Actual Planner

# Test Planner – What logic to be tested

The test plans for the login situation is divided into three possible segments: one is correct details; one is incorrect password against the username and the final one is a username which the system does not have a track of currently. This is done by using the username as part of the search query for the database, which would lead to two possible outcomes. Either a match which would return a password and a null value. If the null value is return, the username does not exist. If the password returns, we compare with password to see if the values match, which will determine whether the user can login or not.

The register test plans incorporate five different segments: one where all the inputs are correct, one where there is a singular field that has missing data, one where the password does not meet the credentials, one where the confirm password does not match the original password and the last test where the inputted username already exists inside the system. The tests can be categorised by inputted all the fields except the password and confirm password, including a used username. Then the password that is weak. Then a confirm password field that does not match the original password. Then ensuring the username already exists. Then a completely new username.

The logging of logging page buttons comes in four main formats. The first one is the cancel, which would cancel any changes that were not saved during that period of session. The discard log would discard all logged inputs that day, included previous saved ones. The Save button would save each periodically change the user added to the logging system. The Submit would submit all the changes to the database as it is the method that allows the user to log their diet every day.

During the aspect where we are adding a new log to the schedule, there are several different segments: unsuccessfully querying the API, successfully search the API and click on a result, Return to search for a different food. Modify the recipe to make it a homemade version (by adding other ingredients to it) and finally add the recipe, whether it has been modified or not.

View the different parts of the Advisor System. This is from connecting to the different pages for the different advice on the different advices. A generic overview would be displayed followed by three buttons that goes into more specific details. The underdone button goes into the details that have been under consumed by the user. The overdone leads to the page that displays the overdone aspects. The Advise button goes to an more in-depth analyse of the dietary recommendations for the user. The Daily status and Weekly status graph have a clickable that leads to a bigger version of their respective graphs, which shows the trends of the logged data for the user. The return button just returns to the main page. The “Test Food Input” is a way for the user to use sample input to see what the result would be if they would schedule input for the diet for mass amount of people.

The View Account details button should allow the user to access their details as a viewable method. The same page allows them to have the ability to edit and update their details if they wish to do so, except their username.

# Evaluation Planner – What parts of the application is to be evaluated

The evaluation would be split into several different areas and questionnaires. The first would be based on the user evaluation through mapping the technical terms of the mobile user ability to terminologies which can be understood by non-technical people, incorporating a metric rating system. The methods to do this would incorporate a number of steps. The first would be taking the platform usability criteria and finding out how usable the application was initially, from the functionality itself and the application design comprehension equating to the users understanding. The next part would be assuring the application provides a form of immediate value to the user to pass the “Provide Value Right Away” criteria, such as the logging aspect or the advisor system itself. Ensuring the navigation is simple alongside concise, specific content to ensure the number of steps required to complete each task and to reduce scrolling methodologies. These steps alone would provide a comprehension evaluation.

After mobile usability evaluation, the 10 heuristics will be mapped next to non – technical terms as well. Within this area, a rating metrics system would be employed in combination to the textual evaluation. The first criteria would be “Visibility of System Status”, which will check if the user knows when the system is properly interacting with them or not through the different stages. An example would be searching the API and knowing the user will get data back and they need to be patient. The next step would be “Match between system & the real world” in terms of expectations, to gather the user’s details and compare to what the developers thought the users wanted. A good example would be the correct data being logged and displayed to the user.

After that, “User control & freedom”, “Consistency & standards” and “Error prevention” would come after, which the evaluator can separate each section in their description or combine all three into one overall evaluation. This is due to all three parts having impacts on each other in NDMA’s case and part of an expectation for users inside NDMA. The next criteria would be the combination of “Recognition rather than recall”, “Flexibility & efficiency of use” and “Aesthetic & minimalist design” into one overall evaluation, as they relate to one another in terms of the design for the user interface. All three sections would impact the user decision to return to the application for continual use, combined with an overall intuitive, sleek look.

Completing that part of the evaluation would lead to the criteria of “Help users recognize, diagnose, and recover from errors” within the application to ensure the user can handle the applications overall errors, such as potential api or server errors. If there is anything additional the user needs access to, that it is clear to get access to such “Help & documentation” in some shape or form, such as access to the applications documentation or a developer’s email address etc.

Using analytics tools as part of the application can really measure the details of the applications, such as the speed when connecting to the remote servers for the applications. The same tools would also test the speed of the applications default loading times and getting from page to page. Such tests can be completed using singular people and multiple people, especially to see the similarities and difference on performance and results for evaluation purposes. Two examples would be accessing the user credentials from the cloud database and see the speed of the results come with singular and multiple users.