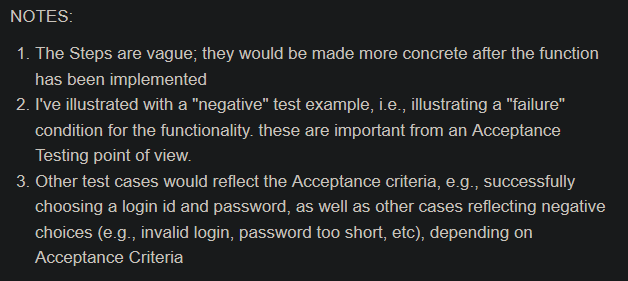
**TEMPLATE/EXAMPLE, LEAVE THIS HERE**

**User Story #0: User Story Name (Group Member Name)**

|  |  |
| --- | --- |
| ID | 0.1 |
| Purpose | Test the Register User with existing username |
| Set up | A user called lcavedon already exists in the system |
| Steps | 1. Navigate to Register New User page  2. Select Register New User  3. Enter lcavendon as username and fill other fields |
| Expected Result | System reports that username already exists; New User not added |

|  |  |
| --- | --- |
| ID | 0.2 |
| Purpose | Test the Register User with existing username |
| Set up | A user called lcavedon already exists in the system |
| Steps | 1. Navigate to Register New User page  2. Select Register New User  3. Enter lcavendon as username and fill other fields |
| Expected Result | System reports that username already exists; New User not added |



**User Story #1: Customer registration (Ryan Babij)**

|  |  |
| --- | --- |
| ID | 1.1 |
| Purpose | Test input validation |
| Set up | Customer clicks “register” and is on the registration page. |
| Steps | 1. Enter invalid input for name (non-name characters) 2. Enter invalid input for email (non-email address) 3. Enter invalid input for password (too short/too long) |
| Expected Result | User should not be registered, and an error message should be presented to the user telling them which input is invalid. |

|  |  |
| --- | --- |
| ID | 1.2 |
| Purpose | Test duplicate registration |
| Set up | Customer clicks “register” and is on the registration page. |
| Steps | 1. Enter a username which already exists 2. Enter valid input for remaining fields |
| Expected Result | User should not be registered, and an error message should indicate that this username already exists. |

**User Story #3: Customer edit bookings (Ryan Babij)**

|  |  |
| --- | --- |
| ID | 3.1 |
| Purpose | Test input validation |
| Set up | Customer is on their bookings page and viewing their bookings |
| Steps | 1. Select an invalid date for the booking |
| Expected Result | This should not be possible, or it should provide an error message indicating that the booking date is invalid. |

|  |  |
| --- | --- |
| ID | 3.2 |
| Purpose | Test booking collision |
| Set up | Customer is on their bookings page and viewing their bookings |
| Steps | 1. Reschedule a booking to a date which is already booked |
| Expected Result | An error message should indicate that the requested date is already booked. The booking should not occur and should rever to the initial booking. |

**User Story #7: User change password (Ryan Babij)**

|  |  |
| --- | --- |
| ID | 7.1 |
| Purpose | Input validation |
| Set up | The user is on the “user details” page and viewing the “change password” section |
| Steps | 1. User inputs invalid password characters into the “new password” field |
| Expected Result | An error message should indicate that the password contains invalid characters. The password should not be updated. |

|  |  |
| --- | --- |
| ID | 7.2 |
| Purpose | Security - Weak password |
| Set up | The user is on the “user details” page and viewing the “change password” section |
| Steps | 1. The user should input a simple password which is easy to guess |
| Expected Result | An error message should indicate that the password is too simple. The password should not update. |

**User Story #9: Customer cancels a booking (Jordan Sorrenti)**

|  |  |
| --- | --- |
| ID | 9.1 |
| Purpose | Test the cancellation of an existing booking when time is more than 48 hours into the future |
| Set up | A customer is logged in  A booking is made for a service that is at a time more than 48 hours in the future |
| Steps | 1. Customer selects the booking  2. Customer selects the option to cancel the booking |
| Expected Result | Customer receives confirmation that booking has been cancelled  Booking has been removed from the system |

|  |  |
| --- | --- |
| ID | 9.2 |
| Purpose | Test the cancellation of an existing booking when time is less than 48 hours into the future |
| Set up | A customer is logged in  A booking is made for a service that is at a time less than 48 hours in the future |
| Steps | 1. Customer selects the booking  2. Customer selects the option to cancel the booking |
| Expected Result | Customer is informed that booking is unable to be cancelled  Booking still remains inside the system |

**User Story #10: Worker views assigned bookings (Jordan Sorrenti)**

|  |  |
| --- | --- |
| ID | 10.1 |
| Purpose | Test that worker can see a booking that is assigned to them |
| Set up | A single booking is made with the worker  That worker with the booking is logged in |
| Steps | 1. Worker goes to view their timetable |
| Expected Result | Worker is shown the booking that they have been assigned to |

|  |  |
| --- | --- |
| ID | 10.2 |
| Purpose | Test that worker cannot see a booking that is assigned to others |
| Set up | A single booking is made with the worker  A worker without any bookings is logged in |
| Steps | 1. Worker goes to view their timetable |
| Expected Result | Worker is not shown any active bookings |

**User Story #11: Customer views worker on time slot (Jordan Sorrenti)**

|  |  |
| --- | --- |
| ID | 11.1 |
| Purpose | Test that customer can see the worker that is allocated to a time slot (potential booking time) |
| Set up | A customer is logged in |
| Steps | 1. Customer clicks on a service  2. Customer selects a time slot for that service |
| Expected Result | Customer is shown the worker that is working at the selected time |

**User Story #12: Home page about/contact us section (Luke Davoli)**

|  |  |
| --- | --- |
| ID | 12.1 |
| Purpose | Ensure that all information relevant to the about/contact us page is present |
| Set up | An individual who is not logged into the service is on the home page of the website |
| Steps | 1. Identify the about section  2. Identify the contact us information |
| Expected Result | The individual is able to locate the relevant information on the page |

**User Story #13: User Login (Luke Davoli)**

|  |  |
| --- | --- |
| ID | 13.1 |
| Purpose | Ensure that login is rejected when user uses an invalid username |
| Set up | An individual who is not logged into the service is on the login page, no user with the username “phil laak” exists |
| Steps | 1. Enter the username “phil\_laak” and any password  2. Click login |
| Expected Result | The login attempt is rejected, and an error message is displayed alerting the user that their username and/or password was incorrect |

|  |  |
| --- | --- |
| ID | 13.2 |
| Purpose | Ensure that login is rejected when user uses an invalid password |
| Set up | An individual who is not logged into the service is on the login page, a user exists in the system with the username “joe\_hachem” and password “quads\_againQQ” |
| Steps | 1. Enter the username “joe\_hachem” and password “quads\_onceJJ”  2. Click login |
| Expected Result | The login attempt is rejected, and an error message is displayed alerting the user that their username and/or password was incorrect |

|  |  |
| --- | --- |
| ID | 13.3 |
| Purpose | Ensure that login is accepted when the user enters a valid username and password |
| Set up | An individual who is not logged into the service is on the login page, a user exists in the system with the username “joe\_hachem” and password “quads\_againQQ”, the user is a customer |
| Steps | 1. Enter the username “joe\_hachem” and password “quads\_againQQ”  2. Click login |
| Expected Result | The login attempt is accepted, and the user is redirected to the appropriate dashboard/page |

**User Story #14: Administrator adds employees (Luke Davoli)**

|  |  |
| --- | --- |
| ID | 14.1 |
| Purpose | Ensure that an administrator can successfully add a new employee to their service |
| Set up | An administrator user is logged in and on their dashboard |
| Steps | 1. Identify the add new employee section  2. Enter valid details for a new employee (expand on this with further info on employee details)  3. Click add employee button |
| Expected Result | The request is posted and the employee now appears in the administrator’s list of employees |

|  |  |
| --- | --- |
| ID | 14.2 |
| Purpose | Ensure that an administrator attempting to add an invalid employee is prevented fromn doing so |
| Set up | An administrator user is logged in and on their dashboard |
| Steps | 1. Identify the add new employee section  2. Enter invalid details for a new employee (expand on this with further info on employee details)  3. Click add employee button |
| Expected Result | The request is not posted, an error message is displayed stating that the input values are not valid (more specific about what is invalid on the page) |

**User Story #15: Administrator view past bookings (Luke Davoli)**

|  |  |
| --- | --- |
| ID | 15.1 |
| Purpose | Ensure that an administrator can view past bookings |
| Set up | An administrator user is logged in and on their dashboard, 2 bookings exist in the system with dates 11/08/2020 and 12/08/2020 at any time |
| Steps | 1. Administrator identifies the past bookings section |
| Expected Result | Both of the bookings in the system appear in the past bookings section |

**User Story #16: Administrator edits business hours times/dates (Luke Davoli)**

|  |  |
| --- | --- |
| ID | 16.1 |
| Purpose | Ensure that an administrator can update the times of the week that their service is available |
| Set up | An administrator user is logged in and on their dashboard, their service is currently available 9am-5pm Monday to Thursday. |
| Steps | 1. Administrator identifies the update business hours section  2. Admin removes Monday from the business hours, adds 9am-5pm Friday and Sunday 11am to 3pm  3. Admin clicks update business hours button |
| Expected Result | The appropriate business hours are shown on the administrator’s dashboard |

**User Story #2: User Logout (Matthies Abera)**

|  |  |
| --- | --- |
| ID | 2.1 |
| Purpose | Test that Users are logged out when clicking the log out button |
| Set up | A user is logged in. |
| Steps | 1. User clicks the logout out button (when logged in).  2. User is redirected to the index page. |
| Expected Result | User is logged out. |

|  |  |
| --- | --- |
| ID | 2.2 |
| Purpose | Ensure that users are still not logged in after clicking log out. |
| Set up | A user is logged in. |
| Steps | 1. User clicks the logout out button (when logged in).  2. User is redirected to the index page. |
| Expected Result | User cannot go backwards of browser history to become logged in again. |

**User Story #5: Customer Available Time Slots (Matthies Abera)**

|  |  |
| --- | --- |
| ID | 5.1 |
| Purpose | Test that registered customers can view the available time slots for any service they wish to possibly book. |
| Set up | A customer logged in and, on their dashboard. |
| Steps | 1. Customer clicks on the service they are interested in  2. Redirected to service page. |
| Expected Result | Customer is shown a list of available times for the specific service. |

|  |  |
| --- | --- |
| ID | 5.2 |
| Purpose | Test that unavailable times are crossed out. |
| Set up | A customer logged in and, on their dashboard. |
| Steps | 1. Customer clicks on the service they are interested in  2. Redirected to service page. |
| Expected Result | Unavailable times of services are crossed out; to establish they have been booked already. |

**User Story #4: Customer View Services (Matthies Abera)**

|  |  |
| --- | --- |
| ID | 4.1 |
| Purpose | Test the registered customers can view list of services available. |
| Set up | Customer is a registered user. |
| Steps | 1. Customer logs into their account. |
| Expected Result | Customer is shown the list of available services. |

**User Story #21: Customer views upcoming bookings (Hollie Steinman)**

|  |  |
| --- | --- |
| ID | 21.1 |
| Purpose | Tests that a customer can see a booking |
| Set Up | A customer account with a booking made |
| Step | 1. Navigate to Customer Details page |
|  | 2. Find upcoming bookings card |
| Expected Result | The future booking is displayed |

|  |  |
| --- | --- |
| ID | 21.2 |
| Purpose | Tests a customer doesn’t see a past booking within the current bookings |
| Set Up | A customer account with a past booking |
| Step | 1. Navigate to Customer Details page |
|  | 2. Find upcoming bookings card |
| Expected Result | Past booking is not displayed |

|  |  |
| --- | --- |
| ID | 21.3 |
| Purpose | Tests a customer can see a future booking’s time and date |
| Set Up | A customer account with a booking made |
| Step | 1. Navigate to Customer Details page |
|  | 2. Find upcoming bookings card |
| Expected Result | The future booking displays the time |

**User Story #19: Administrator view current bookings (Hollie Steinman)**

|  |  |
| --- | --- |
| ID | 19.1 |
| Purpose | Tests that an administrator can see a booking |
| Set Up | An administrator account with a booking made to the service |
| Step | 1. Navigate to Administrator page |
|  | 2. Find upcoming bookings card |
| Expected Result | The future booking for the service is displayed |

|  |  |
| --- | --- |
| ID | 19.2 |
| Purpose | Tests that an administrator doesn’t see a past booking within the current bookings |
| Set Up | An administrator account with a fulfilled (past) booking made to the service |
| Step | 1. Navigate to Administrator page |
|  | 2. Find upcoming bookings card |
| Expected Result | The past booking is not displayed |

|  |  |
| --- | --- |
| ID | 19.3 |
| Purpose | Tests an administrator can see a future booking’s time and date |
| Set Up | An administrator account with multiple bookings made to the service |
| Step | 1. Navigate to Administrator page |
|  | 2. Find upcoming bookings card |
| Expected Result | The time and dates of future bookings are displayed |

**User Story #17: Customer update information (Hollie Steinman)**

|  |  |
| --- | --- |
| ID | 17.1 |
| Purpose | Tests that a customer can change their details |
| Set Up | A customer account with an email [hello@gmail.com](mailto:hello@gmail.com) |
| Step | 1. Navigate to Customer page |
|  | 2. Find customer details card |
|  | 3. Enter in new email, [hello1@gmail.com](mailto:hello1@gmail.com) |
|  | 4. Click on save changes button |
| Expected Result | The email is updated within the system |

|  |  |
| --- | --- |
| ID | 17.2 |
| Purpose | Tests validation on customers changing their details |
| Set Up | A customer account with a phone number 0412 345 678 |
| Step | 1. Navigate to Customer page |
|  | 2. Find customer details card |
|  | 3. Enter in new phone address, abcd |
|  | 4. Click on save changes button |
| Expected Result | An error displaying that it is not a valid format, new details are not saved to the system |

|  |  |
| --- | --- |
| ID | 17.3 |
| Purpose | Tests a user cannot enter the same details as they previously had |
| Set Up | A customer account with a phone number 0412 345 678 |
| Step | 1. Navigate to Customer page |
|  | 2. Find customer details card |
|  | 3. Enter in new phone address, 0412 345 678 |
|  | 4. Click on save changes button |
| Expected Result | An error displaying that it is the same as the phone number already supplied, new details are not saved to the system |

|  |  |
| --- | --- |
| ID | 17.4 |
| Purpose | Tests a user cannot update their details if left blank |
| Set Up | A customer account |
| Step | 1. Navigate to Customer page |
|  | 2. Find customer details card |
|  | 3. Click on save changes button |
| Expected Result | An error displaying that all fields are blank, no changes made in the system |

**User Story #20: Administrator views booking contact details (Hollie Steinman)**

|  |  |
| --- | --- |
| ID | 20.1 |
| Purpose | Tests that an administrator can view customer details on a specific booking |
| Set Up | An administrator account with multiple bookings |
| Step | 1. Navigate to Administrator page |
|  | 2. Find future bookings card |
|  | 3. Click on a booking |
| Expected Result | The customer’s details such as their name and phone number are displayed |