Binary Beasts
Elaboration Document

Online Restaurant Booking System

Team Members:

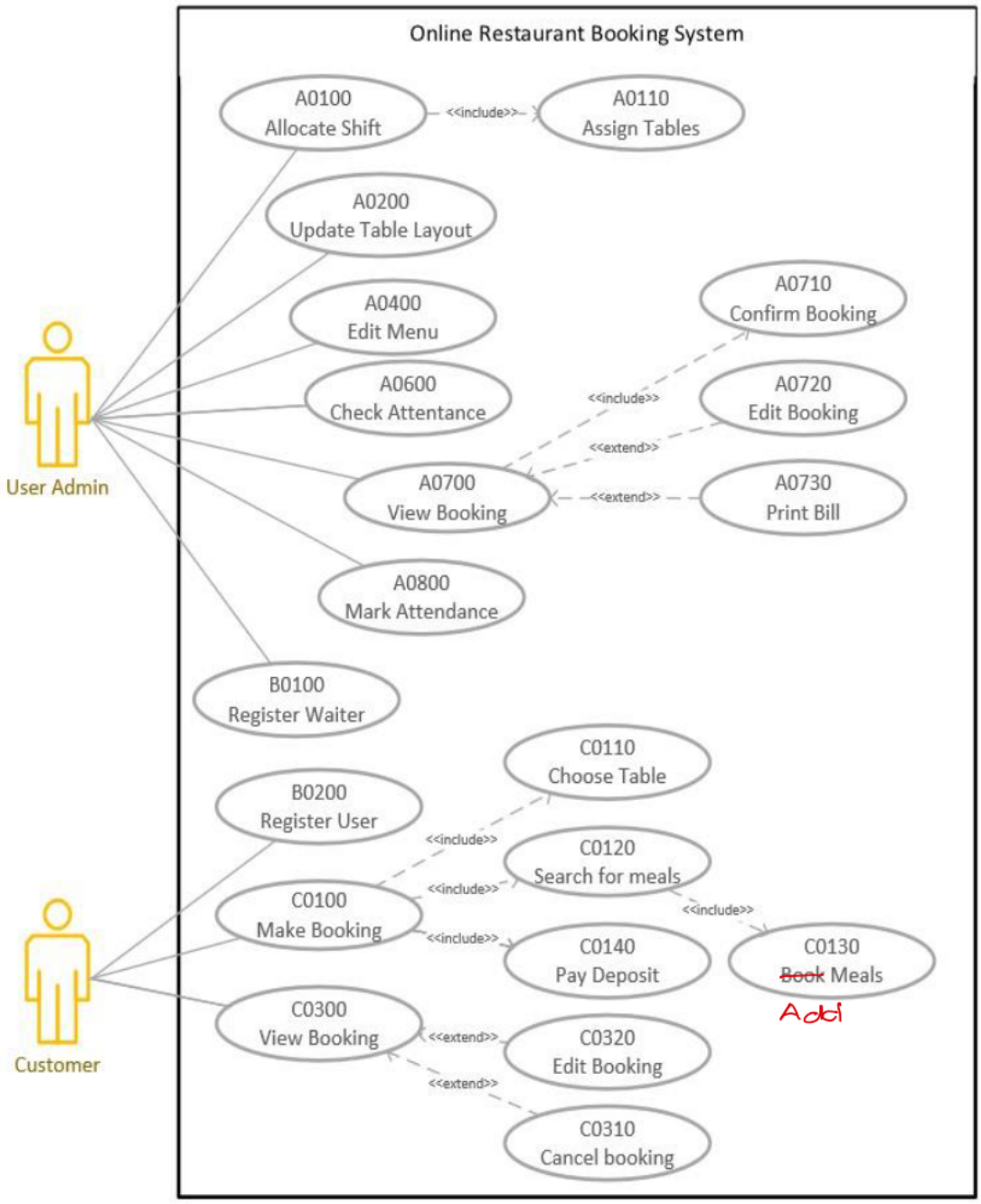
[redacted] 60%
[redacted] 60%
[redacted] 60%

Document Date: 20/05/2022

TABLE OF CONTENTS

1 FUNCTIONAL REQUIREMENTS

1.1 Analysis Use Case Model



1.2 Use Case Glossary and Responsibilities

Team Member Responsible: [redacted]	
Use Case Id	**Use Case Name**
A0100	Allocate Shift
A0110	Assign Tables
A0400	Edit Menu
C0140	Pay deposit
Queries/Reports	
A0200	Update Table Layout
A0800	Mark Attendance
C0100	Make booking

Team Member Responsible: [redacted]	
Use Case Id	**Use Case Name**
B0200	Register User
C0110	Choose table
C0310	Cancel booking Add
C0130	~~Book~~ meal
C0320	Edit booking
Queries/Reports	
C0120	Search for meals
C0300	View booking

Team Member Responsible: [redacted]	
Use Case Id	**Use Case Name**
B0100	Register waiter
A0700	Confirm booking
A0730	Print bill
Queries/Reports	
A0600	Check attendance
A0720	Edit booking
A0710	View booking

2 UI Prototypes

2.1 Team UI Guidelines

User Admin Application :

Once a user is logged on to their account the dashboard will appear. From here the user can navigate the application from the navigation pane on the left of the screen. A user can switch between tabs by using the buttons placed on the navigation pane. Once a user selects a choice on the navigation pane they can always return to the dashboard by clicking on the cancel button. The dashboard will show the latest bookings as those are the most important information to the restaurant. A user can also log out at any time by clicking on the log out button. This navigation scheme ensures that the system is easy to use and makes for an enjoyable user experience.

For this system the colour scheme we chose is white, grey and blue. We will primarily use grey and blue as background colours. We use a combination of black and white as text. We use whatever colour is easier to read and to draw attention to critical ui elements. For the font we will use Open Sans. This font is easy to read by the user and makes navigation easier.

For user input we use a combination of text fields and selection boxes(to choose the tables for a waiter). To upload an image the user can simply drag the new image to the required input field. This will make using the system relatively easy.

Errors will be handled in the code. A user will not be able to complete an action if the details are not filled in correctly. An error message will be displayed if the input is invalid.

This system will focus on functionality over aesthetics. It will mainly be used by the staff and does not need to look as good for the customer. The system working reliably will be the main priority.

Website :

The design and layout of the website is simple to allow for usability. In addition to this, only information that is relevant and necessary is shown to avoid clutter. The colours used for the designs of the interfaces are black, blue, white and grey throughout the website.

Consistency is important therefore all the website's interfaces currently use the font Arial and font colour is white for all navigation menus. The navigation menu will always be at the top of every page and the current open page will be highlighted with the heading page displayed at the top. A customer will be able to navigate between pages by using the navigation menu, by cancelling an action or by completing an action.

Pop up messages will be used to confirm if the customer would like to continue with an action or to notify the customer that a particular action has been completed.

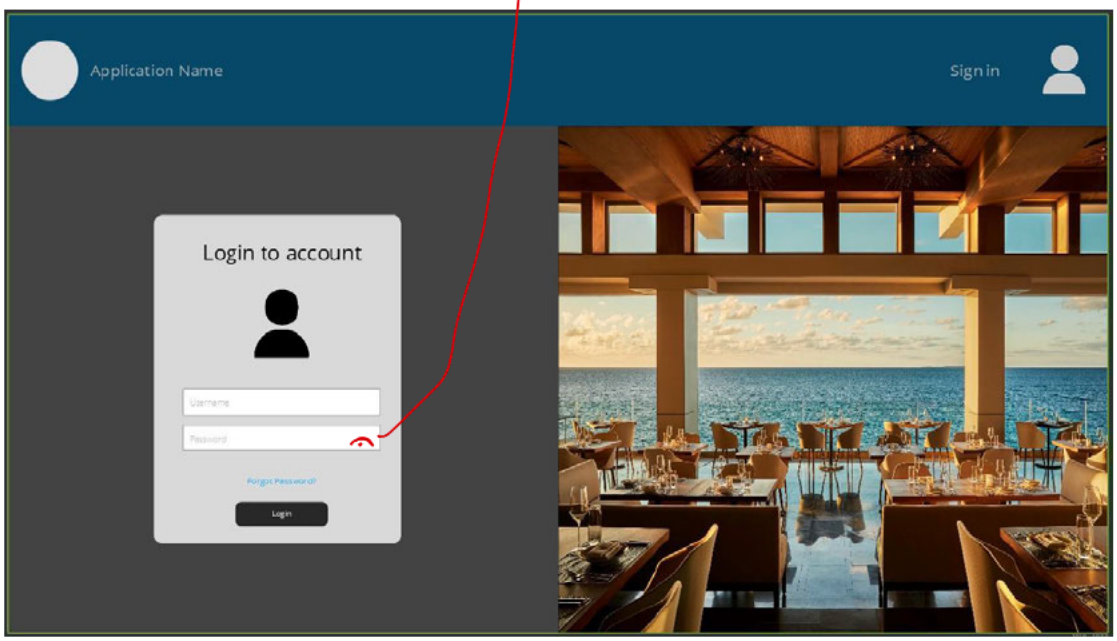
All buttons on the main interfaces are blue and are clickable. Some buttons have links that take the user to a different page, for example the "our story" button on the dashboard takes the customer to the about page. The "make booking" button takes the customer to the pay deposit page, the "edit booking" takes the customer to the edit booking form page.

In addition to using easy and straightforward words to describe page names and buttons, a drop down menu is also used when searching for meals; this helps make the selection of the meal easier for the customer.

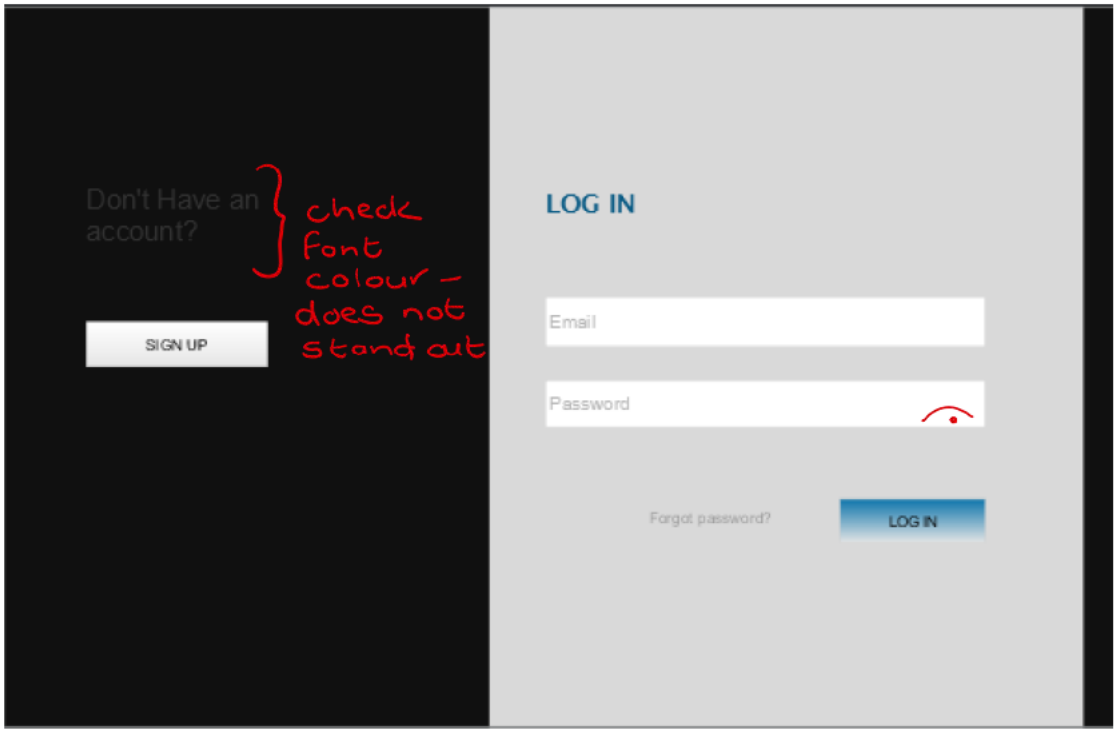
→ make the buttons the same as on the Admin App (rounded corners, color choice, button order)

2.2 Team Design for

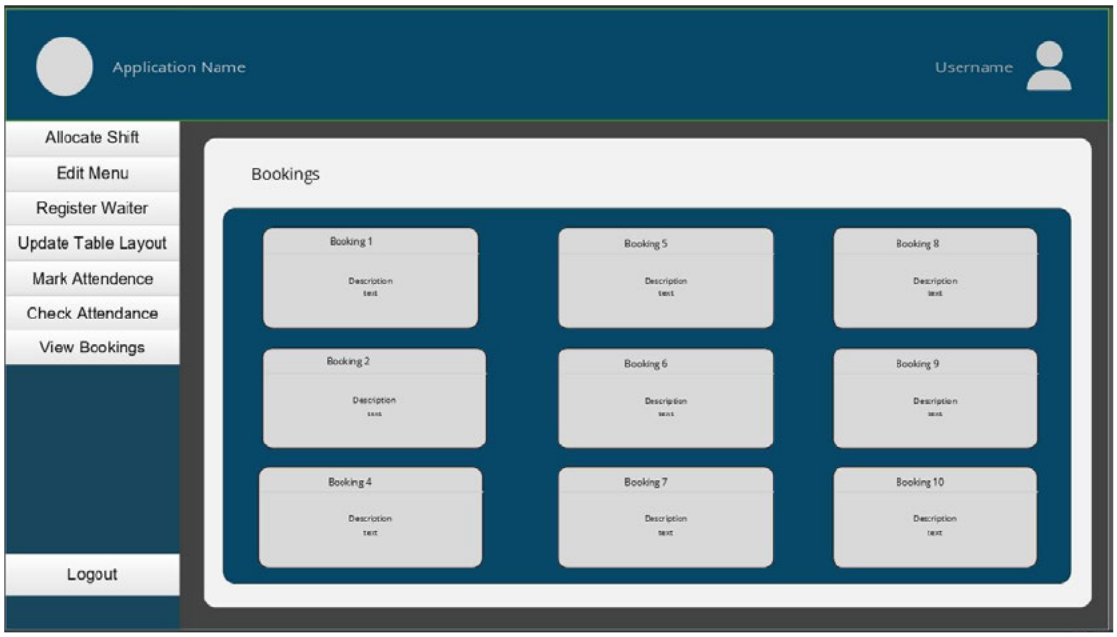
2.2.1 Login / Logout



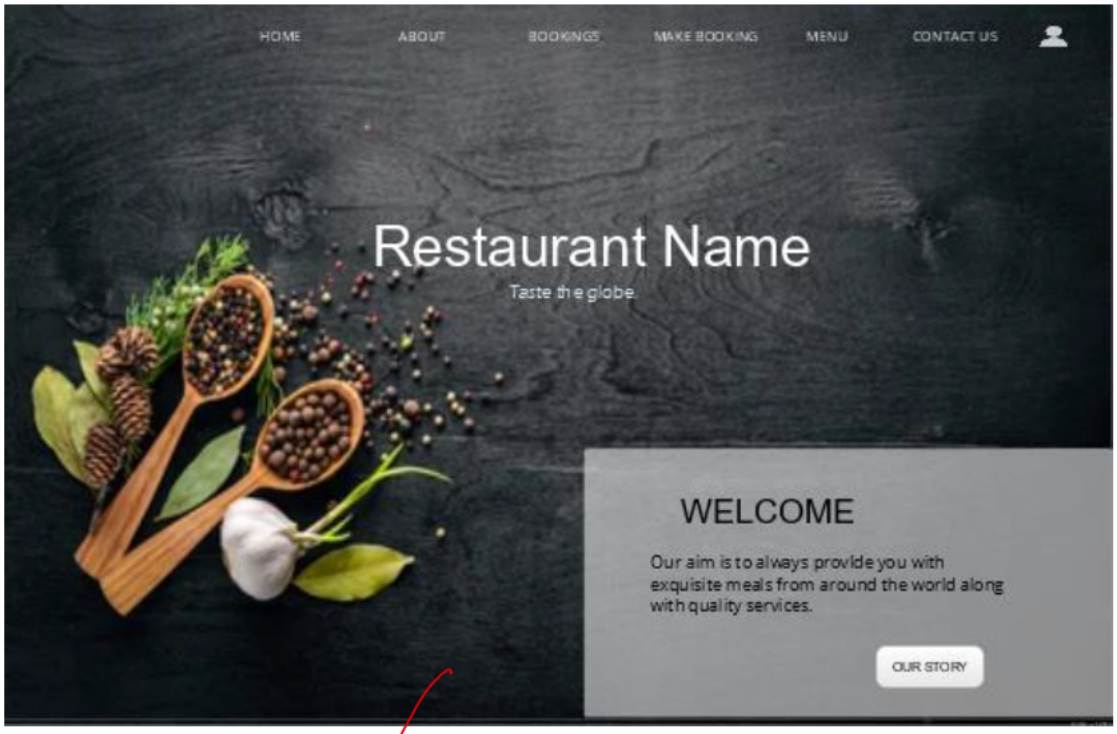
2.2.2 Login Website



2.2.3 Dashboard / Landing Page / Main Page



2.2.4 Website dashboard / Landing Page / Main Page



make background image a bit lighter so menu can stand out more OR have a grey long menu bar with options in black

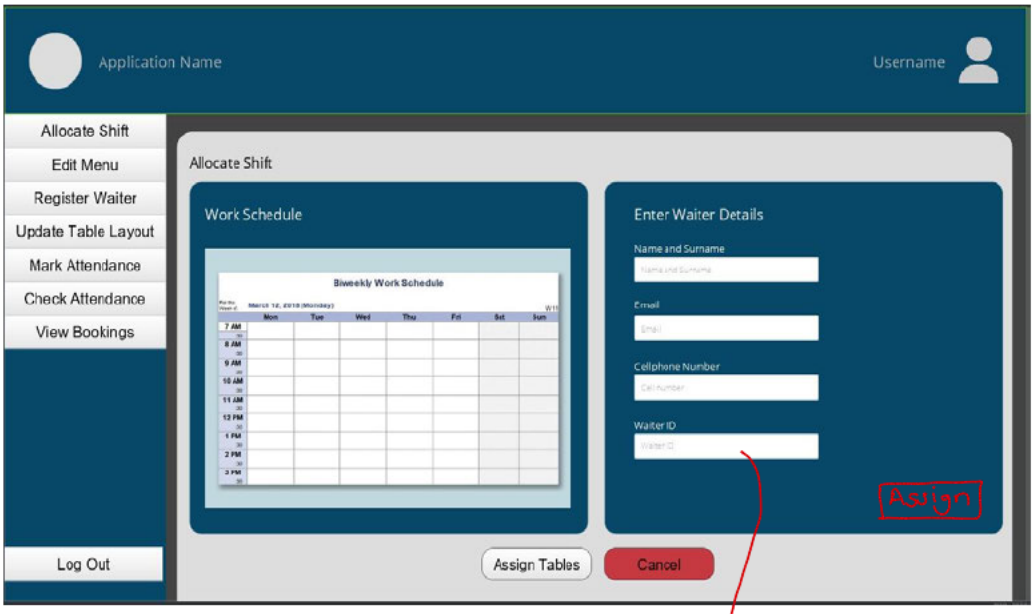
* Use a lighter red for Cancel button or make text white (it is a bit unclear)

△ There are alot of narratives without alternate flows. (i.e. what happens when user selects CANCEL)

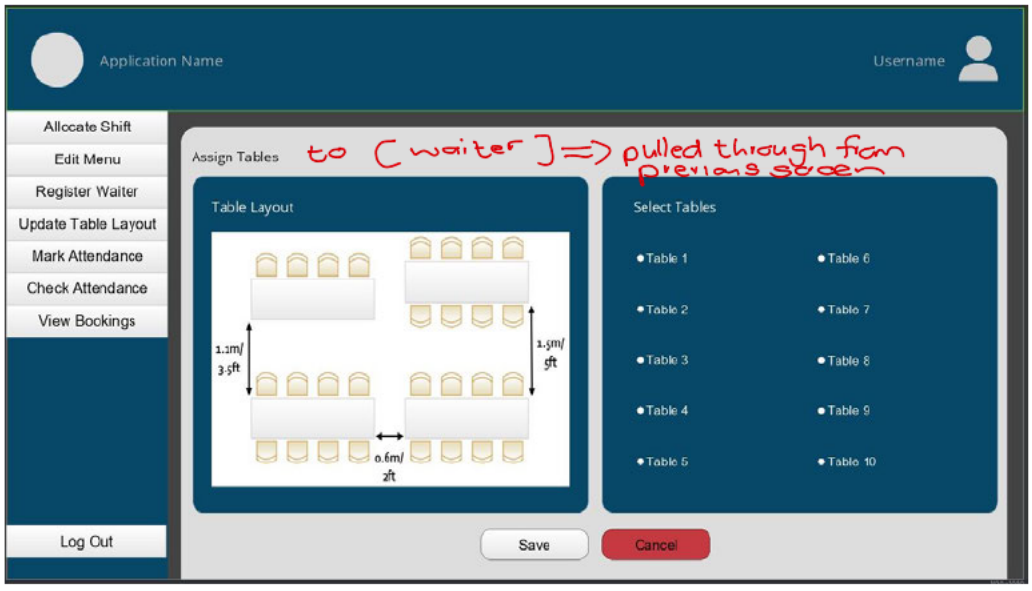
2.3 UI Designs & Updated Analysis Use Case Narratives

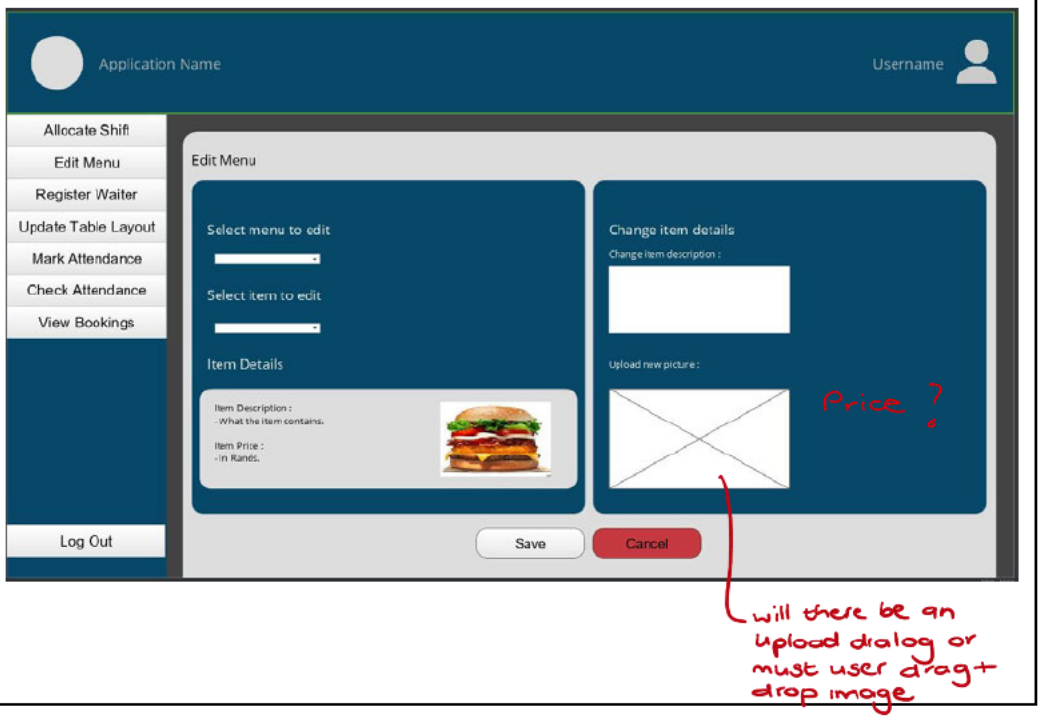
2.3.1 Designed by [redacted]

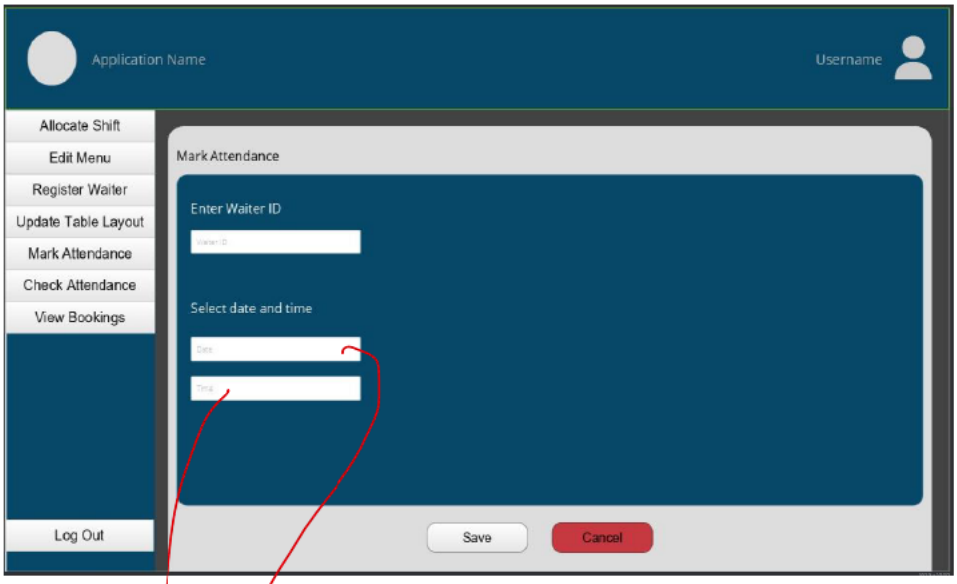
Use Case ID	Use Case Name
A0100	Allocate Shift
Primary Business Actors	**Other participating Actors**
User Admin	Shift
Description	This use case describes the process of allocating a use case to a waiter.
Pre-Conditions	Waiter must be a hired staff member in a restaurant.
Triggers	Select allocate shift from Admin Dashboard.
Post-Conditions	A waiter is allocated a shift in a restaurant work schedule.
Basic Flow of Events	1. System displays the work schedule. , would you need to select shift slot or >1 slot ? 2. Admin enters waiter details. 3. Allocate the waiter for a shift in the work schedule. 4. Call A0110 to assign tables.
Initial UI design	△ can waiter not be selected from list ?



auto generated ?

Use Case ID	Use Case Name
A0110	Assign Tables
Primary Business Actors	
User Admin	
Description	Assign tables to a waiter.
Pre-Conditions	Waiter has been allocated a shift.
Triggers	~~User Admin clicks on Assign Tables~~. called from A100
Post-Conditions	Waiter is assigned to tables.
Basic Flow of Events	1. ~~User admin clicks on Assign tables~~. 2. Table layout is displayed. 3. User admin selects the table numbers of the waiter to assign it.
Initial UI design	

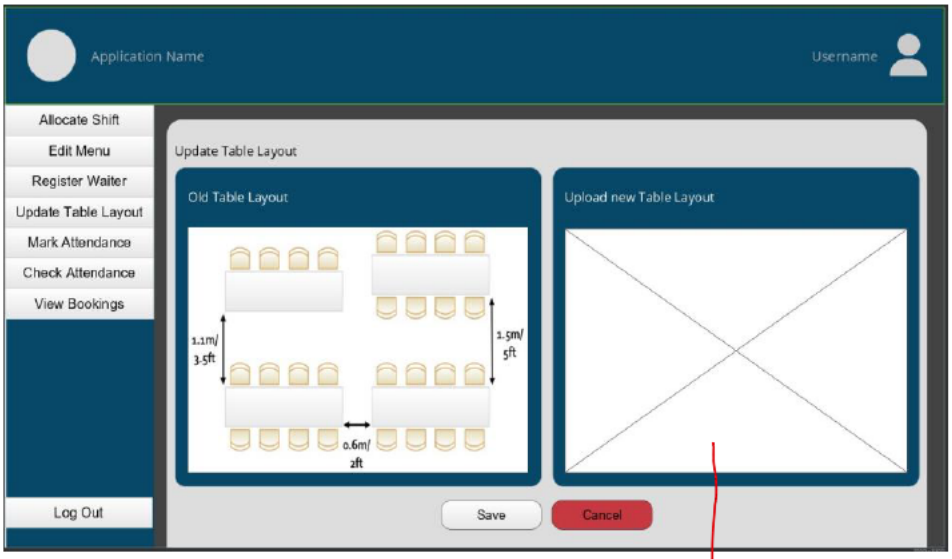
Use Case ID	Use Case Name	
A0400	Edit Menu	
Primary Business Actors		
User Admin		
Description	User Andmin can update the price or add or remove an item on the menu.	
Pre-Conditions	User admin is logged onto the system and a menu is added.	
Triggers	User admin clicks on Edit Menu.	
Post-Conditions	Menu is updated to reflect new changes.	
Basic Flow of Events	1. Select menu to edit. 2. User admin makes changes to the menu. 3. Menu is updated on the website to reflect changes	
Initial UI design		

Use Case ID	Use Case Name
A0800	Mark attendance
Primary Business Actors	
User Admin	
Description	User admin marks waiter attendance at the restaurant.
Pre-Conditions	Waiter has been marked present for their work shift.
Triggers	Select mark attendance from the dashboard.
Post-Conditions	The waiter has been marked present for their shift and is now able to work.
Basic Flow of Events	1. User admin clicks on mark attendance. 2. User admin enters waiterID. 3. User Admin selects the date and time.
Initial UI design	

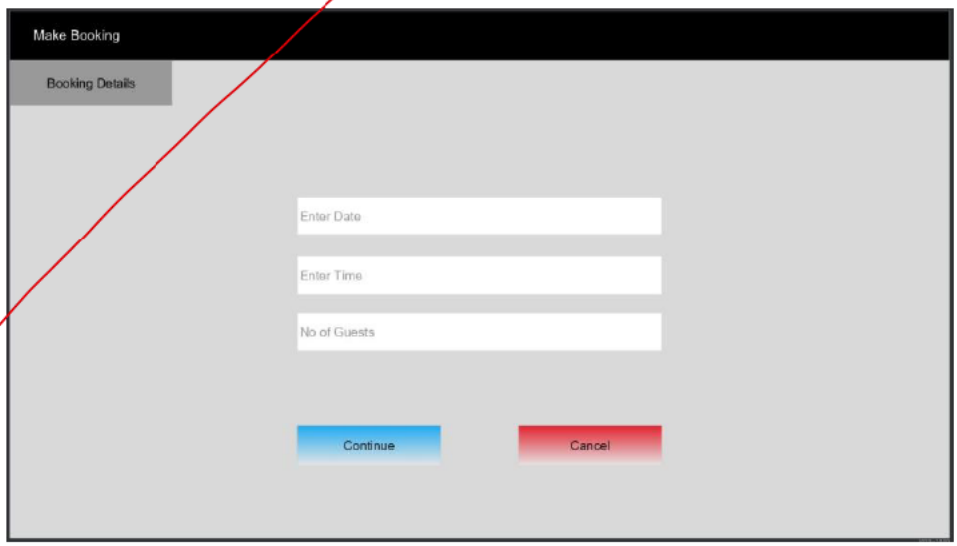
provide calendar selector

* Provide way to enter multiple waiters for specific date/time period ?

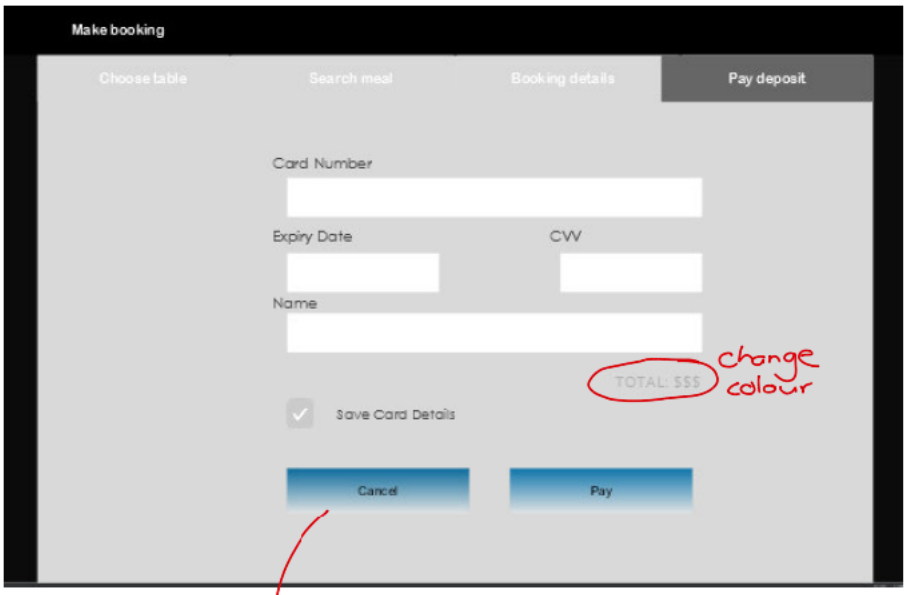
will this be start time only ?

Use Case ID	Use Case Name
A0200	Update Table Layout
Primary Business Actors	
User Admin	
Description	To update the restaurant table layout.
Pre-Conditions	Changes must have been made on the restaurant table layout.
Triggers	Select update table layout from the user admin dashboard.
Post-Conditions	Table layout is updated to the current table layout.
Basic Flow of Events	1. User admin clicks on Update table layout. 2. Old table layout is displayed. 3. System shows an upload dialog. 4. User admin uploads the new table layout.
Initial UI design	

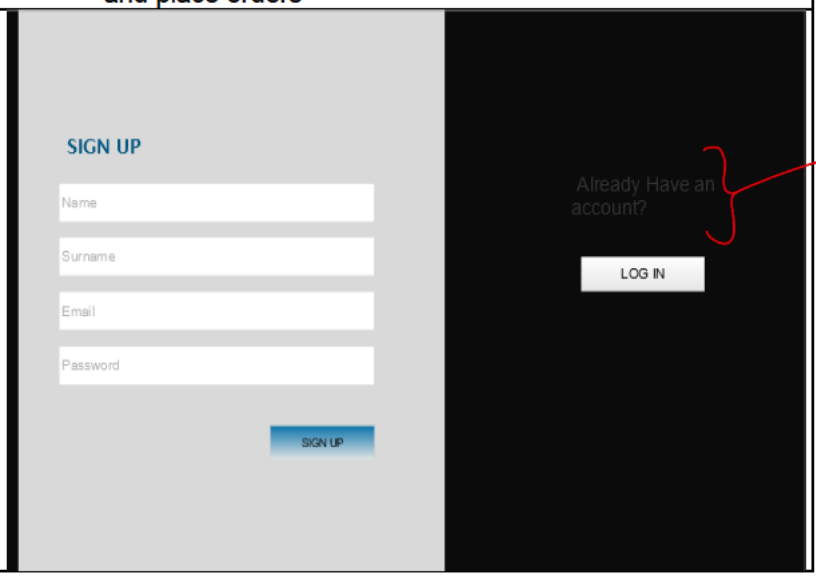
- how ?
- where is the upload file dialog ?

Use Case ID	Use Case Name	
C0100	Make Booking	
Primary Business Actors		**Other participating Actors**
Customer		
Description	Customer can make a booking for a specific time and date at the restaurant.	
Pre-Conditions	User is logged in on their account.	
Triggers	Customer clicks on “Make Booking”.	
Post-Conditions	The customer has a booking at the restaurant for a certain day at a certain time.	
Basic Flow of Events	1. Dashboard is displayed. 2. Customer clicks on make booking. 3. Customer enters details. 4. Customer clicks on continue.(Call C0110)(Call C0120)	
Initial UI design		

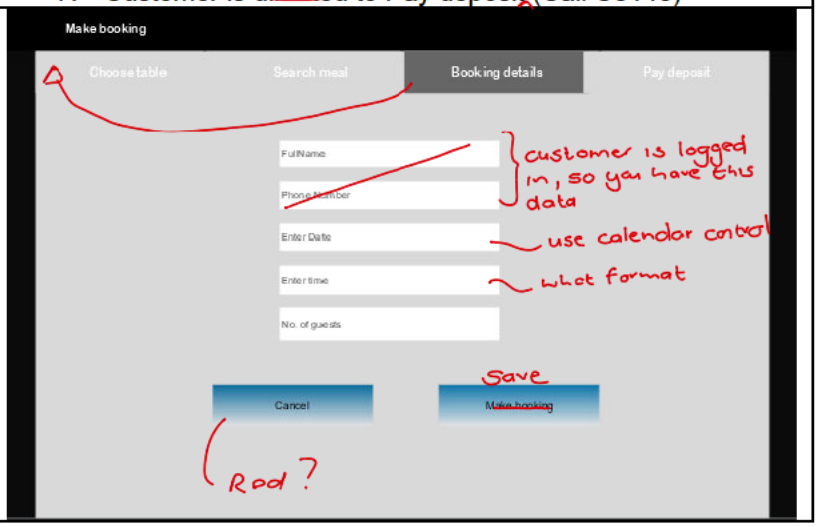
use one on p17 rather

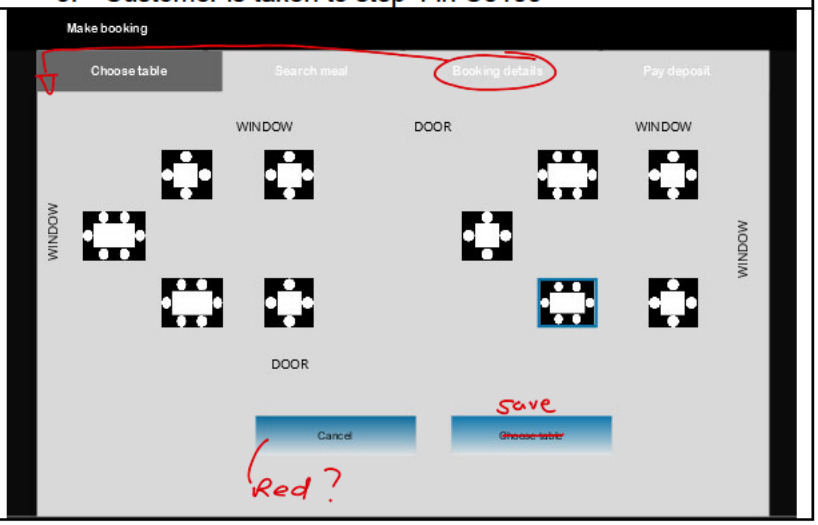
Use Case ID	Use Case Name	
C0140	Pay deposit	
Primary Business Actors		**Other participating Actors**
Customer		
Description	Customer pays a deposit.	
Pre-Conditions	Customer has filled all the important fields in order to make a booking and is now ready to pay their deposit. Pay Deposit tab	
Triggers	Customer clicks "~~Make Booking~~" in C0100.	
Post-Conditions	A booking is sent for confirmation to the restaurant	
Basic Flow of Events	~~1.~~ Call 0100. (why?) 2. Customer is taken to the payment screen by the system. 3. Customer enters all the necessary details in the appropriate places. 4. Customer clicks on "Pay". 5. The deposit is paid and the booking is sent for confirmation..	
Alternate Flow of events △	6. Display messoge "Booking made", then return to Home screen	
Initial UI design		

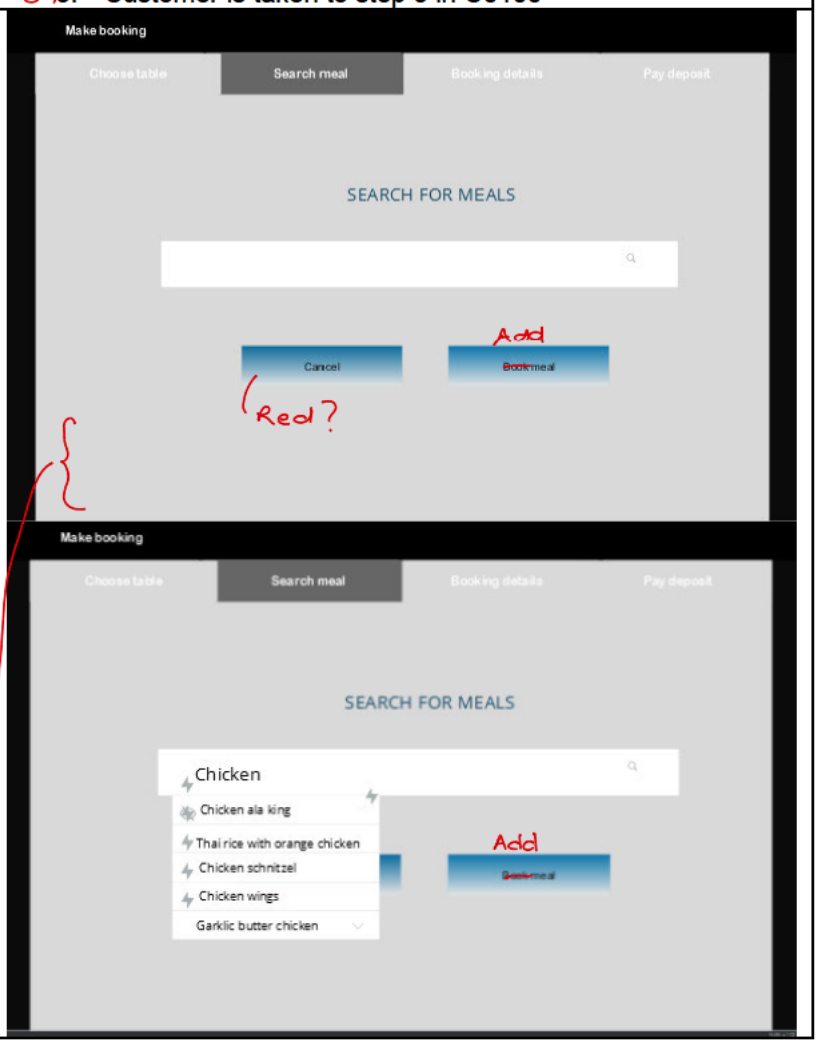
2.3.2 Designed by [redacted]

Use Case ID	Use Case Name
B0200	Register User
Primary Business Actors	
Customer	
Description	A first time customer registers an account for themselves.
Pre-Conditions	The customer is on the registration page, ready to register themselves onto the system.
Triggers	Select Register Sign up
Post-Conditions	The customer is registered onto the system and is now able to make a booking if they wish to do so.
Basic Flow of Events	1. Registration form is displayed 2. Customer inputs the relevant details into the relevant fields on the registration form. 3. Once customer is finished with their registration, they are assigned a customerID and are able to make bookings and place orders
Initial UI design	 text unclear, change colour

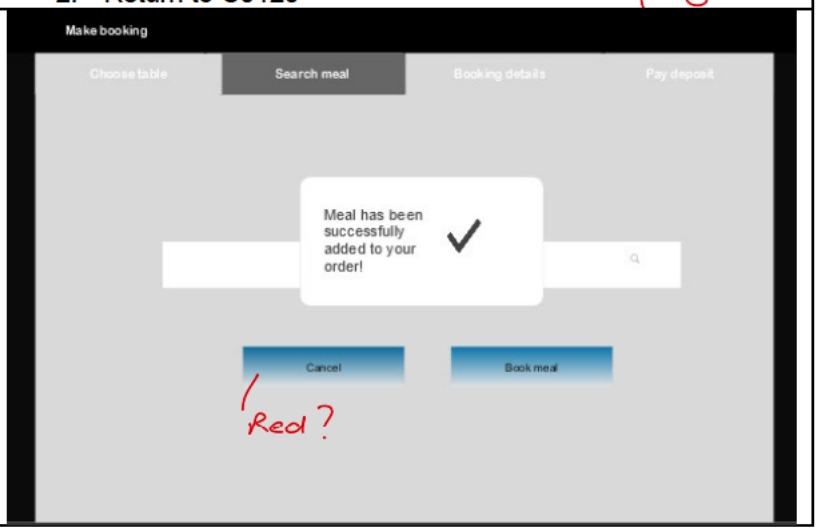
Use Case ID	Use Case Name
C0100	Make Booking
Primary Business Actors	
Customer	
Description	Customer can make a booking for a specific time at the restaurant

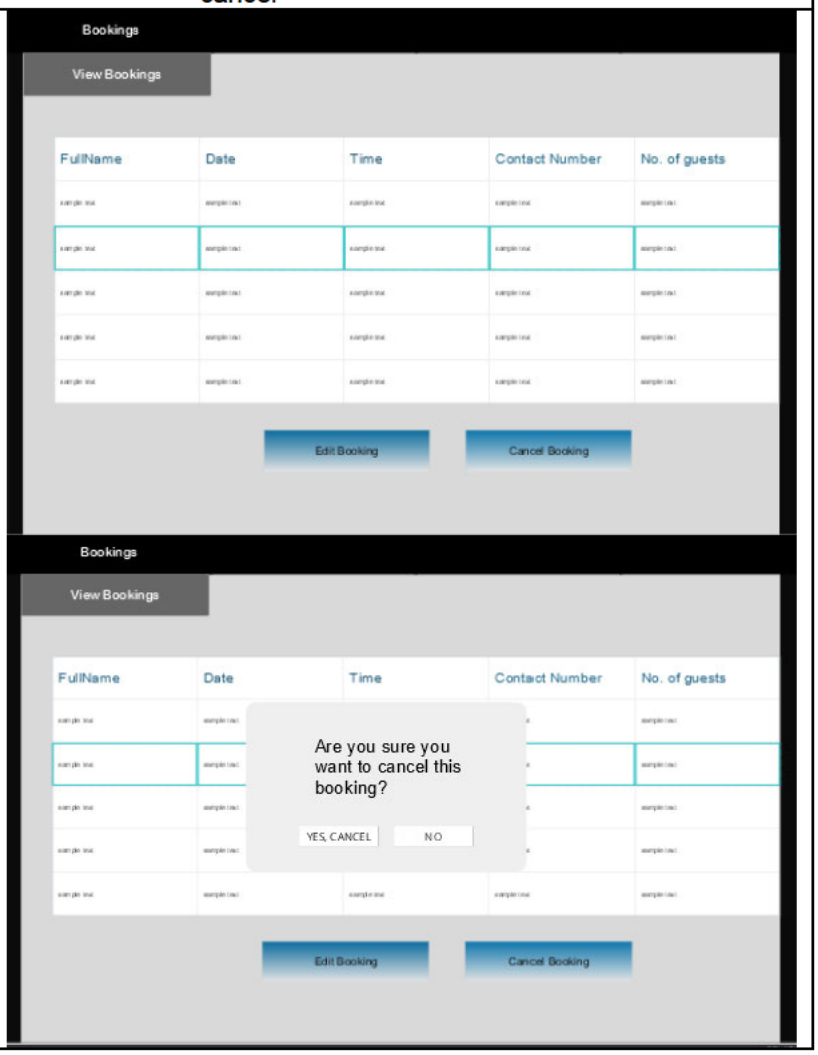
Use Case ID	Use Case Name
C0100	Make Booking
Primary Business Actors	
Customer	
Pre-Conditions	Customer is logged onto system
Triggers	Customer clicks on make booking in Main Menu Bar
Post-Conditions	~~Customer is able to select meals for the reservation~~ Booking mode
Basic Flow of Events	1. Dashboard is displayed with Booking Details tab open 2. ~~Customer clicks in make booking~~ 3. Customer ~~can now~~ select table (Call C0110) 4. Customer ~~searches for~~ meals (Call C0120) 5. Customer fills in booking details + clicks SAVE 6. ~~Customer clicks "make booking"~~ 7. Customer is ~~directed to~~ Pay deposit (Call C0140)
Initial UI design	

Use Case ID	Use Case Name
C0110	Choose table
Primary Business Actors	
Customer	
Description	A customer can choose which table to sit.
Pre-Conditions	Customer is logged onto the system and clicked on to make a reservation
Triggers	Customer clicks select table in C0100
Post-Conditions	Customer table selection is added to the reservation
Basic Flow of Events	1. Table layout is displayed 2. Customer selects a table 3. Customer is taken to step 4 in C0100
Initial UI design	

C0120	Search for meal
Primary Business Actors	
Customer	
Description	Customer can view the menu
Pre-Conditions	Customer is logged onto the system and are on the search page
Triggers	Customer ~~searches for~~ meals in C0100
Post-Conditions	If meal is on the menu it is displayed
Basic Flow of Events	1. Customer can search for a type of meat, starch, vegetable, beverage or dessert. 2. List of meals is displayed 3. Customer selects a meal 4. Call C0130 to add meal to order 5. Customer is taken to step 5 in C0100
Initial UI design	

Add

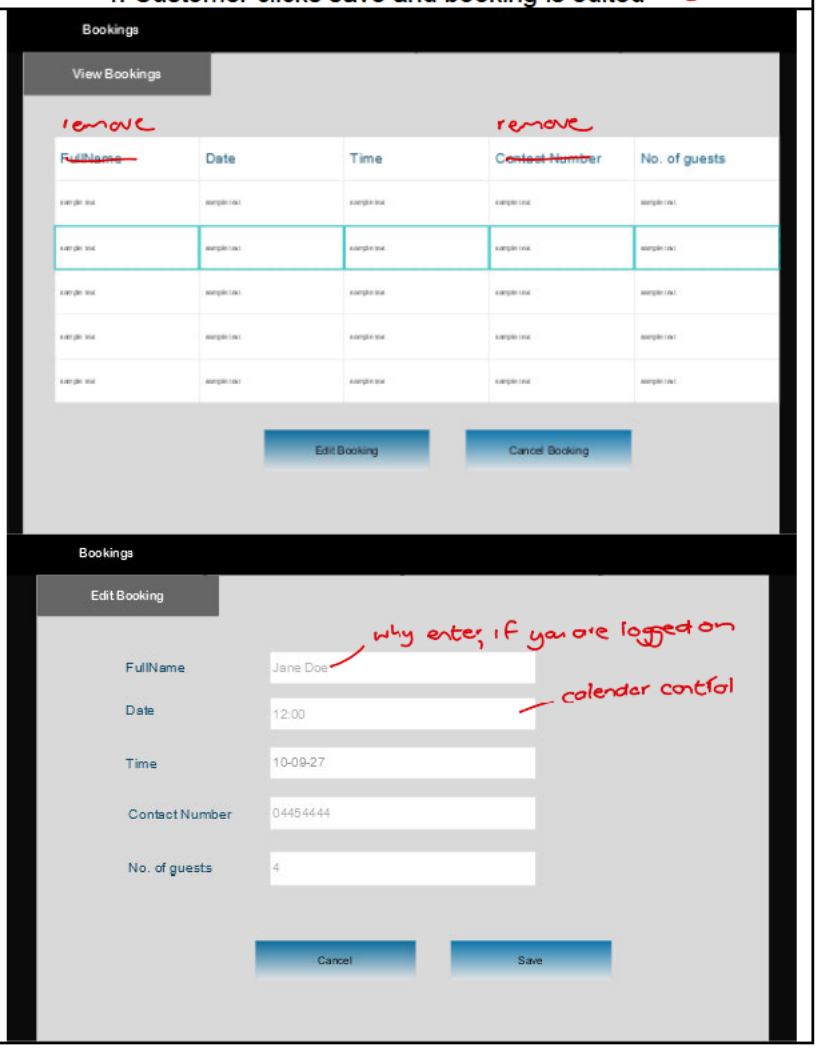
C0130	~~Book~~ meal
Primary Business Actors	
Customer	
Description	Customer can book a meal
Pre-Conditions	Customer is on the search meal page and a list of meals is displayed in C0120
Triggers	~~Customer clicks "Book meal~~" called from C120
Post-Conditions	Meal is added to the reservation
Basic Flow of Events	1. Meal is added to the customer order , confirmation msg displayed 2. Return to C0120
Initial UI design	

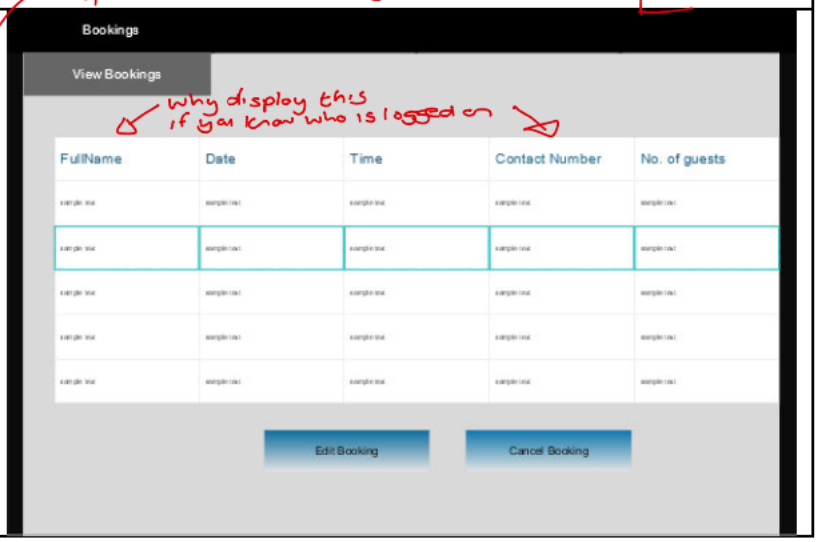
Use Case ID	Use Case Name
C0310	Cancel booking
Primary Business Actors	**Other participating Actors**
Description	Customer can cancel their booking
Pre-Conditions	Customer has already made a booking
Triggers	Customer clicks on ''Cancel Booking''
Post-Conditions	Booking is cancelled and the restaurant is notified
Basic Flow of Events	1. Customer clicks on View booking 2. Selects a booking 3. Customer clicks on ''Cancel booking 4. Booking is cancelled if customer clicks "Yes, cancel"
Initial UI design	

System display pop-up to confirm Cancel

Use Case ID	Use Case Name
C0320	Edit booking

Primary Business Actors	Other participating Actors

Description	Customer can edit their booking
Pre-Conditions	Customer has already made a booking
Triggers	Customer clicks on ''Edit Booking'' in C300
Post-Conditions	Booking is Edited and the restaurant is notified
Basic Flow of Events	1. ~~Customer clicks on View booking~~ } system opens edit booking form 2. ~~Selects a booking~~ 3. ~~Customer clicks on ''Edit booking''~~ } 4. Customer makes changes on edit booking form 4. Customer clicks save and booking is edited 5. Return to C300
Initial UI design	

Use Case ID	Use Case Name
C0300	View booking
Primary Business Actors	
Customer	
Description	Customer can view their bookings
Pre-Conditions	Customer is logged onto system and has made a booking
Triggers	Customer clicks on ~~show~~ bookings in Main Menu Bar
Post-Conditions	Current reservation along with previous bookings are displayed
Basic Flow of Events	1. Customer clicks on View bookings 2. Current and past bookings are displayed
Initial UI design	

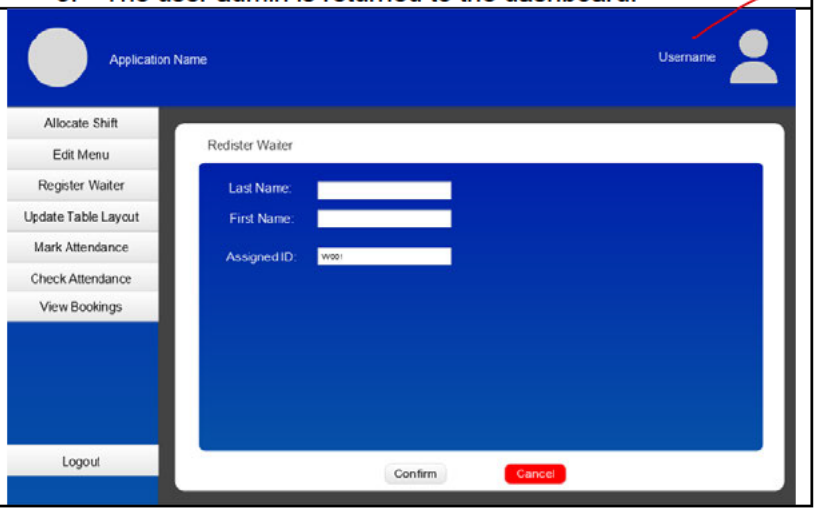
Alt 4. △

Select to Cancel (Call C310)

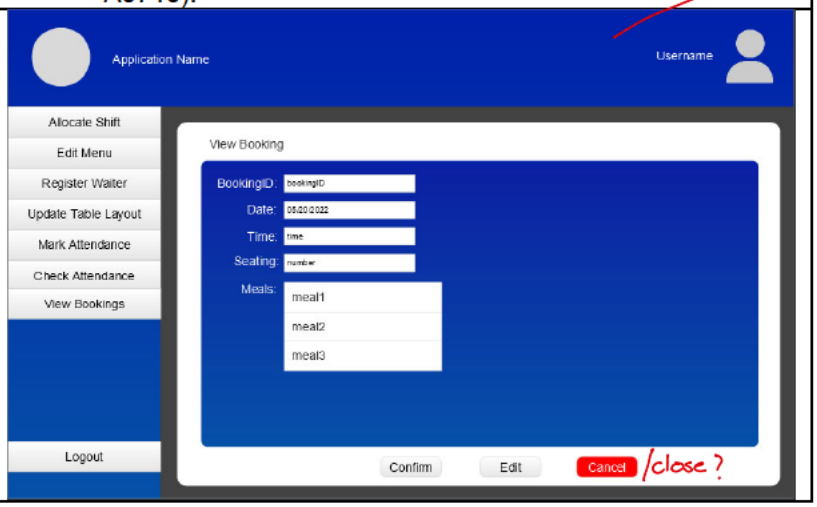
4. Select to Edit (call C320)

3. Customer select a booking in table

2.3.3 Designed by [redacted]

Use Case ID	Use Case Name
B0100	Register waiter
Primary Business Actors	**Other participating Actors**
User admin	
Description	User admin registers a new waiter onto the system.
Pre-Conditions	User admin logged into the system.
Triggers	Select “Register new waiter”.
Post-Conditions	Waiter is registered onto the system and is given a waiterID.
Basic Flow of Events	1. Display new waiter screen. 2. Waiter’s details are entered by the user admin. 3. Waiter’s details are registered by the system. when Confirm is clicked 4. Waiter is assigned a waiterID by the system. 5. The waiter is added to the restaurant work schedule.
Alternate Flow of events	1. Display new waiter screen. 2. User admin clicks on “Cancel". 3. The user admin is returned to the dashboard.
Initial UI design	 colour inconsistent

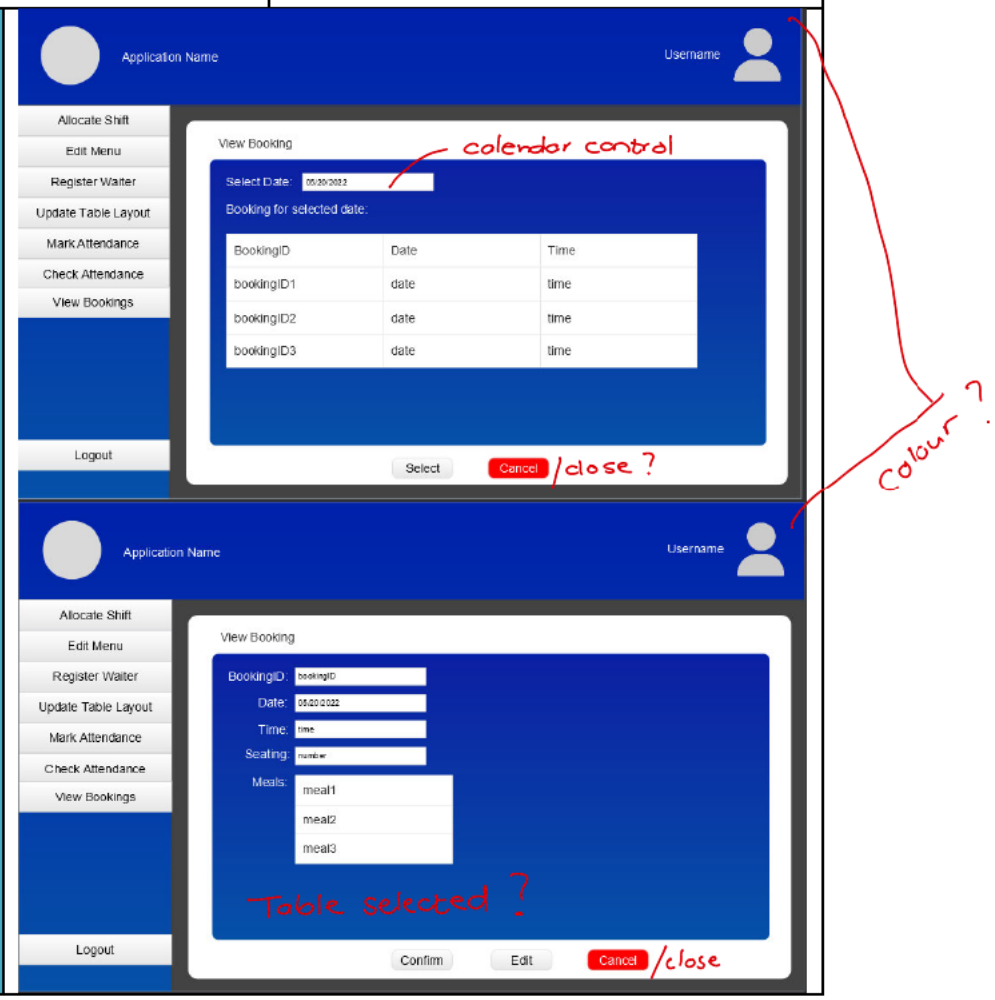
Use Case ID	Use Case Name
A0700	Confirm booking
Primary Business Actors	**Other participating Actors**
User admin	
Description	Admin confirms booking and a notification is sent to the customer.
Pre-Conditions	Customer made a booking, and the user admin is logged into the system.
Triggers	User admin clicks on “Confirm booking” in A0710.
Post-Conditions	Booking is confirmed by the restaurant.
Basic Flow of Events	~~1. Call A0710.~~ 2. User admin clicks on “Confirm”. 3. Booking is confirmed. 4. Notification sent to the customer.
Alternate Flow of events	~~1. Call A0710~~ 2. User admin clicks on “Cancel” 3. The booking is canceled.

Use Case ID	Use Case Name	
A0700	Confirm booking	
Primary Business Actors		**Other participating Actors**
User admin		
	4. The system returns to the "View booking" stage (Calls A0710).	
Initial UI design		

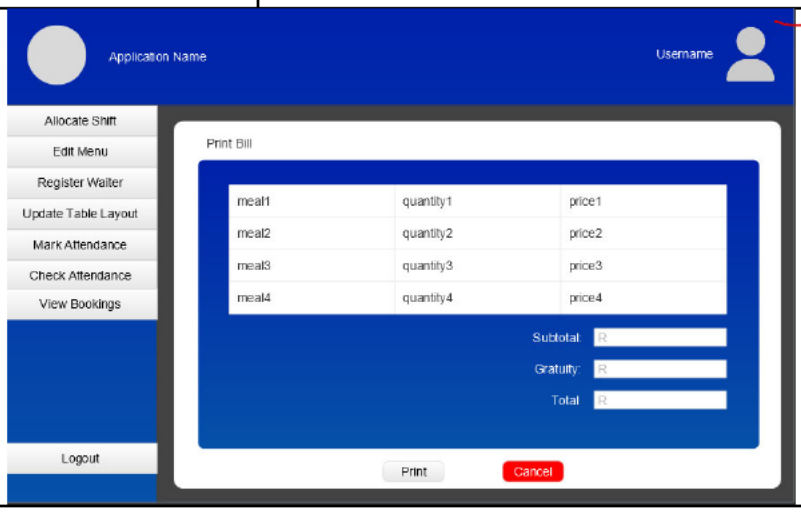
colour inconsistent

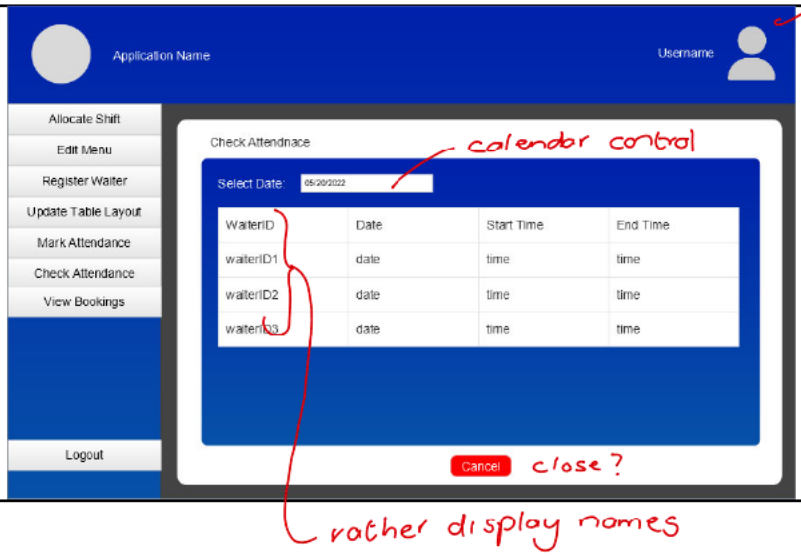
/close ?

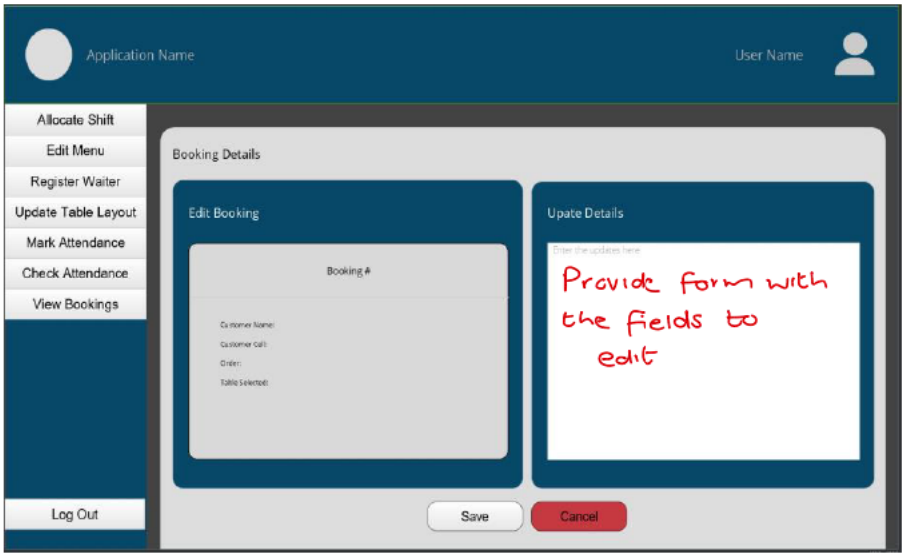
Use Case ID	Use Case Name	
A0710	View booking	
Primary Business Actors		**Other participating Actors**
User admin		
Description	Admin can view the bookings made by customers.	
Pre-Conditions	A booking has been made by the customer.	
Triggers	User admin clicks on "View booking".	
Post-Conditions	Booking is viewed by the user admin.	
Basic Flow of Events	1. Admin selects a particular day. 2. Bookings for that day are displayed. 3. Admin selects a particular booking. 4. Information for that booking is displayed., 5. Selects to confirm (call A700)	
Alternate Flow of events	4.1. Select to Edit, call A720	

Use Case ID	Use Case Name
A0710	View booking
Primary Business Actors	**Other participating Actors**
User admin	
Initial UI design	

Use Case ID	Use Case Name
A0730	Print bill
Primary Business Actors	**Other participating Actors**
User admin	
Description	Admin prints a particular bill.
Pre-Conditions	An order has been made.
Triggers	Admin clicks on "Print bill". from where (no option in A710)
Post-Conditions	Bill is shown on screen and printed.
Basic Flow of Events	1. Bill is displayed on screen. 2. Admin confirms that the bill should be printed. 3. Bill is printed.
Alternate Flow of events	?

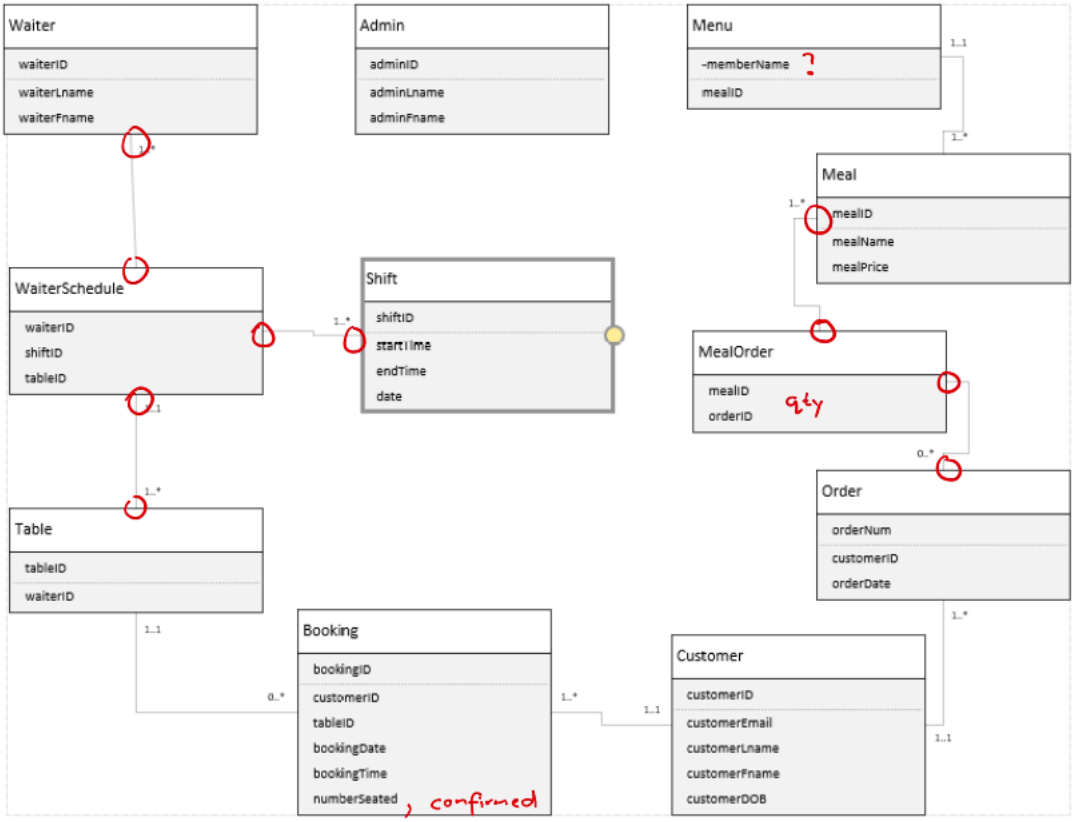
Use Case ID	Use Case Name
A0730	Print bill
Primary Business Actors	**Other participating Actors**
User admin	
Initial UI design	

Use Case ID	Use Case Name
A0600	Check attendance
Primary Business Actors	**Other participating Actors**
User admin	
Description	User admin checks which waiters were present for a particular shift.
Pre-Conditions	User admin is logged onto the system.
Triggers	User admin clicks on “Check attendance”.
Post-Conditions	Waiter attendance is displayed.
Basic Flow of Events	1. User admin selects a date. 2. Staff attendance is displayed for that date.
Alternate Flow of events	3. User admin clicks “Cancel” and is returned to the dashboard.
Initial UI design	

Use Case ID	Use Case Name	
A0720	Edit Booking	
Primary Business Actors		
User Admin		
Description	Admin can add or delete a booking. They can also change the order.	
Pre-Conditions	Alternate flow from A0710 after a booking is selected to edit.	
Triggers	User admin clicks on "Edit" in A0710.	
Post-Conditions	Booking is updated or modified depending on what option is chosen.	
Basic Flow of Events	1. ~~User admin clicks on booking to be viewed (Call A0710).~~ } display fields to edit 2. ~~User admin clicks on "Edit" or "Delete"~~ 3. User admin enters the relevant details (~~if "Edit" was chosen~~). 4. User admin clicks on "Confirm". 5. Booking has been edited ~~or deleted~~.	
Initial UI design		

3 DATA REQUIREMENTS

Implementation Ready Class Diagram



pk , fk not indicated

4 Department of Computing Sciences Plagiarism Declaration - Elaboration Document

Module code: WRRV301

We,

hereby declare that this submission is our own, original work.

We further declare that:

1. No part of this submission has been copied from another person/group,
2. We **did / did not** work with another person/group on this submission,
3. We acknowledged all consulted sources in the text and submitted a list of references, and
4. Parts without references are entirely our own work
5. That we have all equally contributed to the work or as indicated in the contribution % above.

We understand that, should this declaration be false, we may be charged with academic misconduct and/or plagiarism and that a disciplinary hearing may be brought against us.

1. SIGNED: [redacted]

[redacted]

[redacted]