**Project Title: Cleft Lip Aesthetics Tool**

3rd October 2016 – 9th October 2016

Goals of the project established through a meeting and a one-page summary written up regarding the project description and goals. Need to investigate possible multi-platform technologies to create the app in. Work out a method in which to draw around the lip regions. Set up back up of all work produced.

10th October 2016 – 16th October 2016

Ask about whether I need to set up the cloud based side or if it will be given to me. Discuss ideas on how to draw the lip regions. Clarify the possibilities of what the operating system will be of the users of the app. What is the theme of the app? UCL themes are available.

Native application makes updates a hassle because each user would have to manually do this. Web applications are automatically updated.

Use of HTML canvas to have a background of the patient’s image. Found a plugin which allows the user to draw on this canvas. Multiple drawings can be carried out with these able to be shown on top of each other. I feel drawing the lip regions on a small phone would not be practical due to the precise nature of the work. Not using a pen but instead using your finger would probably make this worse. I think ideally you would want to use a mouse or a pen on a larger device such as a tablet.

<http://www.williammalone.com/articles/create-html5-canvas-javascript-drawing-app/>

<http://intridea.github.io/sketch.js/> Can save sketch easily

Process the drawings so that it ends up smoother. Less jagged edges. Button to make smooth. Upload both versions to cloud. Control points, straight lines or Bezier, could have requirements or Should have. Better for smaller screens with zoom in features.

I set up cloud based part, images that someone else uploads already on there. I upload lip drawings and symmetry scores to cloud.

No specific theme. Ensure clean interface, look into Android’s clean theme recommendations.

Use cases and then MoSCoW style requirements, sequence diagrams, 3 sentence summary, user stories. Clinicians and researchers. Researchers focus on data and analytics. Do by next week.

Web application decided.

17th October 2016 – 23rd October 2016

Default page is local data with link to cloud on top. Sync feature available here.

Cloud: view images and their drawings and symmetry scores for each. Download specific images with optional drawings and symmetry downloads.

Local: View locally stored images and if already produced, their drawings and symmetry scores. Sync content with cloud.

Local Image: Create drawing, view drawings and symmetry scores.

Syncing tool needs to be researched further to see exactly how it will be done. Investigate how cloud will be set up. Project plan for the remainder of the project. Gantt chart. MoSCoW style requirements. Mock ups.

24th October 2016 – 30th October 2016

Potentially use Cordova web storage for local data storage. Otherwise the whole app will have to be online.

App for phones and computer distributed separately negate need for login page.

Offline web pages except for the cloud part. Dropbox for uploaded images with specific names? File types of predictions and drawings.

Azure, uploading images outside scope. Lip drawings are PNG format. 0 not lip, 1 corresponds to lip. The more complicated would be JSON. Predictions are PNG drawing to JSON with data and symmetry score.

Set up GitHub, finish mock ups, link requirements to Gantt chart, do project plan and email. UML diagram

31st October 2016 – 6th November 2016

Add Harry to GitHub. [**https://app.moqups.com/farbasmiah@gmail.com/q0nfEJ6QiS/view**](https://app.moqups.com/farbasmiah@gmail.com/q0nfEJ6QiS/view)

Look into cloud side with azure or another storage.

Create UI pages and produce APK file. Push code to GitHub.

Push documentation to GitHub.

Initial UML class diagram.

7th November 2016 – 13th November 2016

Created Phone Gap project with screens and design/theme set out. UML activity diagram created.

14th November 2016 – 20th November 2016

Complete the rest of the screens so that all pages have been created. Buggy drawing feature created that needs to be fixed.

21st November 2016 – 27th November 2016

Server set up so cloud images obtained from server with working downloads on Android. Upload feature is also set up but has not been linked with sync yet.

Drawing feature also working.

28th November 2016 – 4th December 2016

Save drawing locally. Upload drawings to cloud with sync. Set up dummy symmetry scores feature.

Investigate .png 1s and 0s and imagej.

Downloaded image inserts data into local sqlite database for local. Major issues with displaying local features with downloaded images.

5th December 2016 – 11th December 2016

Save images into sqlite database as blob.

Issues fixed regarding local features. ID set up for databases. Drawing image size fix.

Issues with sqlite function calling.

12th December 2016 – 18th December 2016

Fix sqlite, only allow unique downloads, fix drawing lines, minor fixes like this. Bitimage.

Write function definition, parameters etc (bitimage). What is expected as output. Output as percentage of twos. Email this to Harry.

19th December 2016 – 8th January 2017

DB issues partially fixed?

9th January 2017 – 15th January 2017

No duplicate downloads allowed. Downloaded images in sqlite. Minor fixes like local page refreshing, viewing images fix. Database issues fixed. Drawings saved in database and have unique ID. Deleting images delete drawings too. View drawings page shows selected image’s drawings and allows delete, sync, view image and drawing. Drawings uploaded to cloud database.

16th January 2017 – 22nd January 2017

Look into iOS features. Finish up everything else except for the drawing and score aspects. Do interim report. MoSCoW non-functional, user acceptance has users test the app. Ranked 1-5 to see how easy to use app is. Start the dissertation document.

Fixed syncing individual drawings to cloud. Syncing for all drawings to a specific image complete. Interim report complete. Online SQL database used to obtain data for cloud images.

23rd January 2017 – 29th January 2017

Begin to adjust code for iOS applicability. Minor improvements on app. Final project abstract and background information written up. Possibly begin advanced drawing features.

Can’t do iOS without a physical device. Ask about symmetry score function. Fixed downloading drawings by encoding base64.

Dave Twisleton iOS device, set up an Azure account and email Harry.

30th January 2017 – 12th February 2017

Drawing feature, report.

Ask about iOS developer key.

PhoneGap build install feature fixed. Fixed navigation drawer bug. Symmetry score plugin to be created for iOS and Android.

Images to be square.

13th February 2017 – 26th February 2017

Create desktop app on Windows instead of iOS. Save image with editable Bezier curves, possibly as SVG. Write background of report.

Development testing

- Unit testing

- Component testing

- Validation testing for each requirement

27th February 2017 – 19th March 2017

Created symmetry score feature in Java. Get image RGB data for every pixel, make a 2d array of 1s for each pixel with drawings and 0s for each pixel without drawings. Half the 2d array into a left and right section of the image. Alter the right section’s array to make it a mirror image. Create a new 2d array with the sum of the left and right array’s, values for each pixel. Overlap regions will have a value of 2. Obtain a symmetry score percentage by calculating the percentage of 2s in the array compared to 1s.

Converted the Java code to JavaScript. Issue with canvas being tainted by cross-origin data, security issue, fixed by using base64.

App now gives symmetry score for each drawing straight away.

Splitting an array of odd width would cause an issue but this was fixed by always making the initial width an even value.

Ask about handover info.

20th March 2017 – 16th April 2017

Write up, small fixes and testing.