

## Backend Tests

Test No:	API:	Class	Test Name:	Function:	Expected	Result:	Test Date:	Run by:
1	Roster	Controller	givenRoster_whenGetRosterById_thenReturnRoster	Testing that the roster controller correctly returns a collection of roster objects with the corresponding IDs when a get request is sent	We see a collection of roster objects with the correct ID	Passed	16/9/20	Julian
2	Roster	Controller	givenRoster_whenGetAllRosters_thenReturnRosterList	Testing that the roster controller when all of the roster objects are returned when a get request is sent	We see a collection of all of the roster objects in the system	Passed	16/9/20	Julian
3	Roster	Controller	givenRoster_whenDeleteRosterById_thenSuccessfulDeleteRequest	Testing that the roster controller when a delete request correctly removes the corresponding roster object	We see the roster object removed from our roster table	Passed	16/9/20	Julian
4	Roster	Controller	whenInsertRosterWithNoTime_thenReturnFalse	Testing that a roster object will be rejected if the time field is empty	We see a rejection message when we attempt to initialise a roster with a null time field	Passed	16/9/20	Julian
5	Roster	Controller	whenInsertRosterWithNoDate_thenReturnFalse	Testing that a roster object will be rejected if the date field is empty	We see a rejection message when we attempt to initialise a roster with a null date field	Passed	16/9/20	Julian
6	Roster	Controller	whenInsertRosterWithNonExistingEmployee_thenReturnFalse	Testing that a roster object will be rejected if the employee doesn't exist	We see a rejection message when we attempt to initialise a roster with a non-persistent employee	Passed	16/9/20	Julian
7	Roster	Controller	whenInsertRosterWithExistingEmployee_thenReturnTrue	Testing that a roster object will be accepted if the employee exists in the database	We see no rejection message	Passed	16/9/20	Julian
8	Repository	Controller	findAllRostersOfThenEmployee_returnListRoster	Testing that all roster objects are returned that correspond to the given employee	We see a list of rosters that have the given employee as their employee field	Passed	16/9/20	Julian
9	Repository	Controller	findAllRostersForNonExistingEmployee_returnNull	Testing that an empty list is returned when the given employee has no roster objects	We see an empty list returned	Passed	16/9/20	Julian
10	Repository	Controller	getAllRosters_returnListOfficers	Testing that all roster objects in the system are returned when we send a get request	We see a list of all rosters in the system	Passed	16/9/20	Julian
11	Repository	Controller	findAllRostersOnTable_thenAssertTrue	Testing that all roster objects in the system are returned when we send a get request	We see a list of all rosters in the system	Passed	16/9/20	Julian
12	Service	Controller	whenGettingRosterWithValidIdentifier_thenRosterShouldBeFound	Testing that a list of roster objects is returned when a roster object exists with the given employee	We see a list of roster objects with the given employee	Passed	16/9/20	Julian
13	Service	Controller	whenInvalidIdentifier_thenThrowRosterException	Testing that an empty list is returned when the given employee has no roster objects	We see an empty list returned	Passed	16/9/20	Julian
14	Service	Controller	getAllRoster_thenReturnListOfficers	Testing that a list of roster objects is returned when a get request is sent	We see a list containing all roster objects in the system	Passed	16/9/20	Julian
15	Service	Controller	whenDeletingRosterWithValidIdentifier_thenSuccessfulDeleteRequest	Testing that a roster object is deleted when a delete request is sent	We see the roster object removed from our roster table	Passed	16/9/20	Julian
16	Customer	Controller	givenCustomers_whenGetCustomers_thenReturnJsonArray	Testing that sending a get request returns an array containing all customers in the system	We see a list of customer objects	Passed	16/9/20	Jack
17	Controller	Controller	givenNoCustomers_whenGetCustomers_thenReturnNullArray	Testing that an empty customer table returns an empty array	We see an empty list returned	Passed	16/9/20	Jack
18	Customer	Controller	givenNewCustomer_whenPostNewCustomer_thenNewCustomerIsAdded	Testing that sending a post request adds a new customer to the database	We see a new customer added to the database	Passed	16/9/20	Jack
19	Customer	Controller	givenCustomerExists_whenDeleteCustomer_thenCustomerIsDeleted	Testing that sending a delete request removes the relevant customer	We see the customer removed from our customer table	Passed	16/9/20	Jack
20	Customer	Controller	whenInsertCustomerWithBlankUsername_thenReturnFalse	Testing that attempting to input a customer with a blank username is rejected	We see a rejection message returned	Passed	16/9/20	Jack
21	Customer	Controller	whenInsertCustomerWithValidFirstname_thenReturnTrue	Testing that inputting a customer with a non-blank username is accepted	We see a customer created	Passed	16/9/20	Jack
22	Customer	Controller	whenInsertCustomerWithBlankLastname_thenReturnFalse	Testing that attempting to input a customer with a blank lastname is rejected	We see a rejection message	Passed	16/9/20	Jack
23	Customer	Controller	whenInsertCustomerWithValidLastname_thenReturnTrue	Testing that inputting a customer with a non-blank last name is accepted	We see a customer successfully created	Passed	16/9/20	Jack
24	Customer	Controller	whenInsertUserNameWithLessThan3Characters_thenReturnFalse	Testing that attempting to input a customer with a <3 character username is rejected	We see a rejection message	Passed	16/9/20	Jack
25	Customer	Controller	whenInsertUserNameWithGreaterThan20Characters_thenReturnFalse	Testing that attempting to input a customer with a >20 character username is rejected	We see a rejection message	Passed	16/9/20	Jack
26	Customer	Controller	whenInsertUserNameBetween3and20Characters_thenReturnTrue	Testing that attempting to input a customer with a 3 < username.length <20 character username is accepted	We see a customer successfully created	Passed	16/9/20	Jack
27	Customer	Controller	whenInsertBlankEmail_thenReturnFalse	Testing that attempting to input a customer with a blank email is rejected	We see a rejection message	Passed	16/9/20	Jack
28	Customer	Controller	whenInsertBlankAddress_thenReturnFalse	Testing that attempting to input a customer with a blank address is rejected	We see a rejection message	Passed	16/9/20	Jack
29	Customer	Controller	whenInsertBlankPassword_thenReturnFalse	Testing that attempting to input a customer with a blank password is rejected	We see a rejection message	Passed	16/9/20	Jack
30	Repository	Controller	whenCustomerExistsWithUserName_thenReturnCustomer	Testing that when we send a get request containing the username corresponding to a given customer, that the customer object is returned	We see a customer object returned with the given username	Passed	16/9/20	Jack
31	Repository	Controller	whenCustomerDoesNotExistWithUserName_thenReturnNull	Testing that when we send a get request containing the username not corresponding to a given customer, that no object is returned	We see a blank object returned	Passed	16/9/20	Jack
32	Repository	Controller	whenCustomerExistsWithFirstname_thenReturnCustomer	Testing that when we send a get request containing the first not corresponding to a given customer, that no object is returned	We see a customer object returned with the given first name	Passed	16/9/20	Jack
33	Service	Controller	whenValidUserNameIsUsed_CustomerShouldBeReturned	Testing that when we send a post request with all valid inputs, a new customer is created	We see a new customer added to the database	Passed	16/9/20	Jack
34	Service	Controller	whenValidEmailIsUsed_CustomerShouldBeReturned	Testing that when we send a post request with an invalid email input, no customer is created	We see no customer added to the database	Passed	16/9/20	Jack
35	Service	Controller	whenValidEmailIsUsed_NullIsReturned	Testing that when we send a post request with an invalid email input, no customer is created	We see no customer added to the database	Passed	16/9/20	Jack
36	Service	Controller	whenGetAllIsUsed_ReturnListOfCustomers	Testing that when we send a customer get request, all customer objects are returned	We see a list containing all customers returned	Passed	16/9/20	Jack
37	Employee	Controller	givenEmployees_whenGetEmployees_thenReturnJsonArray	Testing that when we send a get request to the employee database, list of all employees is correctly returned	We see a list containing all employee objects	Passed	16/9/20	Theo
38	Controller	Controller	givenEmployeeId_whenGetEmployeeById_thenReturnEmployee	Testing that when we find an employee by their email address, we return the relevant employee	We see an employee object returned	Passed	16/9/20	Theo
39	Controller	Controller	givenEmployeeId_whenDeleteEmployeeById_thenReturnDelete	Testing that when a delete request is sent, the employee is deleted from the database and a valid message is returned	We see the employee deleted from the database	Passed	16/9/20	Theo
40	Controller	Controller	givenEmployeeId_whenAddingNewEmployee_thenSuccessfulPostRequest	Testing that when a post request is sent, a new employee is created in the database	We see a new employee added to the database	Passed	16/9/20	Theo
41	Controller	Controller	getAllEmployees_whenGetUserNameAndPassword_thenSuccessfulPostRequest	Testing that when we send a get request containing username and password, then employee is returned	We see the relevant employee returned	Passed	16/9/20	Theo
42	Repository	Controller	givenEmployeeWithValidEmail_thenReturnEmployee	Testing that when we find an employee by their email address, the relevant employee is returned	We are returned the relevant employee	Passed	16/9/20	Theo
43	Repository	Controller	findEmployeeWithIncorrectEmail_thenReturnNull	Testing that when we find employees by a null matching email, a null object is returned	We are returned a null object	Passed	16/9/20	Theo
44	Repository	Controller	findEmployeeWithPhoneNumber	Testing that when we find an employee by their phone number, the relevant employee is returned	We are returned the relevant employee	Passed	16/9/20	Theo
45	Repository	Controller	findEmployeeWithIncorrectPhoneNumber_thenReturnNull	Testing that when we find an employee by an invalid phone number, a null object is returned	We are returned a null object	Passed	16/9/20	Theo
46	Repository	Controller	findEmployeeWithValidUserName_thenReturnEmployee	Testing that when we find an employee by their username, the relevant employee is returned	We are returned the relevant employee	Passed	16/9/20	Theo
47	Repository	Controller	findEmployeeWithIncorrectUsername_thenReturnNull	Testing that when we find an employee by an invalid username, a null object is returned	We are returned a null object	Passed	16/9/20	Theo
48	Repository	Controller	getAllEmployees_thenReturnListOfEmployees	Testing that when we find all employees, then a list containing all employees in the database are returned	We are returned a list of all employee objects	Passed	16/9/20	Theo
49	Repository	Controller	getEmployeeByValidIdentifier_thenReturnEmployee	Testing that when we find an employee by their identifier, the relevant employee is returned	We are returned the relevant employee	Passed	16/9/20	Theo
50	Repository	Controller	getEmployeeByInvalidIdentifier_thenReturnNull	Testing that when we find an employee by an invalid identifier, a null object is returned	We are returned a null object	Passed	16/9/20	Theo
51	Repository	Controller	getEmployeeByUserNameAndPassword_returnEmployee	Testing that when we find an employee by their username and password, the relevant employee is returned	We are returned the relevant employee	Passed	16/9/20	Theo
52	Repository	Controller	getEmployeeByInvalidUserNameAndInvalidPassword_returnNull	Testing that when we find an employee by an invalid username and password, a null object is returned	We are returned a null object	Passed	16/9/20	Theo
53	Repository	Controller	getEmployeeByValidUserNameAndInvalidPassword_returnNull	Testing that when we find an employee by their username and an invalid password, a null object is returned	We are returned a null object	Passed	16/9/20	Theo
54	Repository	Controller	getEmployeeByInvalidUserNameAndValidPassword_returnNull	Testing that when we find an employee by an invalid username and valid password, a null object is returned	We are returned a null object	Passed	16/9/20	Theo
55	Service	Controller	whenValidIdentifier_ThenEmployeeShouldBeFound	Testing that when we find an employee by a corresponding identifier, the relevant employee is returned	We are returned the corresponding employee	Passed	16/9/20	Theo
56	Service	Controller	whenInvalidIdentifier_ThrowEmployeeException	Testing that when we find an employee by an invalid identifier, we are returned a null object	We are returned a null object	Passed	16/9/20	Theo
57	Service	Controller	whenValidUserName_ThenEmployeeShouldBeFound	Testing that when we find an employee by a valid email, we are returned the valid employee	An exception is thrown	Passed	16/9/20	Theo
58	Service	Controller	whenValidUserName_ThrowEmployeeException	Testing that when we find an employee by an invalid username, an exception is thrown	We are returned the valid employee	Passed	16/9/20	Theo
59	Service	Controller	whenValidEmail_ThenEmployeeShouldBeFound	Testing that when we find an employee by a valid email, we are returned the valid employee	An exception is thrown	Passed	16/9/20	Theo
60	Service	Controller	whenValidEmail_ThenThrowEmployeeException	Testing that when we find an employee by an invalid email, an exception is thrown	We are returned the valid employee	Passed	16/9/20	Theo
61	Service	Controller	whenValidPhoneNumber_ThenEmployeeShouldBeFound	Testing that when we find an employee by a valid phone number, we are returned the valid employee	An exception is thrown	Passed	16/9/20	Theo
62	Service	Controller	whenValidPhoneNumber_ThenThrowEmployeeException	Testing that when we find an employee by an invalid phone number, an exception is thrown	We are returned the valid employee	Passed	16/9/20	Theo
63	Service	Controller	getAllEmployees_returnListOfEmployees	Testing that when we get all employees, a list of all employees in the database are returned	A list of employees in the database are returned	Passed	16/9/20	Theo
64	Service	Controller	getEmployeeUsingValidUserNameAndPassword_returnEmployee	Testing that when we find an employee by their username and password, the relevant employee is returned	We are returned the relevant employee	Passed	16/9/20	Theo
65	Service	Controller	getEmployeeWithInvalidUserNameAndPassword_returnEmployeeException	Testing that when we find an employee by an invalid username and password, a null object is returned	An exception is thrown	Passed	16/9/20	Theo
66	Service	Controller	getEmployeeWithValidUserNameAndInvalidPassword_returnEmployeeException	Testing that when we find an employee by their username and an invalid password, a null object is returned	An exception is thrown	Passed	16/9/20	Theo
67	Service	Controller	getEmployeeWithInvalidUserNameAndValidPassword_returnEmployeeException	Testing that when we find an employee by an invalid username and valid password, a null object is returned	An exception is thrown	Passed	16/9/20	Theo
68	Employee	Controller	whenInsertEmployeeFirstnameWithWrongSize_thenReturnFalse	Testing that when we try to insert an employee with an incorrect length, the employee is rejected	The employee is rejected	Passed	16/9/20	Theo
69	Employee	Controller	whenInsertFirstnameGreaterthanThreeCharacter_thenReturnTrue	Testing that when we try to insert an employee with first name greater than 3 characters, the employee is accepted	The employee is accepted	Passed	16/9/20	Theo
70	Employee	Controller	whenInsertLastnamelessthanThreeCharacters_thenReturnFalse	Testing that when we try to insert an employee with a last name less than 3 characters, the employee is rejected	The employee is rejected	Passed	16/9/20	Theo
71	Employee	Controller	whenInsertLastnamegreaterThanThreeCharacters_thenReturnTrue	Testing that when we try to insert an employee with a last name greater than 3 characters, the employee is accepted	The employee is accepted	Passed	16/9/20	Theo
72	Employee	Controller	insertFirstnameThatIsBlank_thenReturnFalse	Testing that when we insert an employee with a blank first name, the employee is rejected	The employee is rejected	Passed	16/9/20	Theo
73	Employee	Controller	insertLastnameThatIsBlank_thenReturnTrue	Testing that when we insert an employee with a blank last name, the employee is accepted	The employee is accepted into the database	Passed	16/9/20	Theo
74	Employee	Controller	insertEmailThatIsBlank_thenReturnFalse	Testing that when we insert an employee with a blank email, the employee is rejected	The employee is rejected	Passed	16/9/20	Theo
75	Employee	Controller	insertEmailThatIsNotBlank_thenReturnTrue	Testing that when we insert an employee with a non-blank email, the employee is accepted	The employee is accepted	Passed	16/9/20	Theo
76	Employee	Controller	insertAddressThatIsBlank_thenReturnFalse	Testing that when we insert an employee with a blank address, the employee is rejected	The employee is rejected	Passed	16/9/20	Theo
77	Employee	Controller	insertAddressThatIsNotBlank_thenReturnTrue	Testing that when we insert an employee with a non-blank address, the employee is accepted	The employee is accepted	Passed	16/9/20	Theo
78	Employee	Controller	insertUserNameThatIsBlank_thenReturnFalse	Testing that when we insert an employee with a blank username, the employee is rejected	The employee is rejected	Passed	16/9/20	Theo
79	Employee	Controller	insertUserNameThatIsNotBlank_thenReturnTrue	Testing that when we insert an employee with a non-blank address, the employee is accepted	The employee is accepted	Passed	16/9/20	Theo
80	Employee	Controller	insertPasswordThatIsBlank_thenReturnFalse	Testing that when we insert an employee with a blank password, the employee is rejected	The employee is rejected	Passed	16/9/20	Theo
81	Employee	Controller	insertPasswordThatIsNotBlank_thenReturnTrue	Testing that when we insert an employee with a non-blank password, the employee is accepted	The employee is accepted	Passed	16/9/20	Theo
82	Employee	Controller	insertIdentifierThatIsBlank_thenReturnCanNotBeBlank	Testing that when we insert an employee with a blank identifier, the employee is rejected	The employee is rejected	Passed	16/9/20	Theo
83	Employee	Controller	insertCorrectIdentifier_returnTrue	Testing that when we insert an employee with a non-blank identifier, the employee is accepted	The employee is accepted	Passed	16/9/20	Theo
84	ServiceType	Controller	givenServiceType_whenGetServiceType_thenReturnJsonArray	Testing that when we send a get request, all services in the database are returned in a list	A JSON list is returned with all of the services	Passed	16/9/20	Ghaida
85	Controller	Controller	givenNoServiceType_whenGetServiceType_thenReturnNullArray	Testing that when no services exist in the database, a blank list is returned when we send a get request	An empty JSON list is returned	Passed	16/9/20	Ghaida
86	ServiceType	Controller	givenNewServiceType_whenPostNewServiceType_thenNewServiceTypeIsAdded	Testing that when we send a post request containing valid service information, that a new service is created in the database	The new service is accepted and added to the database	Passed	16/9/20	Ghaida
87	ServiceType	Controller	whenInsertServiceTypeWithBlankServiceNo_thenReturnFalse	Testing that when we attempt to insert into the database a service with no service number, the object is rejected	The service is rejected	Passed	16/9/20	Ghaida
88	ServiceType	Controller	whenInsertServiceTypeWithValidServiceNo_thenReturnTrue	Testing that when we attempt to insert into the database a service with all valid fields, the object is accepted	The service is accepted	Passed	16/9/20	Ghaida
89	ServiceType	Controller	whenInsertServiceTypeWithBlankServiceName_thenReturnFalse	Testing that when we attempt to insert into the database a service with a blank service name, the object is rejected	The service is rejected	Passed	16/9/20	Ghaida
90	ServiceType	Controller	whenInsertServiceTypeWithValidServiceName_thenReturnTrue	Testing that when we attempt to insert into the database a service with all valid fields, the object is accepted	The service is accepted	Passed	16/9/20	Ghaida
91	Repository	Controller	whenFindServiceTypeByNo_thenReturnServiceName	Testing that when we find all services by service number, we are returned the relevant service objects	The relevant service objects are returned in a list	Passed	16/9/20	Ghaida
92	Repository	Controller	whenFindServiceTypeByServiceName_thenReturnServiceName	Testing that when we find all services by service name, we are returned the relevant service objects	The relevant service objects are returned in a list	Passed	16/9/20	Ghaida
93	Repository	Controller	whenServiceTypeDoesNotExistWithServiceName_thenReturnNull	Testing that when we find all services by service name, and no services with that name exist, a null object is returned	A null object is returned	Passed	16/9/20	Ghaida
94	Service	Controller	whenValidName_ThenServiceTypeShouldBeFound	Testing that when we find a service by its name, we return the relevant service	The relevant service is returned	Passed	16/9/20	Ghaida
95	Service	Controller	getAlServiceType_returnListOfServiceType	Testing that when we get all service objects, we return all the services in the database in a list	A list of services is returned containing all services in the database	Passed	16/9/20	Ghaida
96	Booking	Controller	givenBooking_whenGetBookingById_thenReturnBooking	Testing that when we find a booking by their id, we are returned the relevant booking in the database	The relevant booking is returned	Passed	16/9/20	Dulshan
97	Booking	Controller	givenBooking_whenDeleteBookingById_thenSuccessfulDeleteRequest	Testing that when we send a delete request, a list of all bookings in the database is returned	A list of all bookings in the database is returned	Passed	16/9/20	Dulshan
98	Booking	Controller	whenInsertBookingWithNoTime_thenReturnFalse	Testing that when we send a delete request, the relevant booking is removed from the database	The relevant booking is removed from the database	Passed	16/9/20	Dulshan
99	Booking	Controller	whenInsertBookingWithNoDate_thenReturnFalse	Testing that when we create a booking with no time attribute, the booking is rejected	The booking is rejected	Passed	16/9/20	Dulshan
100	Booking	Controller	whenInsertBookingWithNoCustomer_thenReturnFalse	Testing that when we create a booking with no date attribute, the booking is rejected	The booking is rejected	Passed	16/9/20	Dulshan
101	Booking	Controller	whenInsertBookingWithNoCustomer_thenReturnFalse	Testing that when we create a booking with no customer attribute, the booking is rejected	The booking is rejected	Passed	16/9/20	Dulshan
102	Booking	Controller	whenInsertBookingWithNoEmployee_thenReturnFalse	Testing that when we create a booking with no employee attribute, the booking is rejected	The booking is rejected	Passed	16/9/20	Dulshan
103	Booking	Controller	whenInsertBookingWithValidValues_thenReturnTrue	Testing that when we create a booking with all attributes valid, then the booking object is created	The new booking is created	Passed	16/9/20	Dulshan

104	Repository	whenFindBookingByValidEmployee_thenAssertTrue	Testing that when we find bookings by an employee, we find all of the relevant bookings in the database	A list of booking objects are returned associated with the relevant employee	Passed	16/9/20	Dulshan
105	Repository	whenFindBookingByNonExistentEmployee_thenAssertFalse	Testing that when we find bookings by an employee that doesn't exist in the database, we are returned an empty list	An empty list is returned	Passed	16/9/20	Dulshan
106	Repository	whenFindBookingByValidCustomer_thenAssertTrue	Testing that when we find bookings by a customer, we find all of the relevant bookings in the database	A list of booking objects are returned associated with the relevant customer	Passed	16/9/20	Dulshan
107	Repository	whenFindBookingByNonExistentCustomer_thenAssertFalse	Testing that when we find bookings by a customer that doesn't exist in the database, we are returned an empty list	An empty list is returned	Passed	16/9/20	Dulshan
108	Repository	whenFindAllThreeBookingsEntered_thenAssertTrue	Testing that when we find all bookings in the database, all of the booking objects are returned	A list with all booking objects are returned	Passed	16/9/20	Dulshan
109	Service	whenGettingBookingWithValidIdentifier_thenBookingShouldBeFound	Testing that when we get bookings by the booking identifier, the relevant booking is returned	The relevant booking object is returned	Passed	16/9/20	Dulshan
110	Service	whenInvalidIdentifier_thenThrowBookingException	Testing that when we get bookings by an invalid identifier, a booking exception is thrown	A booking exception is thrown	Passed	16/9/20	Dulshan
111	Service	getAllBookings_thenReturnListofBookings	Testing that when we get all bookings, a list of all of the bookings in the database are returned	A list of all of the bookings in the database are returned	Passed	16/9/20	Dulshan
112	Service	whenDeletingBookingWithValidIdentifier_thenSuccessfullyDeleteRequest	Testing that when we delete bookings by a valid identifier, the relevant booking is removed from the database	The relevant booking is removed from the database	Passed	16/9/20	Dulshan
113	Service	whenDeletingBookingWithInvalidIdentifier_thenThrowBookingException	Testing that when we delete bookings by an invalid identifier, a booking exception is thrown	A booking exception is thrown	Passed	16/9/20	Dulshan

## Frontend Tests

Test No:	Component	Description	Test Name:	Function:	Expected	Result:	Test Date:	Run by:
1	App	testing custAuth method	should loggedInStatus should be set to LOGGED_IN	Testing that when a customerAuth method is called the logged in status is set correctly	Login Status set to LOGGED_IN	Passed	16/9/20	Theo
2			customer should be set to true if method is called	Testing that when customerAuth is called that the customer state object should be set to true	The Customer state object contains all of the customer info	Passed	16/9/20	Theo
3			user should be created as an object in app state	Testing that the user object will be effectively created in the app state when customerAuth is called	The User state object contains all of the user info	Passed	16/9/20	Theo
4		Testing employeeAuth	if employeeAuth is called then Logged in status should be set to LOGGED_IN	Testing that when employeeAuth is called the logged in status is set correctly	Logged in status is set to LOGGED_IN	Passed	16/9/20	Theo
5			if employeeAuth is called the employee should be set to true	Testing that when employeeAuth is called the logged in status is set correctly	Logged in status is set to LOGGED_IN	Passed	16/9/20	Theo
6			if employeeAuth method is called a user should be in the App state	Testing that when employeeAuth is called the user state object is correctly updated	User state object contains all of the relevant employee info	Passed	16/9/20	Theo
7		Testing the toggleLogout method	Customer should be false	Testing that when toggleLogout is called, the customer state object is reset correctly	Customer object is set to false	Passed	16/9/20	Theo
8			Employee should be false	Testing that when toggleLogout is called, the employee state object is reset correctly	Employee object is set to false	Passed	16/9/20	Theo
9			loggedInStatus should be set to NOT_LOGGED_IN	Testing that when toggleLogout is called, the logged in status state object is change accordingly	Logged in status is set to NOT_LOGGED_IN	Passed	16/9/20	Theo
10			user should be blank	Testing that when toggleLogout is called, the user state object is updated accordingly	User state is updated to blank	Passed	16/9/20	Theo
11	Booking	<Booking /> component tests	when user redirected to home if not logged in	Testing that when a user navigated to the booking page without the logged in status set correctly, they are handled accordingly	User is navigated to the home page	Passed	16/9/20	Jack
12			Testing Services are Set when Booking is rendered	Testing that when the <Bookings> component has been rendered, the service state object is updated correctly	The services state object is set correctly	Passed	16/9/20	Jack
13			Testing Service updates state when selected	Testing that when a service is selected, the selected service state object is set correctly	The selected service state object is set to the relevant service number	Passed	16/9/20	Jack
14			Testing Service updates to null when null state when selected	Testing that when the service is selected to non-null selection, then update to the null selection, the selected service state object updates correctly	The selected service state object is set to null	Passed	16/9/20	Jack
15			Testing Employees Field is disabled when no service is selected	Testing that when no service is selected, the employee selection field is disabled	The employee selection field is set to disabled	Passed	16/9/20	Jack
16			Testing Date Field is disabled when no service is selected	Testing that when no service is selected, the date selection field is disabled	The date selection field is set to disabled	Passed	16/9/20	Jack
17			Testing Time Field is disabled when no service is selected	Testing that when no service is selected, the time selection field is disabled	The time selection field is set to disabled	Passed	16/9/20	Jack
18			Testing Submit Button is disabled when no service is selected	Testing that when no service is selected, the submit button is disabled	The submit button is set to disabled	Passed	16/9/20	Jack
19			Testing Employees update when service selected	Testing that when a service is selected, the employees state object is updated to contain the relevant employees	The employees state object contains the correct employees	Passed	16/9/20	Jack
20			Testing TimeSlots Are set When date is set	Testing that when a date is selected, the times state object is set and contains the relevant times	The times state object is set to be the correct length	Passed	16/9/20	Jack
21			Testing already booked times are not displayed	Testing that when a date is selected containing pre-booked times, no already booked times are displayed	Only the non-booked times are displayed	Passed	16/9/20	Jack
22			Testing submit button is enabled when all fields are selected	Testing that when all of the fields are set to not null, the submit button is enabled	The submit button is set to enabled	Passed	16/9/20	Jack
23			Testing submit button is disabled when fields are deselected	Testing that when the fields are selected then deselected, the submit button is disabled	The submit button is set to disabled	Passed	16/9/20	Jack
24	Header	Header elements	should contain nav elements Home, about us, staff Login, login and register	Testing that when user is not logged in, the navbar contains all of the relevant links	The navbar contains a link to home, about us, login, staffLogin and createLogin	Passed	16/9/20	Theo
25			handle login when employee is logged in	Testing that when an employee is logged in, the navbar contains the login component	The navbar contains a login component	Passed	16/9/20	Theo
26			handle logout when customer is logged in	Testing that when a customer is logged in, the navbar contains the logout component	The navbar contains a logout component	Passed	16/9/20	Theo
27	Login	<Login /> test for login component	Username check	Testing that when a user types their name into the login, the username state object is correctly set	The username state object is updated to the correct value	Passed	16/9/20	Theo
28			Password check	Testing that when a user types their password into the login, the password state object is correctly set	The password state object is updated to the correct value	Passed	16/9/20	Theo
29			toggleGet has been implemented	Testing that the toggleGet method correctly sends a get request and updates the state correctly	The loggedIn state object is set correctly	Passed	16/9/20	Theo
30			axios get request is successful should set logged in state to true	Testing that when the toggleGet method is called and the get request succeeds, the logged in status is set correctly	The loggedIn state is set to true	Passed	16/9/20	Theo
31			axios get request is unsuccessful and loggedIn state is false	Testing that when the toggleGet method is called and the get request is called unsuccessfully, the logged in status is set to false	The loggedIn state is set to false	Passed	16/9/20	Theo
32			Page redirects successfully after successful axios request	Testing that when a user successfully logs in, they are redirected to the home page	The component has now been rendered to the front end	Passed	16/9/20	Theo
33	Signup	Signup page check fields	First Name check	Testing that when the first name field is updated the state is updated correctly	The first name state object is set correctly	Passed	16/9/20	Theo
34			Last Name check	Testing that when the last name field is updated, the state is updated correctly	The last name state object is set correctly	Passed	16/9/20	Theo
35			Email check	Testing that when the email field is updated, the state is updated correctly	The email state object is set correctly	Passed	16/9/20	Theo
36			Address check	Testing that when the address field is updated, the state is updated correctly	The address state object is set correctly	Passed	16/9/20	Theo
37			Phone Number check	Testing that when the phone number field is updated, the state is updated correctly	The phone number state object is set correctly	Passed	16/9/20	Theo
38			Username check	Testing that when the username field is updated, the state is updated correctly	The username state object is set correctly	Passed	16/9/20	Theo
39			password check	Testing that when the password field is updated, the state is updated correctly	The password state object is set correctly	Passed	16/9/20	Theo
40			Confirm password check	Testing that when the confirm password field is updated, the state is updated correctly	The confirm password state object is set correctly	Passed	16/9/20	Theo
41			validate password to be true when passwords are the same	Testing that when both of the input password fields are set to the same value, the validate method returns true	The validate method returns true	Passed	16/9/20	Theo
42			Validate password to be false when passwords are not the same	Testing that when both of the input password fields are set to different values, the validate method returns false	The validate method returns false	Passed	16/9/20	Theo
43			should successfully perform a post request will set isSignedUp to true	Testing that a successful post request sets the isSignedUp state object to be true	The isSignedUp state object is set to true	Passed	16/9/20	Theo
44			should unsuccessfully perform a post request and isSignedUp Should be false	Testing that an unsuccessful post request will set isSignedUp to be false	The isSignedUp state object is set to false	Passed	16/9/20	Theo
45			should redirect page upon successful post request to homepage	Testing that a successful sign up will redirect the user to the home page	The home screen has been rendered	Passed	16/9/20	Theo
46	StaffLogin	Tests for Staff login Component	Username check in state	Testing that when the username field is updated, the state is updated correctly	The first name state object is set correctly	Passed	16/9/20	Theo
47			Password check in state	Testing that when the password field is updated, the state is updated correctly	The password state object is updated to the correct value	Passed	16/9/20	Theo
48			toggleGet has been implemented	Testing that the toggleGet method correctly sends a get request and updates the state correctly	The loggedIn state object is set correctly	Passed	16/9/20	Theo
49			axios get request is successful should set logged in state to true	Testing that when the toggleGet method is called and the get request succeeds, the logged in status is set correctly	The loggedIn state is set to true	Passed	16/9/20	Theo
50			axios get request is unsuccessful and loggedIn state is false	Testing that when the toggleGet method is called and the get request is called unsuccessfully, the logged in status is set to false	The loggedIn state is set to false	Passed	16/9/20	Theo
51			Page redirects successfully after successful axios request	Testing that when a user successfully logs in, they are redirected to the home page	The component has now been rendered to the front end	Passed	16/9/20	Theo
52	Customer	Customer Page information display	First Name check	Testing that when the user logged in changes the first name displayed is correct	The first name state object is set correctly	Passed	20/9/20	Dulshan
53			Last Name check	Testing that when the user logged in changes the last name displayed is correct	The last name state object is set correctly	Passed	21/9/20	Dulshan
54			Email check	Testing that when the user logged in changes the email displayed is correct	The email state object is set correctly	Passed	22/9/20	Dulshan
55			Address check	Testing that when the user logged in changes the address displayed is correct	The address state object is set correctly	Passed	23/9/20	Dulshan
56			Phone Number check	Testing that when the user logged in changes the phone number displayed is correct	The phone number state object is set correctly	Passed	24/9/20	Dulshan
57			username check	Testing that when the user logged in changes the user name displayed is correct	The user name state object is set correctly	Passed	25/9/20	Dulshan
58	Employee	Employee Page Information display	First Name check	Testing that when the user logged in changes the first name displayed is correct	The first name state object is set correctly	Passed	20/9/20	Dulshan
59			Last Name check	Testing that when the user logged in changes the last name displayed is correct	The last name state object is set correctly	Passed	21/9/20	Dulshan
60			Email check	Testing that when the user logged in changes the email displayed is correct	The email state object is set correctly	Passed	22/9/20	Dulshan
61			Address check	Testing that when the user logged in changes the address displayed is correct	The address state object is set correctly	Passed	23/9/20	Dulshan
62			Phone Number check	Testing that when the user logged in changes the phone number displayed is correct	The phone number state object is set correctly	Passed	24/9/20	Dulshan
63	AccountEdit	Account edit page details display correctly	First Name check	Testing that when the user edits their details, their existing first name is already entered, and changes affect the state value	The first name state object is set correctly	Passed	22/9/20	Julian
64			Last Name check	Testing that when the user edits their details, their existing last name is already entered, and changes affect the state value	The last name state object is set correctly	Passed	22/9/21	Julian
65			Email check	Testing that when the user edits their details, their existing email is already entered, and changes affect the state value	The email state object is set correctly	Passed	22/9/22	Julian
66			Address check	Testing that when the user edits their details, their existing address is already entered, and changes affect the state value	The address state object is set correctly	Passed	22/9/23	Julian
67			Phone Number check	Testing that when the user edits their details, their existing phone number is already entered, and changes affect the state value	The phone number state object is set correctly	Passed	22/9/24	Julian
68			Logged out user	Test the change details form will not render	Form was not rendered, message appears instead	Passed	22/9/25	Julian
69			Logged in user	Test the change details form will render	Form was rendered	Passed	22/9/26	Julian
70	PasswordChange	Check for user logged in status	Logged out user	Test the change password form will not render	Form was not rendered, message appears instead	Passed	22/9/27	Julian
71			Logged in user	Test the change password form will render	Form was rendered	Passed	22/9/28	Julian