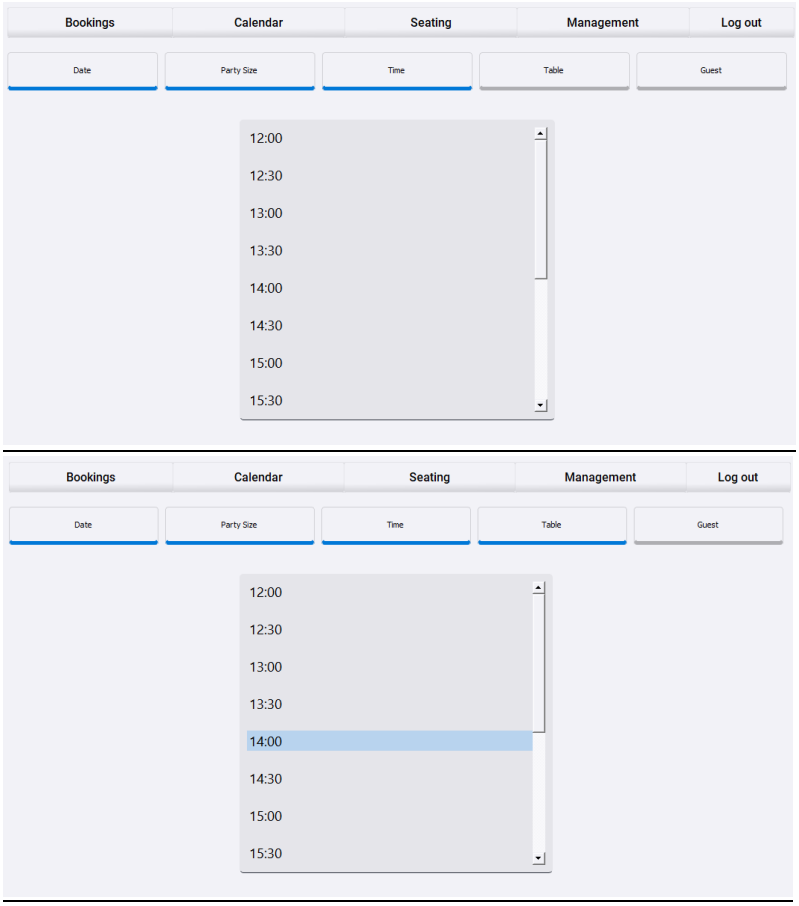


Booking Page – Party Size Selection



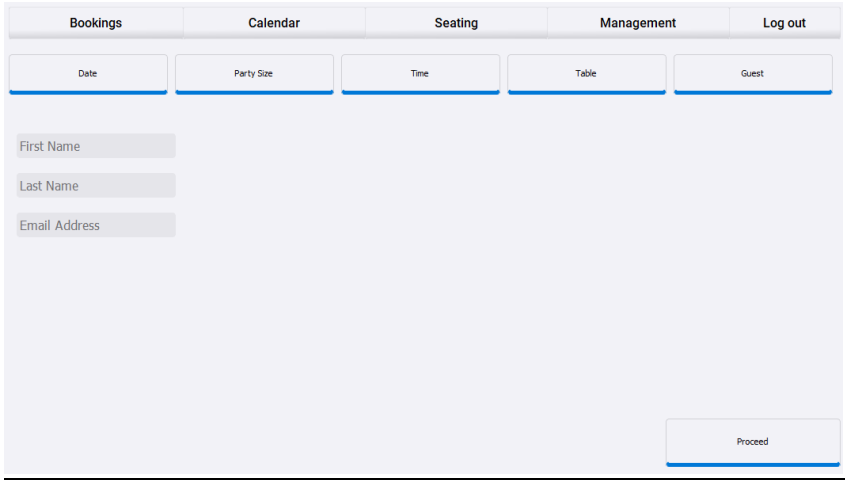


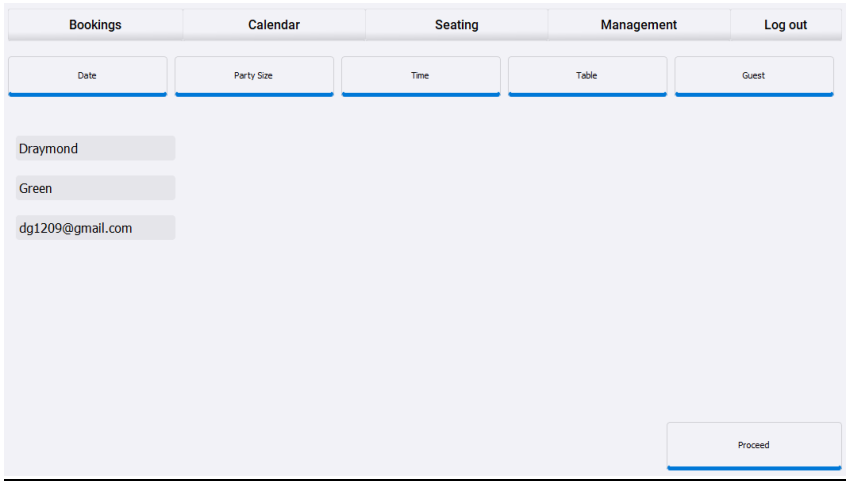
Booking Page – Time Selection



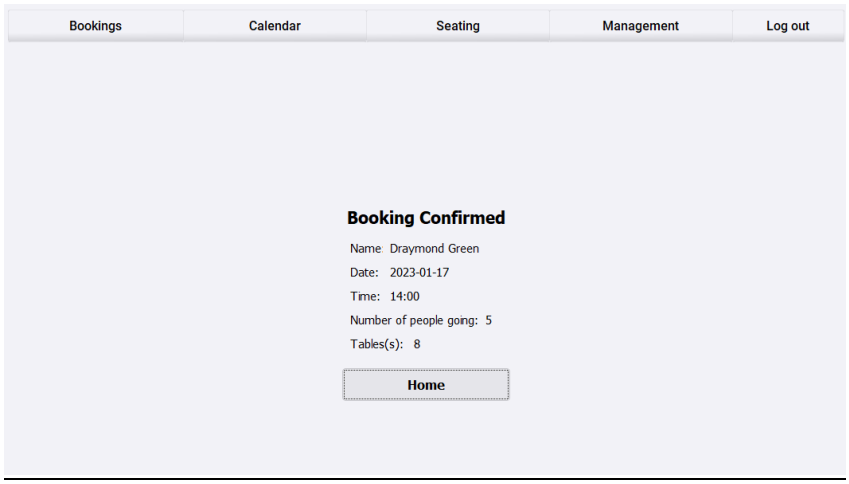
Booking Page – Table Selection

Booking Page – Guest Details

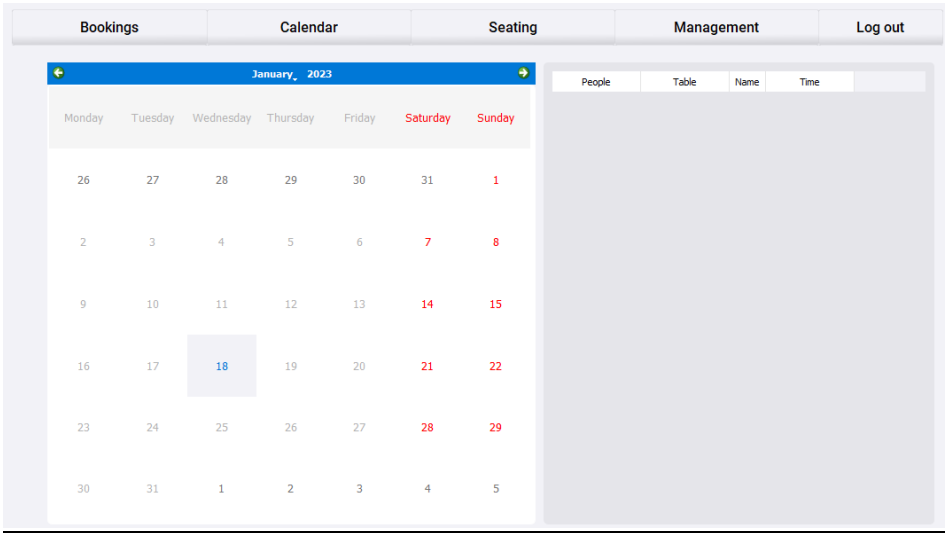


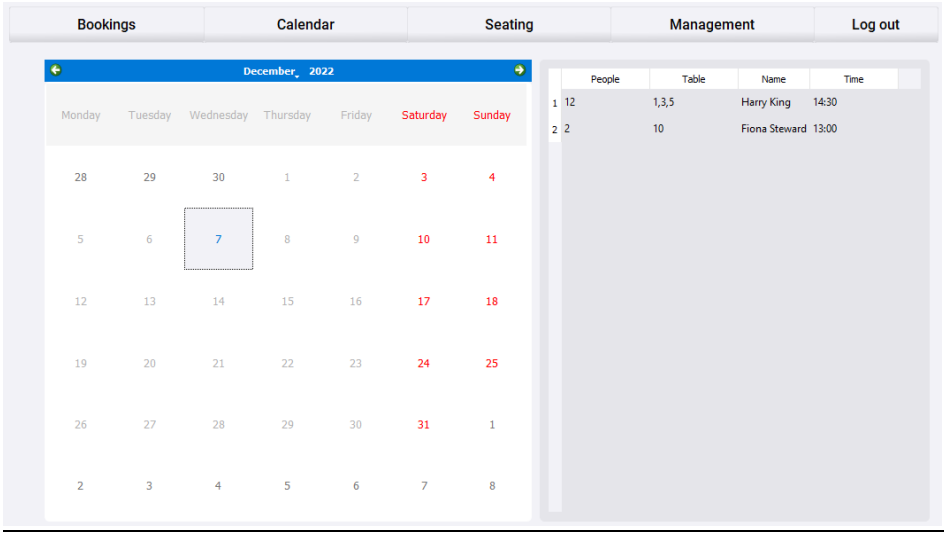


Booking Page – Reservation Confirmed



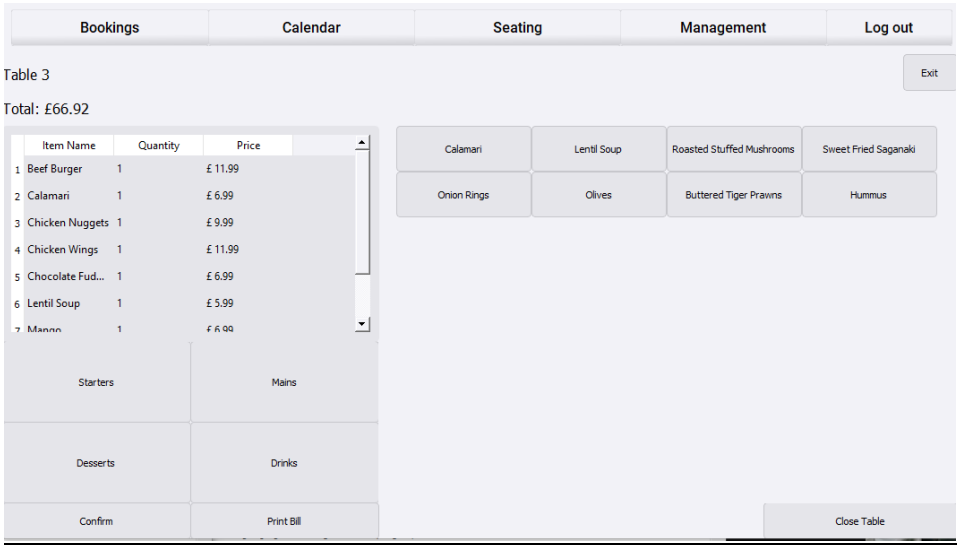
Calendar Page



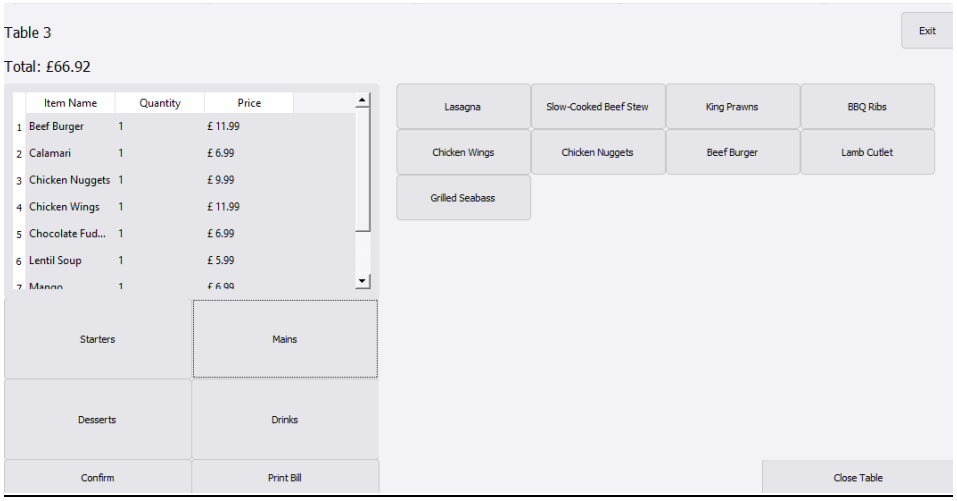


Seating Page

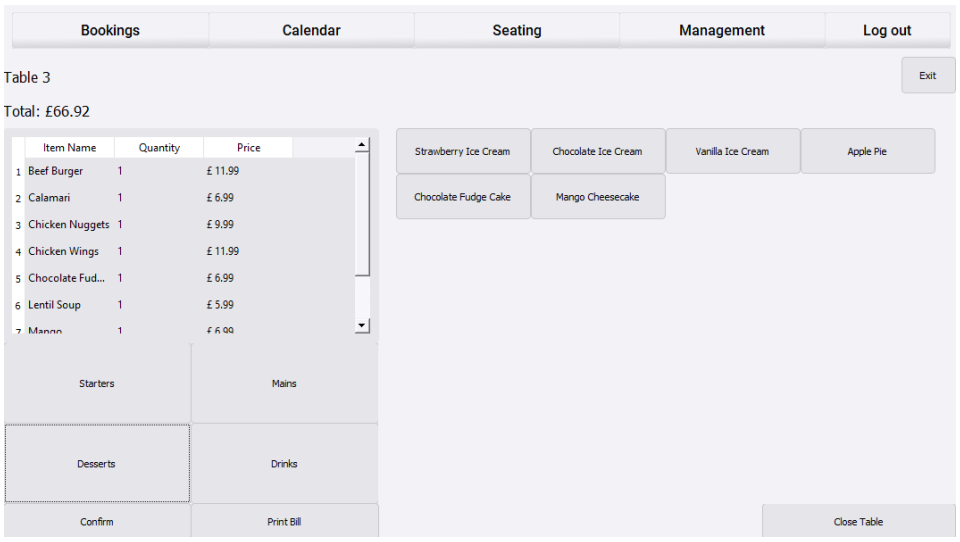
Order Page – Starters



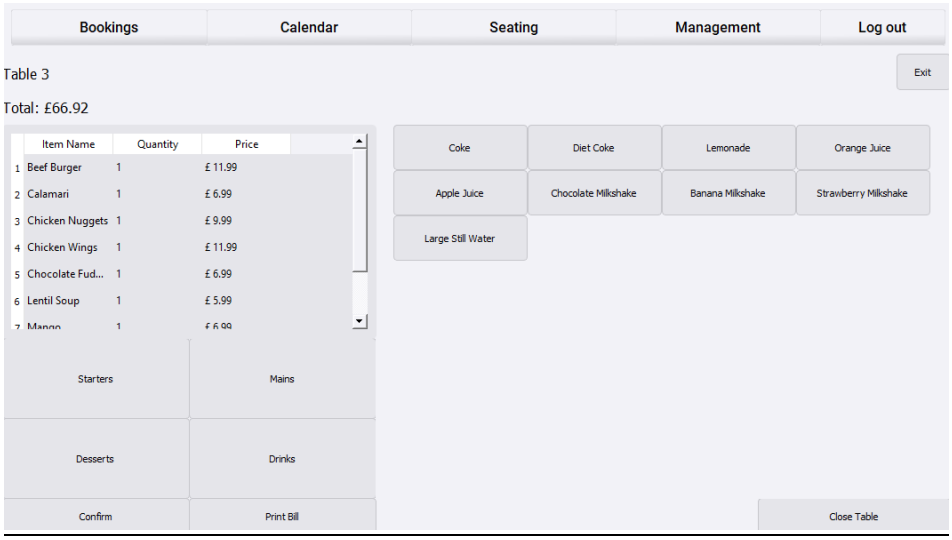
Order Page – Mains



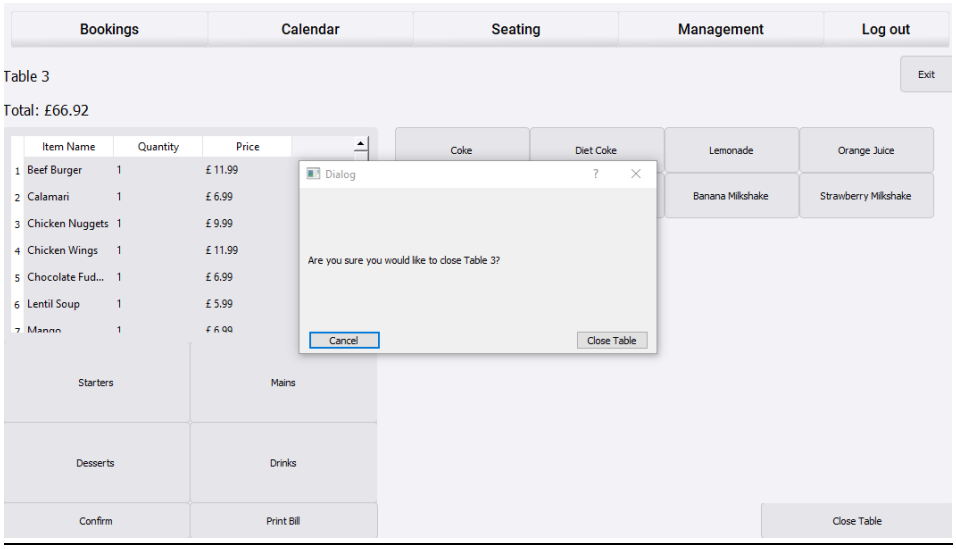
Order Page – Desserts



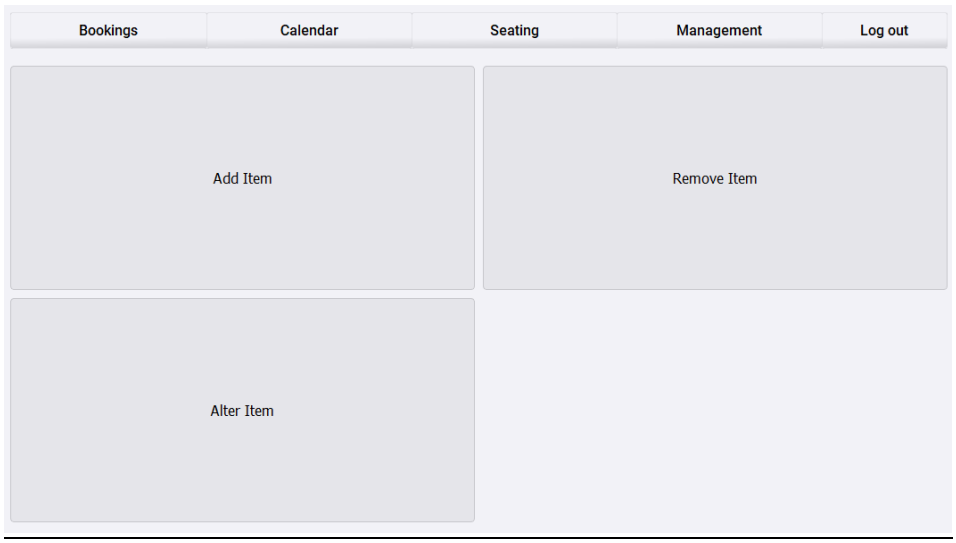
Order Page – Drinks



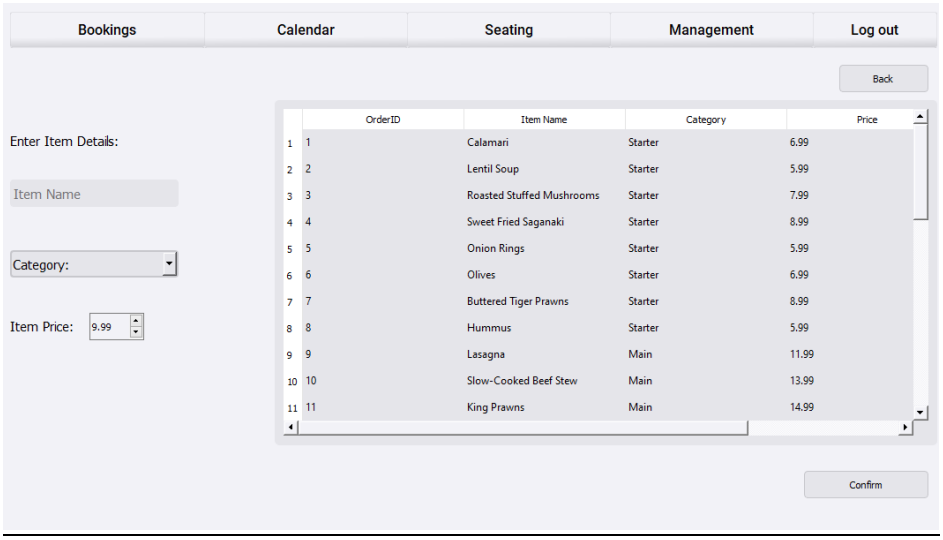
Order Page – Close Table Confirmation



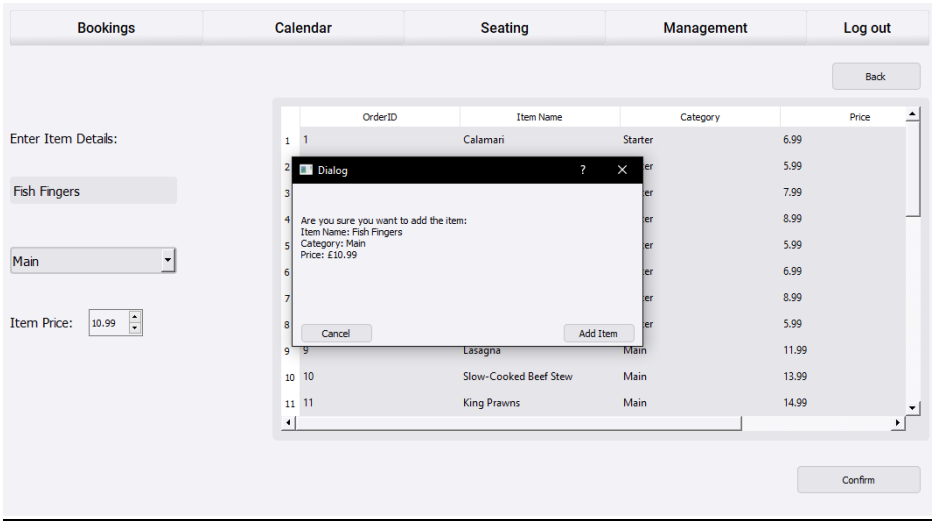
Menu Management



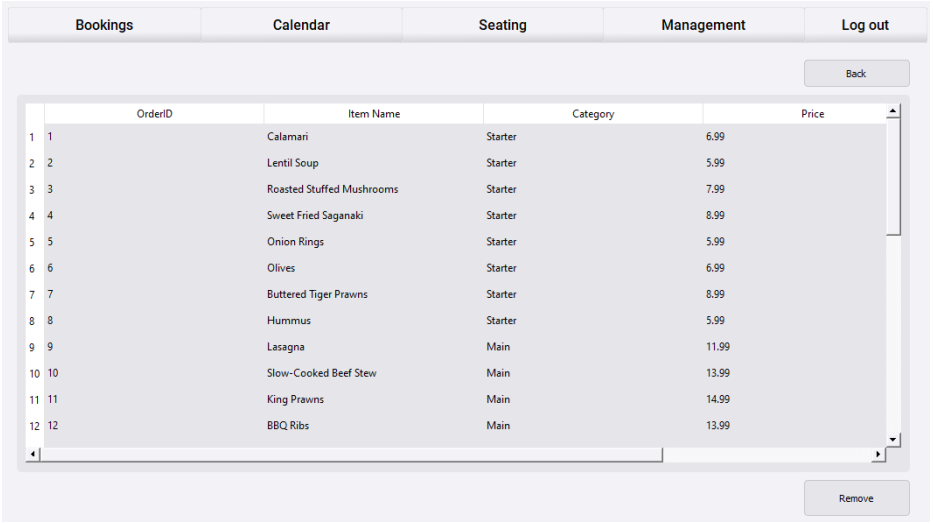
Menu Management – Add Item

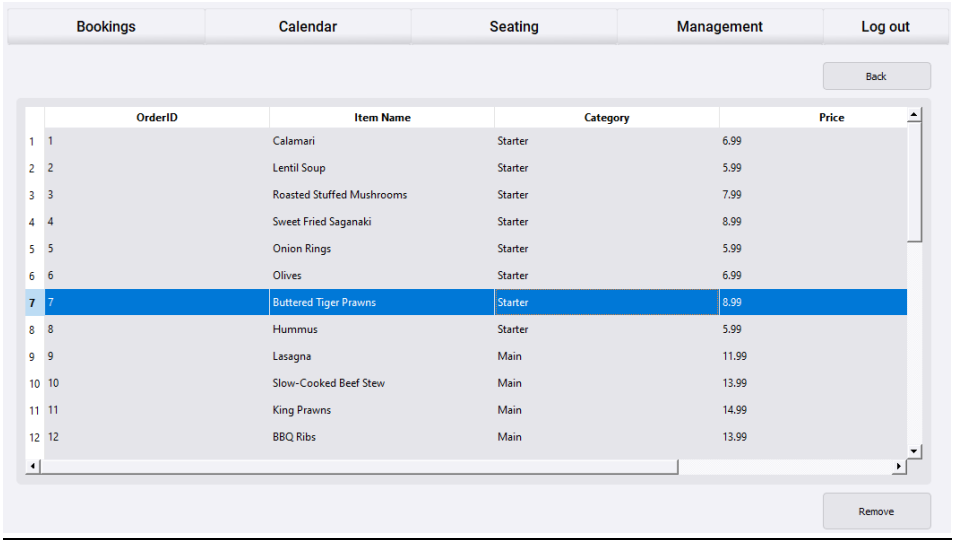


Menu Management – Add Item Confirmation

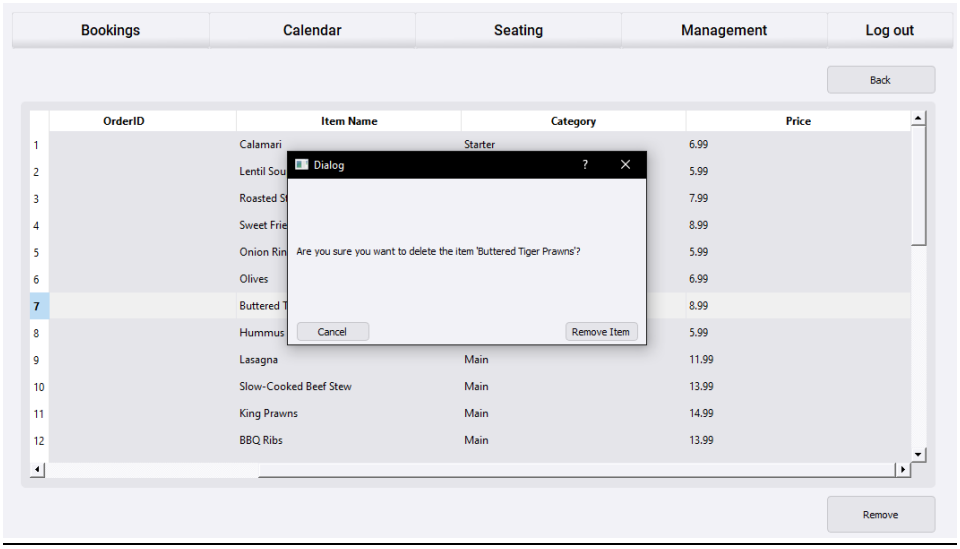


Menu Management – Remove Item

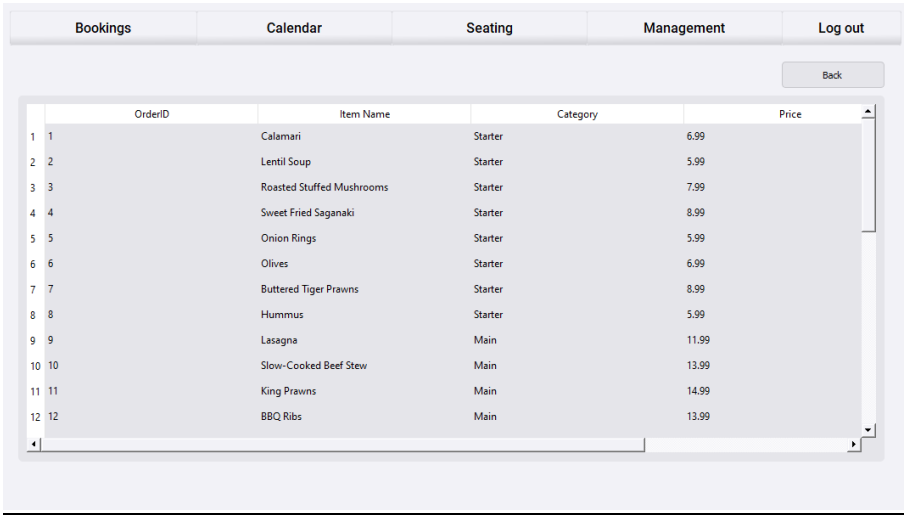


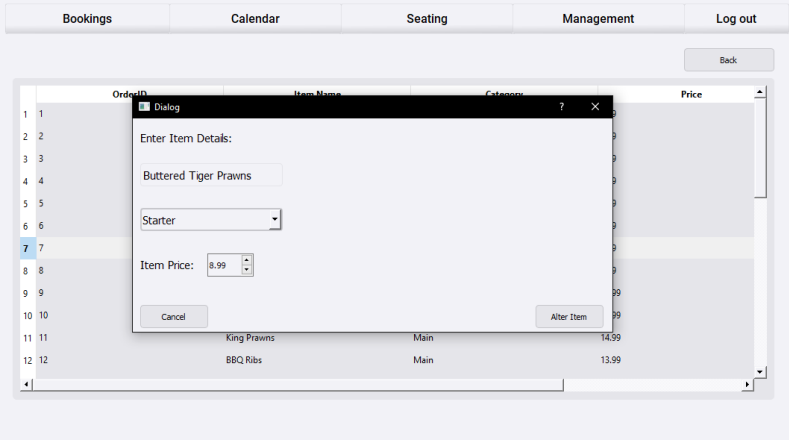


Menu Management – Remove Item Confirmation



Menu Management – Alter Item





Evaluation of current system

Evaluate your system, you should have completed enough of the user requirements to evaluate against these. Justify the good features of your system, highlighting these (brag a little)

Overall, I believe that this prototype has provided the user with a substantial amount of features that meet the user's requirements.

In particular, the requirements met include:

Straightforward Booking Experience

- As this was my main focus when developing the program, I believe that this requirement has been met. This is due to the fact that my booking process follows a smooth, structured process.
- Structure is an important aspect for the booking experience. Therefore, I have added a feature that will only allow the user to proceed or toggle to the next page once the necessary details of the current page have been filled. This ensures that the booking process follows the same chronological order.

Adaptability

- As errors or changes may be prone when running a restaurant, I believe that adaptability must be an essential requirement for the system.
- This adaptability is displayed in the Menu Management section where users can alter, add, or remove menu items. Not only does this allow the user to create new items, but it also allows them to correct or remove previous mistakes/changes.
- In addition to this, this system provides adaptability for the customer as well, since it allows the user to book multiple tables for one reservation. This can be very useful for large party sizes.

Security

- In order to ensure that only staff members can use this system, I have included a login page which will verify that the user is authorised.
- Also, when altering or deleting data from the database, the system will provide the suer with a confirmation message to ensure that they are making the right decision

and to avoid any mistakes. This is very important for the system as the majority of the system is linked together through the use of tables, etc.

Shortcomings and improvements

Explain the shortcomings and limitations or that you have identified in general about your current system. For each of these you must have specific suggestions for improvements.

During my presentation, my peers have identified certain changes that need to be made in order to further advance my program. Some changes include:

Some features difficult to identify

- In particular, some features such as the custom guests amount checkbox is difficult to identify. This can easily be fixed if I make the text and checkbox bigger, while also including instructions to guide the user.
- In addition to this, it is difficult to identify how many guests are remaining to be seated as the only identifier of the table capacity is the colour and design of the table. This can be fixed by adding an identifier at the bottom of the screen which identifies how many guests have seated and how many are remaining.

Aspect Ratio

- In order to support a variety of devices, the aspect ratio of the program is important. As a result of this, I would like to change the aspect ratio of the program to allow full screen usage. This will provide the user with a pleasant experience as it will become more visually appealing.

Alter/Cancel Bookings

- Since this feature is present in the program, but instead of altering booking, it alters menu items, I have left this out of the prototype as it will be similar code to the previously stated function. However, this will be present in the final program as it will be a useful and important feature for the system to have.

Highlight Which Tab is Selected

- As it is difficult to identify which page the user is currently viewing, the user may be confused and may click other tabs which could result in a loss of the data they were inputting. In order to fix this, I could implement a feature into the system which highlights the tab which corresponds to the page the user is currently on.

No Statistical Data of the Restaurant

- Another limitation is that my system lacks any statistical data that may be useful for the manager. Some statistical data may include the distribution of table bookings, distribution of food sales, etc.
- This can be implemented by adding another section to the management page which will display all this data whether it being displayed through a graph or table.