Development Testing

#### **Strategy and Approach**

Following the test-driven development outline, we will draw from our user stories, client comments and project requirements to create the tests. This will allow us to order the development of the different features of the app, based upon which parts depend on each other. For each of the app features the group has discussed in detail with the client what the feature should achieve and how it should take information from the user. All information should be displayed in a user-friendly format. This format will be achieved by showing the app to our friends and family throughout our development process and seeing how well they interact with it.

Our app will have quite a lot of screens as well as a group of people working on it at the same time. As a result, it is important to ensure we are not accidently making changes to screens that will then adversely affect other group members changes. Using the renderer module from the react-test-renderer package we take a ‘snapshot’ of how the screen should look. Whenever any updates are uploaded to GitHub these snapshots are used to check for any unexpected changes. The tests can be re-run in our terminals to update these snapshots when intentional changes are made.

Throughout the development process, we are using Expo to create and simulate our react native app. The Expo app can be downloaded on our phones and with this we can simulate how our app would run on an actual mobile device. This is incredibly helpful to check the formatting because it works on both iOS and Android devices. Expo will also enable us to show our client the progress we are making and allow them to interact with the app.

#### **Core Component**

A core component of our system (as specified by our client) is the feature that allows the user to take a picture of their skin lesion/mole. They will have diary entries for moles that they have already begun tracking and will have the ability to add new diary entries for new moles. In this test description we will focus on the scenario where the user is adding new images to an existing diary entry. In order to help the user track their moles they must be able to take periodic photos so that a doctor can look for changes.

|  |  |
| --- | --- |
| **Test** | **Explanation** |
| The middle button on the camera page bar takes a picture that matches what the user can see on their screen. | When this button is pressed a picture is taken and held temporarily. After pressing the button, the user is taken to the review page to check that image. |
| The camera flip button when pressed changes the camera view from back camera to front camera and vice versa. | Access to both cameras is helpful because it means that for users who do not have any help, some pictures will be more easily taken with the front camera. This button is available on the camera bar on both the far shot and near shot page. |
| If the user rejects an image after taking it this image is not saved and they cannot see it in their diary screen. | The diary page displays all the users saved images. If the image is rejected the image should not be added to the database therefore it will not show up in the diary screen. |
| The ghost images for both the near shot and the far shot are translucent. | The user still needs to be able to see what camera is showing behind the ghost image. The ghost image should also be toggled on and off with the ghost button. The ghost images have 50% opacity. |
| The relevant ghost image is shown to the user. This is based on what existing entry they have selected. | For an existing entry update the user only needs to take a new near shot. The near shot ghost image should show the first near shot the user took of that entry (therefore there is no ghost image for new entries). |
| On the final screen of the picture taking process the user has the option of scheduling a notification. This notification will trigger 30 days later, reminding them to take a new picture. | For testing we set this time frame to be a much shorter trigger so that we can see it in action. |
| When the user adds a new mole entry, they should be taken through the process of taking a far shot and a near shot. When the user is updating an existing entry, they should only be able to take a near shot. | The far shot is import for clarifying where the mole is on the body, therefore this only needs to be taken once. The near shots are important for assessing any changes in the mole. |
| The following screens will have a back button in the top left: choose mole, near shot camera. | No issues are caused if the user decides they want to go back to the previous screen from this list. The back button has been removed from the other parts of the picture taking process to avoid any issues with saving the images in the database. |