4 – Prototype

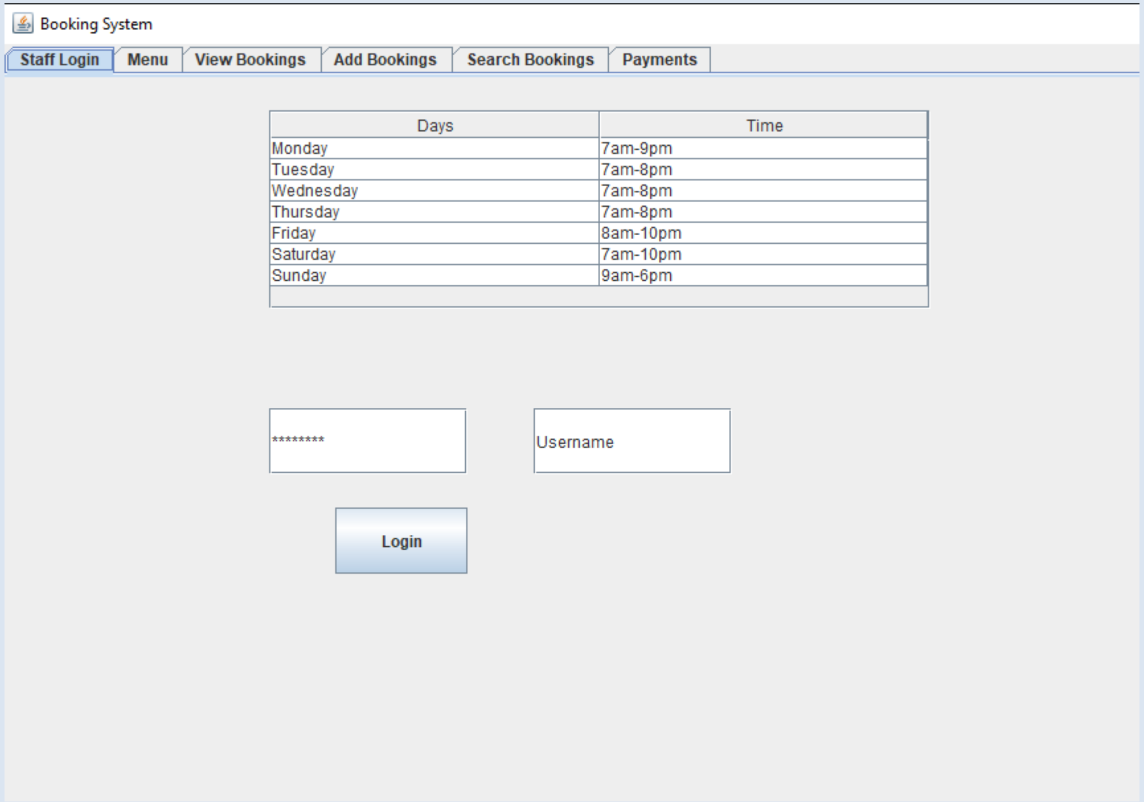
Contents

4.1 Justification of Prototype

#	Area of System	Included / Omitted	Justification
1	Staff Login screen	Included	I have included the screen for the staff login which is there to display the functionality of what would be in the final implementation, therefore necessary for the prototype. I have also included a table to display opening times for the business which is necessary to have the functionality of reading from a file.
2	Customer login screen	Omitted	I have decided not to include the screen for the customers login as it is identical to the staff login, therefore it isn't necessary for the prototype.
3	Add bookings	Included	I have included this as it is one of the main features of the system. This feature takes input from the user and can write the data to a database. I have decided not to include validation for this stage as my main goal is functionality of each screen.
4	View bookings	Included	I have included this part of the system to display the booking through a table. This is a key section as it allows the user to see all booking they need to see. This is done through reading the database file which holds the booking information, storing it as an array and using that data to display in a table.
5	Search bookings	Included	I have included the ability to search bookings by ID. In the prototype there will be a linear search. I have included this as a basic version of the final design for this screen, which will have various searches. (i.e. search by name, date)
6	Edit bookings	Omitted	I have not included the ability to edit bookings for the prototype of my system as right now it would require the system to search for bookings and replace them by overwriting a specific line within the database file. I feel that this process would be too complex at this stage especially
7	Menu	Included	I have decided to include a menu as it will keep the system simple and easy to navigate.
8	Payments	Omitted	I have decided not to include the payment function of my system. This is because it doesn't affect the main functionality of, they system which is to make bookings and view them. This would also make testing more difficult as the process would require calculations and therefore would need some form of validation in place, which I have decided not to do.
9	Payments table	Omitted	Again, I have decided not to include a table displaying payments as it is quite like the table displaying bookings. The process of reading and writing to the database file is identical therefore for the prototype I feel as though it is not necessary to repeat similar processes.
10	Price calculations	Omitted	I have also decided to exclude price calculations on my bookings as I will not be creating a functional payment feature.
11	Creating a confirmation email	Omitted	I have decided not to include the creation of the confirmation email, similarly to the payment calculations, this feature requires the payment feature to work so that all relevant information can be displayed in the email.

4.2 Screens and Outputs

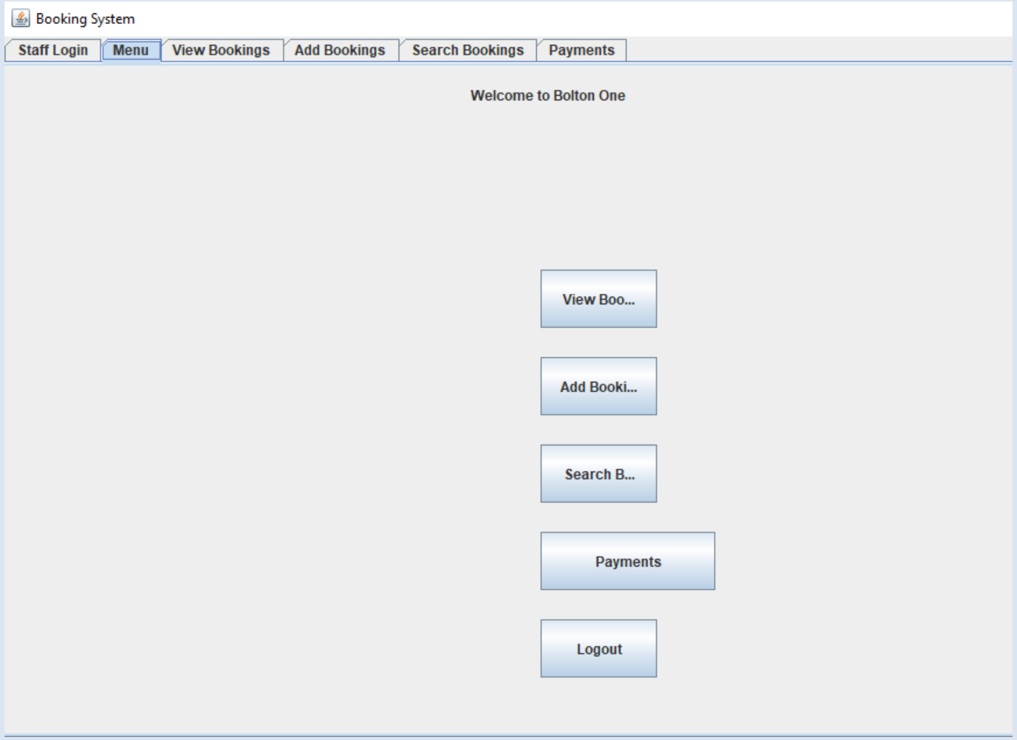
4.21 Screens (no Data)

Staff Login Screen	Features
	• Login button • Username text field • Password text field • Table displaying opening times

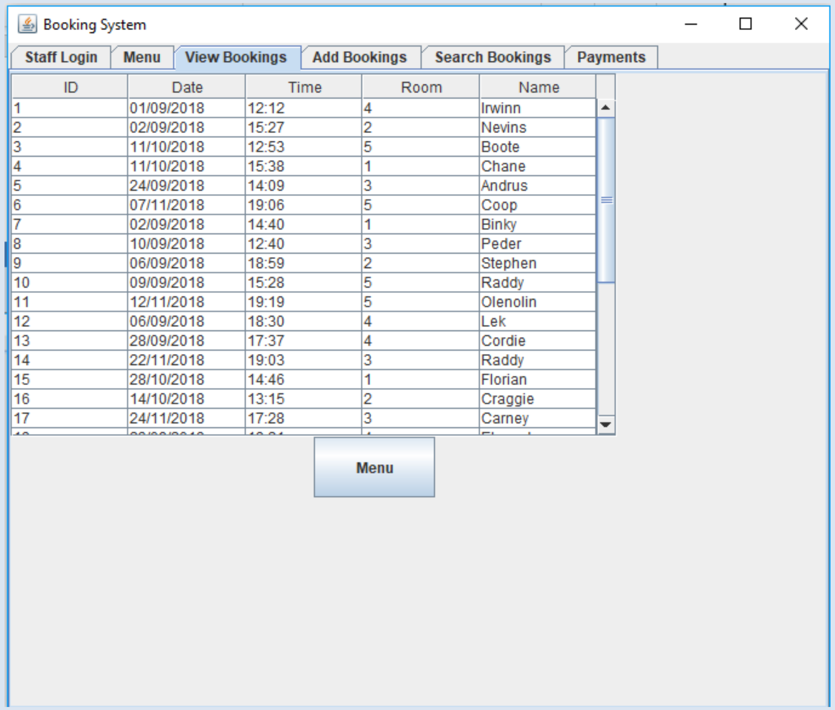
Days	Time
Monday	7am-9pm
Tuesday	7am-8pm
Wednesday	7am-8pm
Thursday	7am-8pm
Friday	8am-10pm
Saturday	7am-10pm
Sunday	9am-6pm

****

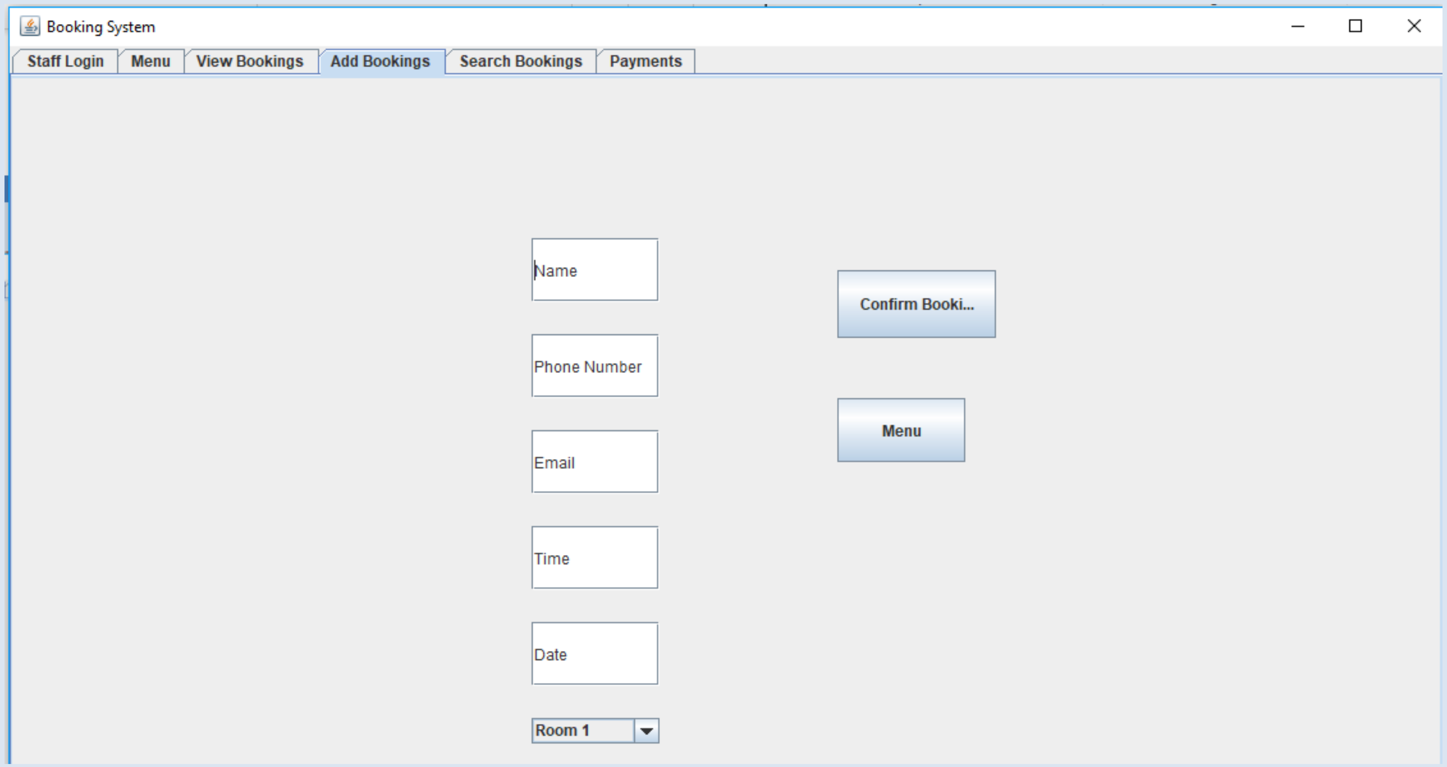
Username

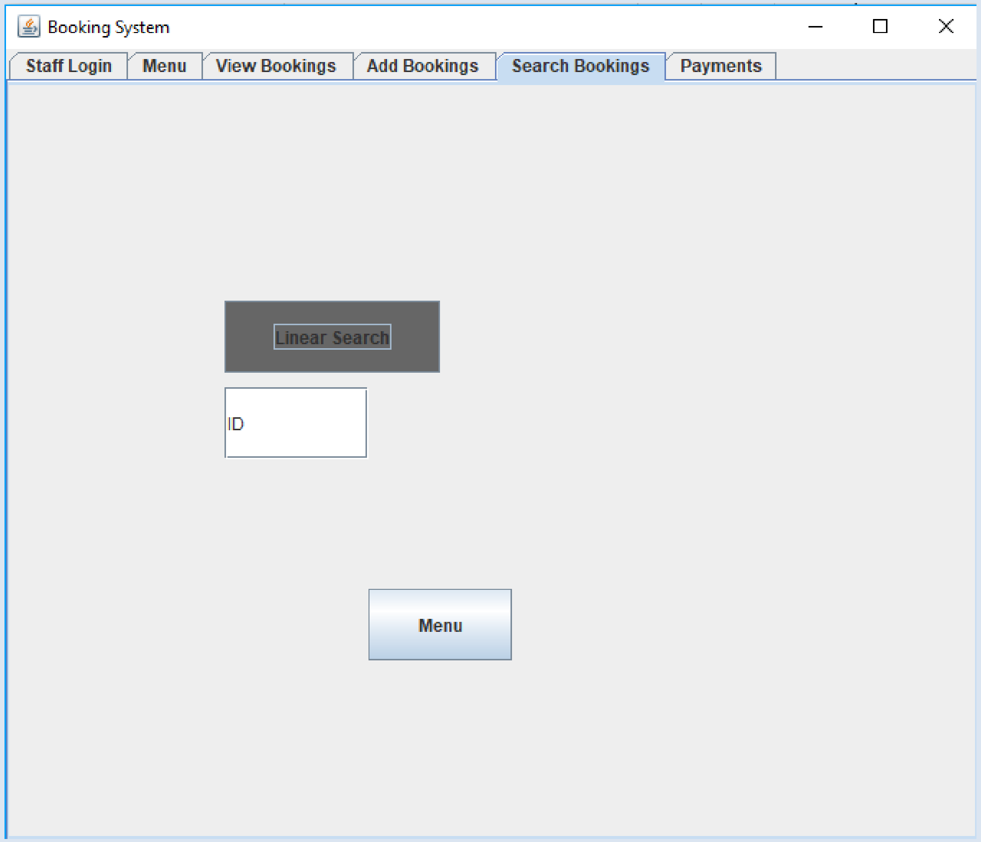
Menu Screen	Features
	• Label displaying welcome message. • Button – to access view booking page. • Button – to access add bookings page. • Button – to access search bookings page. • Button to logout.

Welcome to Bolton One

Admin Screen	Features
	• Table displaying the bookings • Button to return to menu

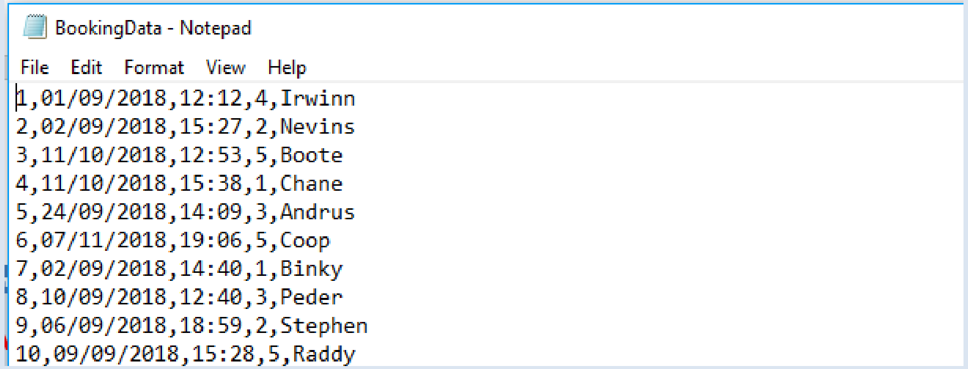
ID	Date	Time	Room	Name
1	01/09/2018	12:12	4	Irwinn
2	02/09/2018	15:27	2	Nevins
3	11/10/2018	12:53	5	Boote
4	11/10/2018	15:38	1	Chane
5	24/09/2018	14:09	3	Andrus
6	07/11/2018	19:06	5	Coop
7	02/09/2018	14:40	1	Binky
8	10/09/2018	12:40	3	Peder
9	06/09/2018	18:59	2	Stephen
10	09/09/2018	15:28	5	Raddy
11	12/11/2018	19:19	5	Olenolin
12	06/09/2018	18:30	4	Lek
13	28/09/2018	17:37	4	Cordie
14	22/11/2018	19:03	3	Raddy
15	28/10/2018	14:46	1	Florian
16	14/10/2018	13:15	2	Craggie
17	24/11/2018	17:28	3	Carney

Add Bookings Screen	Features
 Booking System Staff Login Menu View Bookings Add Bookings Search Bookings Payments Name Phone Number Email Time Date Room 1 Confirm Booki... Menu	• Textfield – name • Textfield – Phone number • Textfield – Email • Textfield – Time • Textfield - Date • Combo box – Room • Button - Confirm Booking • Button – Return to menu

Search Bookings Screen	Features
	- Button – for linear search - Text field – input for searches - Button – menu

4.22 Outputs

Booking File

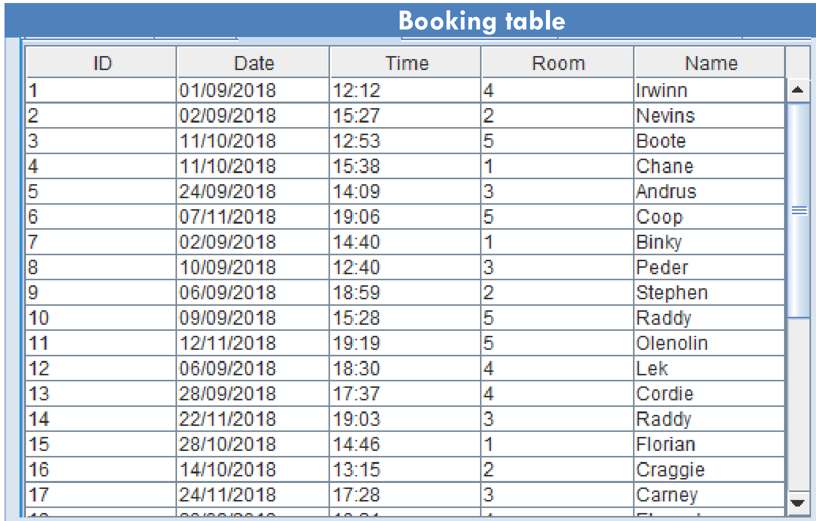


```
1,01/09/2018,12:12,4,Irwinn
2,02/09/2018,15:27,2,Nevins
3,11/10/2018,12:53,5,Boote
4,11/10/2018,15:38,1,Chane
5,24/09/2018,14:09,3,Andrus
6,07/11/2018,19:06,5,Coop
7,02/09/2018,14:40,1,Binky
8,10/09/2018,12:40,3,Peder
9,06/09/2018,18:59,2,Stephen
10,09/09/2018,15:28,5,Raddy
```

Data & Purpose

Here is a screenshot of the database file which holds the data for each booking. Each line holds the Booking ID, Date, Time and Name associated with the booking.

Booking table

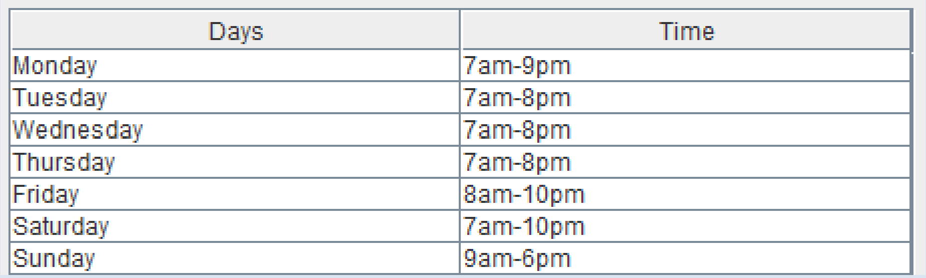


ID	Date	Time	Room	Name
1	01/09/2018	12:12	4	Irwinn
2	02/09/2018	15:27	2	Nevins
3	11/10/2018	12:53	5	Boote
4	11/10/2018	15:38	1	Chane
5	24/09/2018	14:09	3	Andrus
6	07/11/2018	19:06	5	Coop
7	02/09/2018	14:40	1	Binky
8	10/09/2018	12:40	3	Peder
9	06/09/2018	18:59	2	Stephen
10	09/09/2018	15:28	5	Raddy
11	12/11/2018	19:19	5	Olenolin
12	06/09/2018	18:30	4	Lek
13	28/09/2018	17:37	4	Cordie
14	22/11/2018	19:03	3	Raddy
15	28/10/2018	14:46	1	Florian
16	14/10/2018	13:15	2	Craggie
17	24/11/2018	17:28	3	Carney

Data & Purpose

This is the table displaying the data for the all the bookings. The data is read from a database file which is used to display the data under the correct headings.

This is for the user to view all the bookings on the system.

<table>
<tr><th>Opening times table</th><th>Data &amp; Purpose</th></tr>
<tr><td>

<table>
<tr><th>Days</th><th>Time</th></tr>
<tr><td>Monday</td><td>7am-9pm</td></tr>
<tr><td>Tuesday</td><td>7am-8pm</td></tr>
<tr><td>Wednesday</td><td>7am-8pm</td></tr>
<tr><td>Thursday</td><td>7am-8pm</td></tr>
<tr><td>Friday</td><td>8am-10pm</td></tr>
<tr><td>Saturday</td><td>7am-10pm</td></tr>
<tr><td>Sunday</td><td>9am-6pm</td></tr>
</table>
</td><td>This is a table used to display the openings times of the business which is shown on the login screen for users to see.</td></tr>
</table>

4.3 Functioning System (with Data)

4.31 Processes to be Evidenced

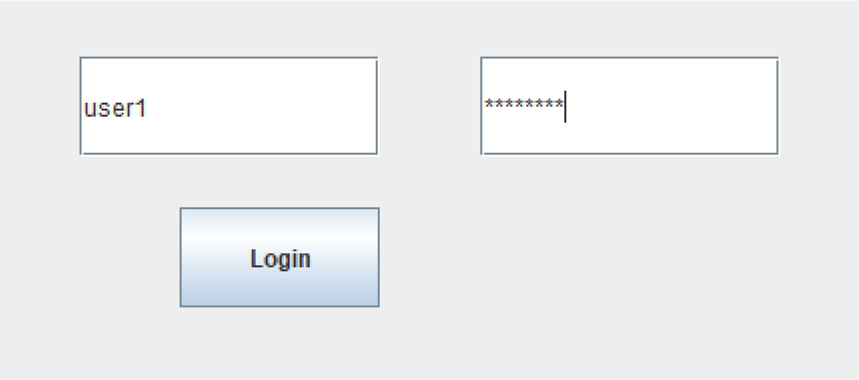
Process	Description	Realistic Data Used	Data Output and Storage Requirements
1	Login screen – The user must enter their username and password into the given fields. The input is then checked against the list of logins within an array to see if they match. The user will only be allowed access if both username and password are correct and of the same record. There is also a table to display the opening times of the business, which is created using data stored in a database.	Record Example: user1, password123 Monday,7am-9pm	Output to the command line and switches to different page. The data for the staff logins are stored in a text file with both username and password also being read and stored into an array of login data The data for the table is stored in a database text file, which is read from once the system is started up. Each record within the file is stored into an array of opening data which is used to display the data within the table.
2	Menu Screen – there is multiple buttons used to navigate to different pages within the system.	N/A	Output to command line and pages are changed. Storage N/A
3	View Bookings Screen – this screen shows the table which holds bookings along with all relevant information using data read from the database. There is also a menu button to navigate to the menu again.	Record Example: 1,01/09/2018,12:12,4,Irwinn	output to command line, and the table with the data is outputted to the user from the file. The data in the table is read from a sperate database file which holds the data about the bookings. Once this data is read from the database file, it is stored into an array of booking data, which is used to

			display it with in a table. The data is stored in the array correlating to the correct variable, allowing the data to be in the correct order in the table.
4	Add Bookings Screen - there are multiple text fields and a combo box used for the user to input data into the system. Once input is finished the user will press the confirm booking button to add the booking. When the button is pressed, the data is taken from each input and combined into a string, and added to a list, which is written to the database file holding booking data.	Record example: Shak Shah 01204 555555 Shak@gmail.com 19:00-20:00 12/11/18 Room 2	Output to the command line, and the page is moved to the payments page The data entered by the user is added to the booking array as a record. This is then written to the database holding the booking data.
5	Search Bookings Screen – on this screen the user can input the ID of the booking they are searching for and it will perform a linear search to find the location of the booking.	ID example: 5	Output to the command line, also a pop up is displayed to inform the user on the location of the booking. There is no storage of data for this screen, instead it will access data from the booking database.
6	Payment Screen - There are two text fields which take input from the user on their card details for them to pay. As this is my prototype when data is entered, and the button pressed nothing will happen as I decide not to include the functionality of the payments in this stage of development.	Card number/holder example: 0165 4598 4678 4625 Shak Shah	Output to command line. The data will be stored into a database file holding the records for payment information. The data will be stored into an array of payment data before being written to the booking file.

4.32 Evidence

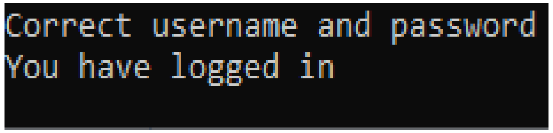
Process	1 - Login Screen

Evidence

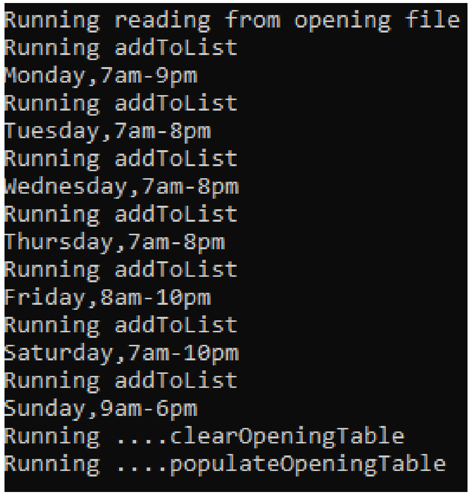


user1

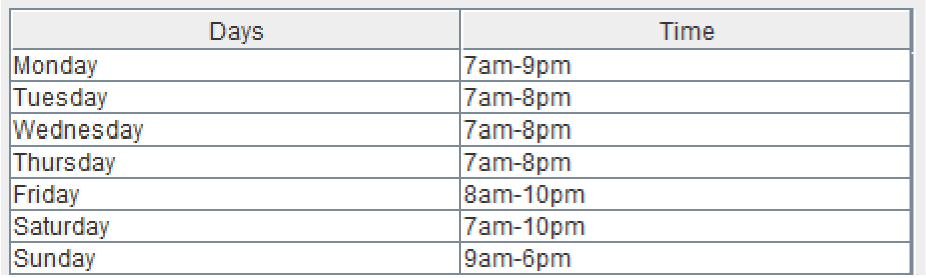
****



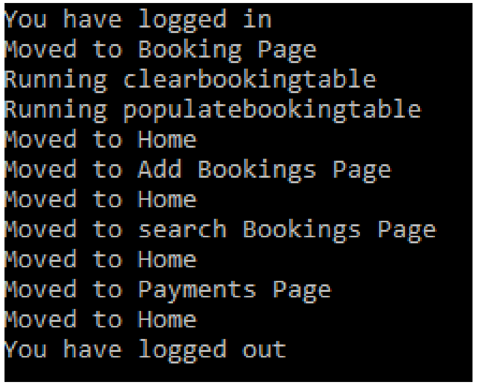
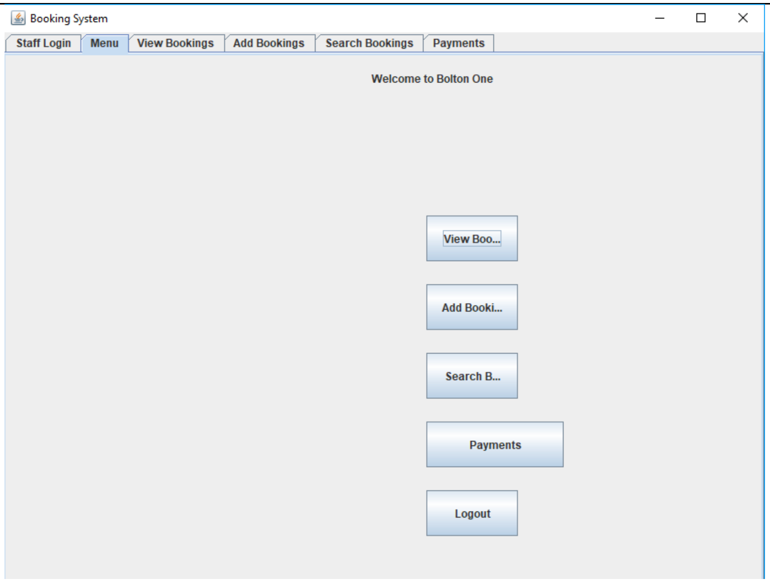
```
Correct username and password
You have logged in
```



```
Running reading from opening file
Running addToList
Monday,7am-9pm
Running addToList
Tuesday,7am-8pm
Running addToList
Wednesday,7am-8pm
Running addToList
Thursday,7am-8pm
Running addToList
Friday,8am-10pm
Running addToList
Saturday,7am-10pm
Running addToList
Sunday,9am-6pm
Running ....clearOpeningTable
Running ....populateOpeningTable
```

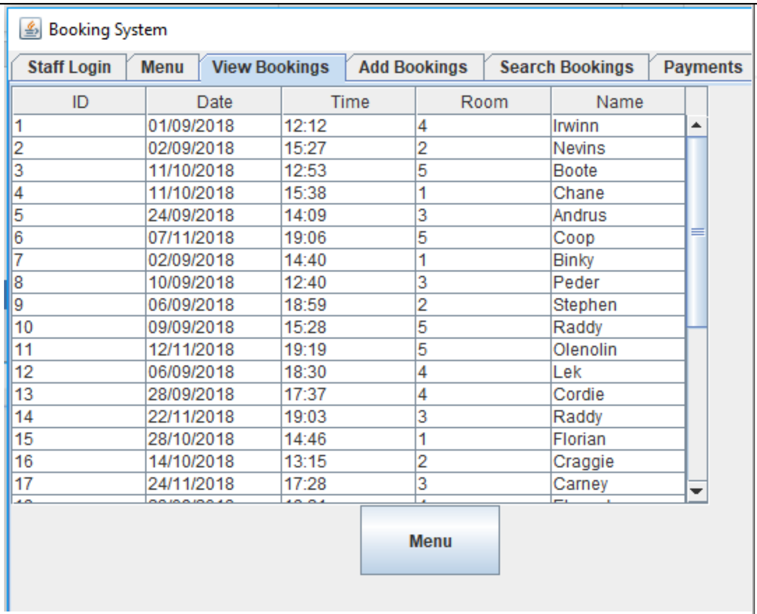


Days	Time
Monday	7am-9pm
Tuesday	7am-8pm
Wednesday	7am-8pm
Thursday	7am-8pm
Friday	8am-10pm
Saturday	7am-10pm
Sunday	9am-6pm

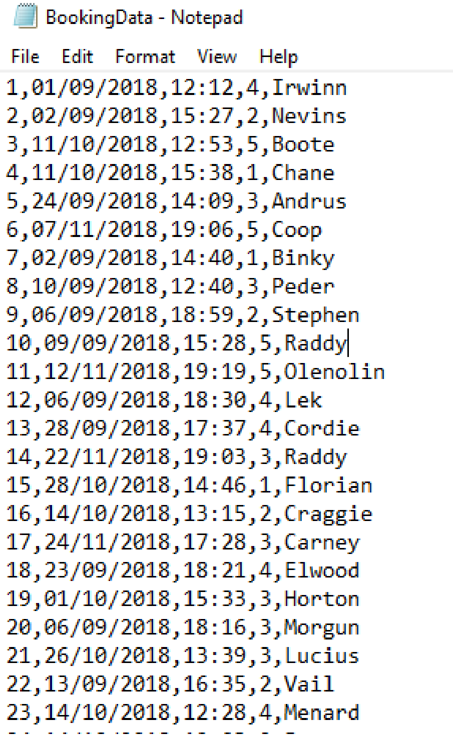
Process	2 - Menu Screen
Evidence	 

```
You have logged in
Moved to Booking Page
Running clearbookingtable
Running populatebookingtable
Moved to Home
Moved to Add Bookings Page
Moved to Home
Moved to search Bookings Page
Moved to Home
Moved to Payments Page
Moved to Home
You have logged out
```

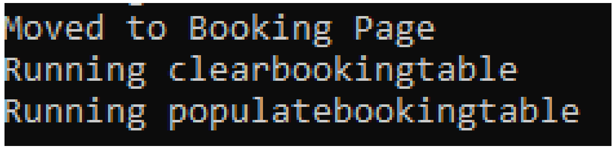
Process	3 – View Bookings Screen
Evidence	



ID	Date	Time	Room	Name
1	01/09/2018	12:12	4	Irwinn
2	02/09/2018	15:27	2	Nevins
3	11/10/2018	12:53	5	Boote
4	11/10/2018	15:38	1	Chane
5	24/09/2018	14:09	3	Andrus
6	07/11/2018	19:06	5	Coop
7	02/09/2018	14:40	1	Binky
8	10/09/2018	12:40	3	Peder
9	06/09/2018	18:59	2	Stephen
10	09/09/2018	15:28	5	Raddy
11	12/11/2018	19:19	5	Olenolin
12	06/09/2018	18:30	4	Lek
13	28/09/2018	17:37	4	Cordie
14	22/11/2018	19:03	3	Raddy
15	28/10/2018	14:46	1	Florian
16	14/10/2018	13:15	2	Craggie
17	24/11/2018	17:28	3	Carney

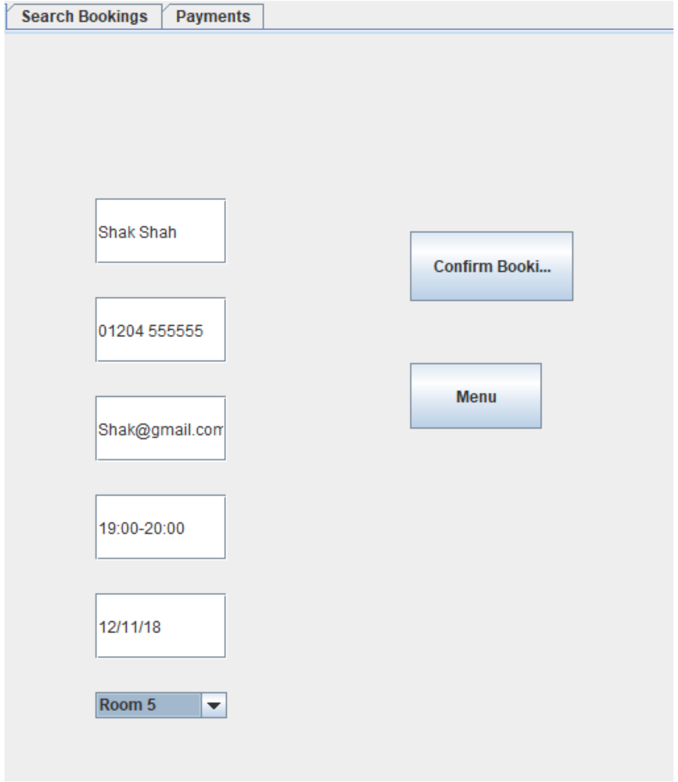
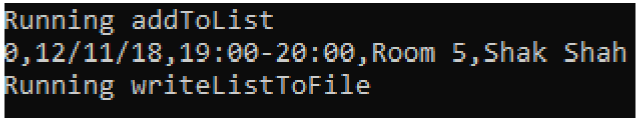


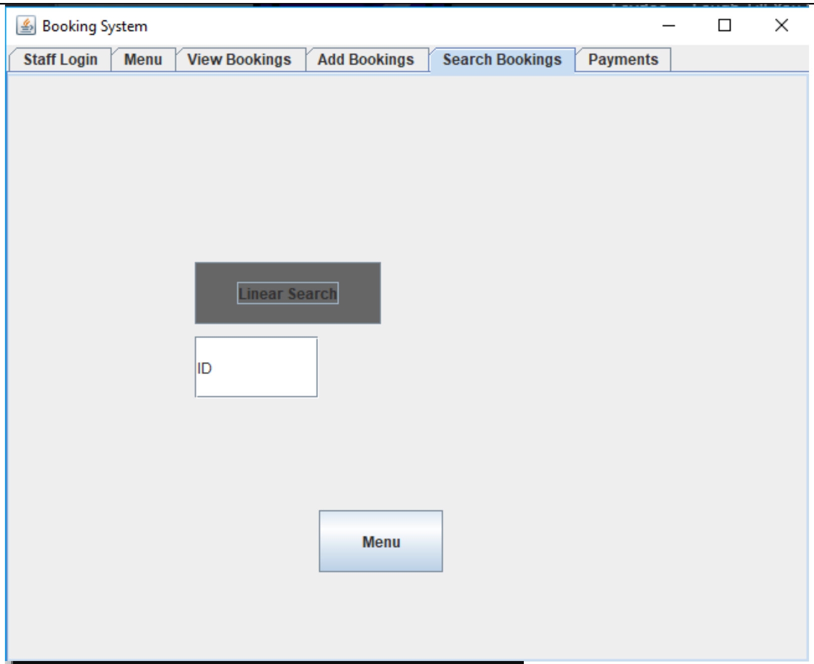
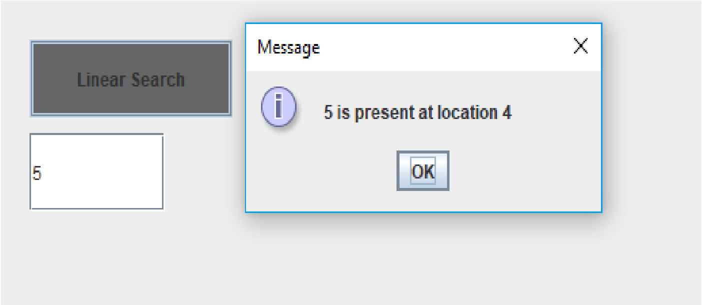
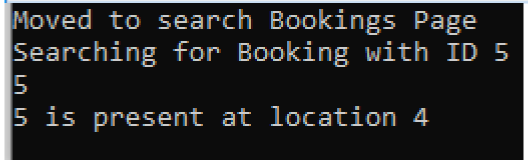
```
1,01/09/2018,12:12,4,Irwinn
2,02/09/2018,15:27,2,Nevins
3,11/10/2018,12:53,5,Boote
4,11/10/2018,15:38,1,Chane
5,24/09/2018,14:09,3,Andrus
6,07/11/2018,19:06,5,Coop
7,02/09/2018,14:40,1,Binky
8,10/09/2018,12:40,3,Peder
9,06/09/2018,18:59,2,Stephen
10,09/09/2018,15:28,5,Raddy
11,12/11/2018,19:19,5,Olenolin
12,06/09/2018,18:30,4,Lek
13,28/09/2018,17:37,4,Cordie
14,22/11/2018,19:03,3,Raddy
15,28/10/2018,14:46,1,Florian
16,14/10/2018,13:15,2,Craggie
17,24/11/2018,17:28,3,Carney
18,23/09/2018,18:21,4,Elwood
19,01/10/2018,15:33,3,Horton
20,06/09/2018,18:16,3,Morgun
21,26/10/2018,13:39,3,Lucius
22,13/09/2018,16:35,2,Vail
23,14/10/2018,12:28,4,Menard
```

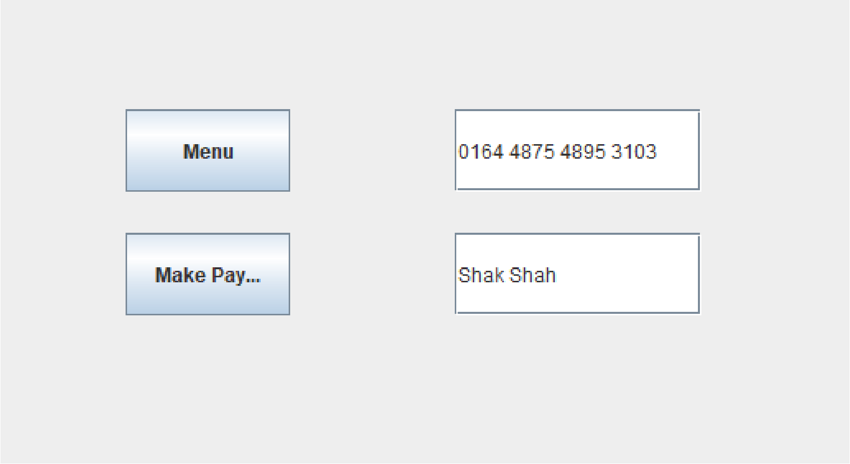
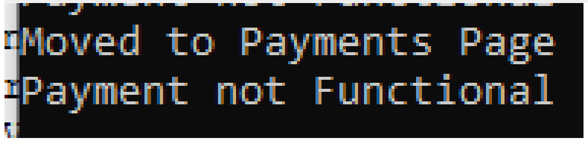


```
Moved to Booking Page
Running clearbookingtable
Running populatebookingtable
```

Process	4 – Add Bookings Screen

Evidence	
	Shak Shah
	01204 555555
	Shak@gmail.com
	19:00-20:00
	12/11/18
	Room 5
	
	`Running addToList`
	`0,12/11/18,19:00-20:00,Room 5,Shak Shah`
	`Running writeListToFile`

Process	5 - Search Bookings Screen
Evidence	 Booking System Staff Login Menu View Bookings Add Bookings Search Bookings Payments Linear Search ID Menu  5 is present at location 4  Moved to search Bookings Page Searching for Booking with ID 5 5 5 is present at location 4

Process	6 – Payments Screen
Evidence	 0164 4875 4895 3103 Shak Shah  Moved to Payments Page Payment not Functional

4.4 Self-Evaluation

4.41 Functioning of the Prototype

Overall my prototype meets the minimum requirements for the functionality of my system at this stage. The system currently can add bookings, display them in a table, and allows one search filter. The screens are quite simple and basic, but easy to navigate for users. Each screen has button which are clear on their purpose, allowing users to understand what each screen does. There are some features that will be implemented for the final development of the system which have not been included as this is the prototype, to show the basic functionality of the system. The current implementations will also be refined for that stage in the development process.

4.42 Good Features

Feature	Description
Tables to display data	My prototype displays two separate tables one **displaying the opening times** for the business and the other to **display the all bookings of** the system to the user. This feature **provides a simple an effective method of displaying** useful information to the user and **can be updated while the system is in use** i.e. if a booking is added the table would update on the view bookings page.
Writing to the text file to store data	The system **can effectively add new booking data to the to record of booking** data stored into an array, which can easily be written to the text file.
Reading from text file	the system can **effectively read from the text file** by reading each line of the data and splitting the line to produce the record of booking data. **This process is effective as the data can be read once** when the system is first run and stored into an array **to be accessed efficiently many times throughout** the use of the system.
Searches	The **linear search** function is available to allow the user to find a certain booking. This is effective as it allows the user to search for booking, they want to find using a **simple and easy method** where all they do is search for the ID of the booking, which **is user friendly.**
Staff Login	The user can enter **their username and password** to gain access to the system. The data entered is **compared with pre-stored login** data which will be **accepted if both attributes** of the login record match.

4.43 Shortcomings

Feature	Description
Efficiency of searching	The there is **a large amount of booking records,** meaning that a **linear search may not be the most effective** way to search for bookings. This is because each index of the array would need to be cycled through and compared to the search value which will **subsequently slow down the system.** As the system will be used by **multiple users simultaneously** this will have a large strain on the system, potentially **causing crashes or slowdowns.**
Efficiently of general use	Although the system is quite straightforward to use, little things such as having **to remove text from text fields** before entering data may **slow down the use of the system,** making the system **less intuitive and inefficient.** Also due to this the **screens seem empty,** with a lot of wasted space as well as lack **of titles/headings of screens** which do not effectively illustrate the purpose of each screen.
Sorts	There are **no sorts** in the prototype which would have shown the **basic functionality** of the proposed system to be built on later in development. This prevents the bookings to be filtered on the table **limiting the use of the view bookings screen and its effectiveness.**
Lack of output to users	There is **little output on the system** such as when the user adds a booking and clicks the confirm button, although the booking is added, there is **no feedback to confirm** to the user that it has happened. This can make the system feel **unresponsive and may confuse users,** both novice and familiar, making the system **not user friendly enough** to be used effectively and may cause repeat bookings as the user may try to create a booking using the same data which could cause it to crash, but also would create **redundant data in the database file.**
Add bookings page – input fields	On the add bookings page the fields for selecting the date and time for the booking **requires the user to enter a time rather than select a free slot.** This could cause the user to choose a time or date which has a booking in that slot, **increasing the likelihood of incorrect booking** being made. Also, **it's unclear on the format** in which the user should enter the data as there is **no example for** the user to base their input off, which could **cause further confusion.**

4.44 Suggestions for Improvements

Feature	Description
Efficiency of searching	The system could've been more efficient **in including a different search, such as a binary search** to improve the search of the bookings, as there is a large amount of records to search from. This would be a suitable idea to be implemented into the final system, post prototype, as it **would decrease the number of comparisons** per search which **would improve efficiency massively** as it would allow searches to be conducted much quicker.
Efficiently of system use	This could easily be solved by allowing the **use of larger text fields** which would allow for all inputs to fit within the text field, but also **include labels to clearly illustrate** the use of each feature on screen. The use of headings for each screen would also allow novice users to understand the look and feel of the system easier **making it more user friendly.**
Sorts	**A simple bubble sort** could've been introduced to allow for the filtration of the bookings on the view bookings page to **show the full purpose** of the screen and show the feature even if not fully refined.
Lack of output to users	This could be improved by allowing **a form of feedback such as a pop up** to confirm actions to the user, which would make **the human-computer interaction smoother**, allowing users to effectively navigate and use the system.
Add bookings page – input fields	To prevent incorrect input from the user, combo boxes could be used for date and time to allow the user to select a free slot but also to see unavailable slots, to prevent incorrect bookings being made onto the system but also so that the user isn't misled on whether they have successfully added their choice of booking.