Cleft Lip Aesthetics Tool

A cross-platform app that allows the success of a cleft palate surgery to be determined through symmetry

Mohammed Farbas Miah

BSc Computer Science

Submission Date: 2nd May 2017

Supervisor: Dr Harry Strange

This report is submitted as part requirement for the BSc Degree in Computer Science at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

# Abstract

The project is about determining the success of cleft lip and palate surgeries. Using a mobile app, paediatric plastic surgeons should be able to evaluate the aesthetic outcome of the surgery by determining how symmetrical the lips are. The user should be able to draw around the lip region of the target image and then receive a symmetry score, determining the successfulness of the surgery. This would replace the previous, subjective method of having a panel of people determine success. Having the app be multi-platform would be ideal which meant PhoneGap was used for development. There should also be offline functionality which means the app will be synced to an online server to allow data to be transferred as required.

The app was developed with PhoneGap to allow multiplatform functionality to be implemented quicker. The user interface was developed with focus on ease of access. Once all the pages were set up, the app was linked to a server which allow downloading and uploading of drawings and images to the online database. The drawing feature was then implemented and ImageJ was used to produce a symmetry score based off the drawing.

The goals of the project have successfully been reached which means a multiplatform app with offline features has been produced which allows users to create a drawing based off a patient’s drawings and receive a symmetry score. These drawings can be uploaded to the cloud for future analytical work by researchers to allow further improvements to the app. A user manual was produced to demonstrate the app’s features in a simple manner. A system manual has also been produced regarding the app code to assist in further development.

# Contents

[Abstract 2](#_Toc474962192)

[Contents 3](#_Toc474962193)

[1. Introduction 5](#_Toc474962194)

[1.1. Problem 5](#_Toc474962195)

[1.2. Aims & goals 5](#_Toc474962196)

[1.3. Project overview 5](#_Toc474962197)

[1.4. Report overview 6](#_Toc474962198)

[2. Context 7](#_Toc474962199)

[2.1. Background information 7](#_Toc474962200)

[2.2. Related work 7](#_Toc474962201)

[2.3. Research done 7](#_Toc474962202)

[2.4. Tools & software 7](#_Toc474962203)

[3. Requirements & Analysis 7](#_Toc474962204)

[3.1. Problem statement 7](#_Toc474962205)

[3.2. MoSCoW style requirements 7](#_Toc474962206)

[3.3. Use cases 7](#_Toc474962207)

[3.4. Requirements analysis 7](#_Toc474962208)

[4. Design & Implementation 7](#_Toc474962209)

[4.1. Application architecture 7](#_Toc474962210)

[4.2. User interface 8](#_Toc474962211)

[4.3. Drawings & scores 8](#_Toc474962212)

[4.4. Local data 8](#_Toc474962213)

[4.5. Cloud data 8](#_Toc474962214)

[4.6. Local storage 8](#_Toc474962215)

[4.7. Cloud storage 8](#_Toc474962216)

[4.8. Multiplatform functionality 8](#_Toc474962217)

[5. Results Evaluation 8](#_Toc474962218)

[5.1. Testing 8](#_Toc474962219)

[6. Conclusions 8](#_Toc474962220)

[6.1. Summary of achievement 8](#_Toc474962221)

[6.2. Critical evaluation 8](#_Toc474962222)

[6.3. Future work 9](#_Toc474962223)

[6.4. Final thoughts 9](#_Toc474962224)

[7. Bibliography 9](#_Toc474962225)

[7.1. References 9](#_Toc474962226)

[8. Appendices 9](#_Toc474962227)

[8.1. System manual 9](#_Toc474962228)

[8.2. User manual 9](#_Toc474962229)

[8.3. Supporting documentation 9](#_Toc474962230)

[8.4. Test results 9](#_Toc474962231)

[8.5. Evaluation data 9](#_Toc474962232)

[8.6. Project plan 9](#_Toc474962233)

[8.7. Interim report 9](#_Toc474962234)

[8.8. Code listing 10](#_Toc474962235)

# Introduction

## Problem

Children born with a cleft lip and palate often have surgery at a young age to fix the problems associated with the condition. One of the improvements on the patient, post-surgery would be the aesthetic improvement to the lip region. To determine the success of this procedure, a panel is used to determine how aesthetic the post-operation lip region is. This is quite subjective and requires multiple people which is inefficient. This project aims to develop an app that can use symmetry to determine the success of a surgery to replace the current method being used. This would save a lot of time and provide faster and more accurate results once surgery has been completed.

## Aims & goals

The successful completion of this project requires multiple key goals to be fulfilled which have been listed below.

1. A fully functional multi-platform application that determines the symmetry of the lips from a patient’s image to determine the success of the surgery.
2. Full and clear documentation of the app for future development.
3. Concise user manual to demonstrate how to use the app.
4. An intuitive method of drawing around lip regions of a patient’s image to produce a trace for determining success.
5. Integrated cloud connectivity with syncing and offline features for the app.
6. A method of determining the symmetry score of a drawing created by the user.
7. A final report detailing the planning, methodology and results of the project.

To accomplish each of these deliverables, the aims listed below need to be achieved first.

1. Understanding of PhoneGap and the programming languages required to develop an app that is functional on multiple platforms. The bulk of the design is created in HTML and CSS with JavaScript being used for scripting. Individual platforms also require their corresponding native programming language.
2. Knowledge of good coding practice to allow for code to be clearly presented and understood easily by other developers.
3. Understanding of the programming languages and tools required to set up a server and database system for app connectivity. SQL and PHP are used in this project to achieve this.

## Project overview

The project was carried out in distinct sections, relating to the aims. The first section related to research and background knowledge to determine a clear set of requirements for the project and to determine the approach of development. This is documented in chapters 2 and 3.

The next section relates to the development portion which was split further into sub-sections. Offline and Online portions of the app were developed individually and then linked together. The main drawing and symmetry score feature was then implemented. Finally, the multiplatform functionality was enabled by adding additional code to allow for iOS and Android compatibility.

Lastly, evaluation and documentation was carried out. This involved testing the app and logging issues and future ideas for development. Documentation produced includes a user and system manual for further guidance to future app users and developers, respectively.

## Report overview

Chapter 1 – Introduction

Provides an outline of the problem being worked on and what the challenges goals are.

Chapter 2 – Context

Details background information on the subject matter, going over previous work and research carried out for the project. The tools and software used to develop the app shall be explained here.

Chapter 3 – Requirements & Analysis

Provides a clear and structured set of requirements for the project. Expectations for the final product are established.

Chapter 4 – Design & Implementation

Shows how each piece of the system architecture has been implemented and how they are linked together. The reasoning behind specific design decisions are given here.

Chapter 5 – Results Evaluation

Details the testing approach with results and specifics for the app that has been developed.

Chapter 6 – Conclusions

This chapter gives a summary of what the project has achieved with a critical analysis. Ideas for future work on the project and final thoughts are given.

Chapter 7 – Bibliography

A list of sources that were made use of during the project which have not been referenced in the text.

Chapter 8 – Appendices

The appendices provide additional relevant information that have not been included in the other chapters. This includes manuals, documentation produced throughout the course of the project and code samples.

# Context

## Background information

To determine the success of a cleft lip and palate surgery, a panel is required to visually judge whether the outcome has been successful. The aim of this project is to produce a means of replacing the current method with an app that can be used to judge the symmetry of a post-surgery image. The reasoning behind this is due to the flaws and inefficiencies in the current method. Currently, multiple people are required for each post-surgery panel which wastes time and resources. The judgement of success being left to people to decide is subjective as everyone has different preferences and ideal facial structures. In using a panel to determine the outcome, there would be a large delay between the time of the surgery and the judgement.

From creating an app to determine the success of a cleft lip and palate surgery, the above inefficiencies can be bypassed, therefore allowing for a quicker, more accurate and cheaper solution. In the past, attempts have been made to seek solutions to the given problem. Research on these are discussed below.

## Related work

In the early stages of the project, similar work that have been carried out on the topic were reviewed. From doing so, a better understanding of the need for a solution was gained. The issues of the current method for determining success of a surgery were also better understood.

The first research paper being reviewed discusses the applicability of an Index that had been developed to determine the successfulness of a unilateral cleft lip and palate surgery for five year olds. [1] This Index is required to help determine success at a fast rate instead of waiting until the patient was older to determine the outcome. In doing so, surgeons would can relate minor changes in the procedure to the outcome of a surgery. The paper only covers unilateral cleft lips which excludes the more severe case of bilateral cleft lips. However, in the case of unilateral cleft lips, the paper concluded the Index was the fastest indicator of success of a surgery.

A computer program to determine lip symmetry post-surgery is discussed in another paper. SymNose compares the patient’s lip regions to a control group who have never had a cleft lip. [2] The study only seeks to compare unilateral cleft lips and excludes bilateral cleft lips. Usage of the application was carried out by letting the user draw around the lip region of a patient. To determine symmetry, the percentage mismatch was calculated from overlaying the lip regions. The results demonstrated that the use of SymNose was a good tool as a quantitative assessment of success of surgery. In addition, this application presents an efficient and cheap solution to the problem at hand.

In a follow-up paper, the use of SymNose for comparison of bilateral cleft lip patients are examined. [3] Although the sample size was small at 15, the results supported the usage of SymNose for assessing the symmetry of bilateral cleft lip patients. Post-surgery, a significant level of asymmetry occurred for bilateral patients’ lips whereas the nose region proved to be more symmetrical. This contradicts the original panel assessments and therefore suggests nose symmetry should play a part in determining the success of a surgery.

From reviewing the above three papers, an outline of the requirements for the app were obtained. It would be important to allow the determination of success for both unilateral and bilateral cleft lip patients. The results should be obtainable quickly to allow the analysis of results to occur faster, therefore letting surgeons know how minor variations in methodology affects a patient’s facial symmetry. Use of nose regions provides an additional source of comparison to determine success, especially for bilateral cleft lip patients. The SymNose application is not publicly available which meant an examination of the program was not possible. In producing an open source app, future work to advance the project would be much simpler by allowing the problem at hand to be solved and improved upon much faster.

## Research done

Before beginning development for a multi-platform app, it was important to know what skills would be required beforehand. Therefore, preparatory learning of some subjects took place before the development phase. Once the decision was mad to develop a multi-platform app, research was undertaken to determine which pieces of software would be used to carry this out. Additionally, during the development phase, further research took place to determine the best solution to problems and to assist in solving them. The main resources used for research and understanding aspects of the project are listed below.

**Stack overflow**

Throughout the course of the project, multiple minor issues arose resulting in code not working in the expected manner or a fix to a certain problem could not be identified. These are problems that have occurred to other people in the past that have been solved already. Stack Overflow [4] is a Q&A style forum with topics based on computer programming that can help to answer a large amount of the problems that occurred during this project. Using Stack Overflow, many issues have been solved through searching for related problems.

**PhoneGap and Cordova**

Once it was decided to produce a multi-platform app with a single set of code, PhoneGap [5] was chosen as the best solution. PhoneGap is an open source development framework for mobiles that helps to build multi-platform applications with a single set of code in HTML, CSS and JavaScript. Making use of Apache Cordova [6], it is simple to build and test apps on multiple platforms including Windows, iOS and Android. Documentation for both PhoneGap and Cordova was constantly referred to during development.

**Cordova plugins**

When creating apps that are not native, they would presumably be limited by the capabilities of web apps. However, plugins in Cordova allow for the app to carry out native functionalities such as storage or camera usage. Specific native features were required for this project which would have to be implemented using plugins. npmJS [7] was repeatedly used to search for plugins to use for the app.

**Programming tutorial sites**

To code the app, HTML, CSS and JavaScript would be used for the front-end side. These languages had been partially learnt prior to commencing the project so there was not a need to learn these from scratch. Nonetheless, multiple aspects needed to be brushed up on or referenced which was done by using W3Schools [8]. W3Schools is a web development learning site which provides tutorials, references and examples for HTML, CSS and JavaScript. For the back-end, PHP and MySQL was to be used which are also taught on W3Schools.

Being new to development with PhoneGap, tutorials for example apps were viewed to gain a better understanding of how aspects of the technology worked. PhoneGapPro [9] is a tutorial site that demonstrates how functionalities of PhoneGap can be implemented. From viewing the multiple tutorial apps available on the site, the applications of specific parts of PhoneGap were understood. These tutorials included usage of key plugins such as device storage and the application of PHP with MySQL databases.

## Tools & software

Various software, library code, frameworks and other tools were used in the development of the app. Each of these are listed below with the reason for selection and what it was used for outlined.

**GitHub**

As a means of version control for source code, GitHub [10] was used. GitHub allows all previous uploads of code to be viewable from multiple computers. This also worked as a means of backing up the content produced over the course of the project. A private repository was created for free using a student account. This software was chosen over alternatives due to previous experience using GitHub, resulting in familiarity with the features.

**PhoneGap**

PhoneGap has multiple features that were made use of during development of the app. The key feature of this framework is to convert web page code into an app without having to rely on native platform-specific APIs. To create an initial code template for the PhoneGap app, the PhoneGap CLI [11] was used. This included all the relevant files and plugins to set up an app using HTML, CSS and JavaScript. The PhoneGap CLI allowed for various functionalities to be used which include the building of the app file, such as an APK for Android. It also allowed for quick testing of the app by building and running it directly onto a mobile device.

To implement native functionalities of a platform, plugins were used. A specific plugin was implemented to allow for SQLite databases [12] to be created and edited which would otherwise have not been possible to do. A file transfer plugin [13] was used to allow for files to be downloaded onto a device.

**Android SDK**

The Android Software Development Kit (SDK) [14] contains a comprehensive set of tools required for development on the Android platform. The SDK is a requirement to allow for APK packages to be created. When an Android app is built using PhoneGap, the Android SDK is required.

**Windows 10 SDK**

The Windows Software Development Kit for Windows 10 [15] which comes integrated with Visual Studio 2015 was also used. This SDK is a requirement when creating any Windows 10 app. This SDK is required when building a Windows 10 app with PhoneGap. Windows 10 is unique in that an AppX package can target all device families in the Windows 10 line-up.

**Font Awesome**

Font Awesome [16] is a font and icon toolkit which provides over 600 different scalable vector icons. It is the second most popular third-party Font Script with a 20% market share. This toolkit was used to provide visual icons of specific commands in the app, such as drawing icon in front of the create a drawing feature.

**Paper.js**

Paper.js [17] is a vector graphics scripting framework which makes use of the HTML Canvas. It provides a range of functionalities to create and edit drawings, making use of bezier curves and control points. This framework was used in the development of the app to create the drawing feature which would allow the user to draw the lip regions of a patient so that a symmetry score can be generated.

**Snap.js**

Snap.js [18] is a simple library for creating a navigation drawer with JavaScript. It provides multiple features to improve user experience such as the ability to slide the drawer with a flick of the finger. This library was used as part of the core user interface of the app, providing the navigation drawer which is used in most pages of the app.

**jQuery**

jQuery [19] is JavaScript library which makes multiple aspects of development simpler with an API that is easy to use. Some major features are its event handling, animation and Ajax functionalities. jQuery is the most widely used JavaScript library and is therefore supported on many platforms. For this app, it was used in multiple features with the main aspects being its Ajax and event handling implementation.

# Requirements & Analysis

## Detailed problem statement

To deliver a high quality and robust product, the problem detailed in chapter 1 was expanded upon. This was done through multiple methods which included identifying the user’s needs, comparison with previous research and discussions with the project supervisor.

….

User stories:

As a researcher I want to easily be able to view relevant input and output data for images so that I can analyse the results without much hassle.

As a researcher I want to be able to compare relevant data for the same image so that I can determine how a specific change in the input changed the output result.

As a clinician I want to be able to use the app without having to spend a long time learning how to use it so that I can save time.

As a clinician I want to obtain symmetry scores on a patient’s image quickly without having to go through large amounts of options so that I can use the app without much hassle.

## Structured list of requirements

Based off the detailed problem statement, a structured set of requirements were produced. These were produced in the MoSCoW style which details the priority levels of each individual requirement through Must have, Should have, Could have and Would have. As the project progressed in the early stages, these were refined slightly which resulted in the final 16 requirements, categorised by its activity, as shown in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Details | Type | Activity | Priority |
| RQ1 | The app shall let the user sync their data for each image with the cloud so new drawings and symmetry scores are uploaded. | Functional | Cloud | Must have |
| RQ2 | The app shall be connected to a cloud based server in which images are stored. | Functional | Cloud | Must have |
| RQ3 | The app shall have an intuitive method of allowing users to draw around the lip regions of images stored locally. | Functional | Drawings | Must have |
| RQ4 | The app shall determine symmetry scores from the user’s drawings and display them to determine success of surgery. | Functional | Processing | Must have |
| RQ5 | The app shall allow users to view corresponding lip drawings and symmetry scores for each image on the cloud. | Functional | User interface | Must have |
| RQ6 | The app shall allow users to view lip drawings and symmetry scores for each image stored locally. | Functional | User interface | Must have |
| RQ7 | The app shall display a list of all images stored locally with options to draw lip regions and view previous drawings and symmetry scores. | Functional | User interface | Must have |
| RQ8 | The app shall require minimal training to use in a clinical setting. | Non-functional | Accessibility | Must have |
| RQ9 | The app shall have a clean interface based on Android guidelines. | Non-functional | User interface | Must have |
| RQ10 | The app shall be capable of running on multiple platforms. | Functional | Accessibility | Should have |
| RQ11 | The app shall allow users to download images locally from the cloud. | Functional | Cloud | Should have |
| RQ12 | The app shall be built in a manner so that other people can build upon it easily due to good coding practices being followed. | Non-functional | Development | Should have |
| RQ13 | The app shall process the user’s drawings to make them smoother. | Functional | Drawings | Could have |
| RQ14 | The app shall upload the original and smoother drawing to the cloud. | Functional | Cloud | Could have |
| RQ15 | The app shall have control points for drawing around the lip regions with a zoom in feature for further precision. | Functional | Drawings | Could have |
| RQ16 | The app shall have offline functionality. | Functional | Accessibility | Could have |

## Use cases

Use cases were made to show the interactions between an actor and the system to achieve a goal. In doing so, an overview of what the system does and does not do was determined. Some of the issues that arise from each step an actor takes to achieve a goal were also understood here. A use diagram and use case titles can be seen below with the full set of use cases appearing in the appendix.

|  |  |
| --- | --- |
| **Use case title** | **Alternative flow** |
| DownloadImage | DownloadImage:Error |
| CloudImage | CloudImage:Error |
| DownloadDrawing | DownloadDrawing:Error |
| CloudDrawing | CloudDrawing:Error |
| LocalImage | N/A |
| DeleteImage | N/A |
| SyncImage | SyncImage:Error |
| CreateDrawing | CreateDrawing:Redo |
| LocalDrawing | N/A |
| DeleteDrawing | N/A |
| SyncDrawing | SyncDrawing:Error |
| CloudDrawing:Error | CloudDrawing:Error |

## Requirements analysis

# Design & Implementation

## Application architecture

## User interface

## Drawings & scores

## Local data

## Cloud data

## Local storage

## Cloud storage

## Multiplatform functionality

# Results Evaluation

## Testing

# Conclusions

## Summary of achievement

## Critical evaluation

## Future work

## Final thoughts

# Bibliography

## List of references

[1] Atack, Nikki E., et al. "A new index for assessing surgical outcome in unilateral cleft lip and palate subjects aged five: reproducibility and validity." *The Cleft palate-craniofacial journal* 34.3 (1997): 242-246.

[2] McKearney, Richard M., John V. Williams, and Nigel S. Mercer. "Quantitative computer-based assessment of lip symmetry following cleft lip repair." *The Cleft Palate-Craniofacial Journal* 50.2 (2013): 138-143.

[3] Russell, James HB, Harriet C. Kiddy, and Nigel S. Mercer. "The use of SymNose for quantitative assessment of lip symmetry following repair of complete bilateral cleft lip and palate." *Journal of Cranio-Maxillofacial Surgery* 42.5 (2014): 454-459.

[4] Stack Overflow. (2017). Stackoverflow.com. Retrieved from http://stackoverflow.com/

[5] PhoneGap Documentation | PhoneGap Docs. (2017). Docs.phonegap.com. Retrieved from http://docs.phonegap.com/

[6] Documentation - Apache Cordova. (2017). Cordova.apache.org. Retrieved from https://cordova.apache.org/docs/en/latest/

[7] cordova. (2017). npm. Retrieved from https://www.npmjs.com/package/cordova

[8] W3Schools Online Web Tutorials. (2017). W3schools.com. Retrieved from https://www.w3schools.com/

[9] Phonegap Tutorial for Beginners to Advanced | PhonegapPro. (2017). PhoneGapPro.com. Retrieved from https://phonegappro.com/phonegap-tutorial/

[10] Build software better, together. (2017). GitHub. Retrieved from https://github.com/

[11] PhoneGap, A. (2017). Get Started. Phonegap.com. Retrieved from http://phonegap.com/getstarted/

[12] cordova-sqlite-storage. (2017). npm. Retrieved from https://www.npmjs.com/package/cordova-sqlite-storage

[13] cordova-plugin-file-transfer. (2017). npm. Retrieved from https://www.npmjs.com/package/cordova-plugin-file-transfer

[14] Notes, S. (2017). SDK Tools Release Notes | Android Studio. Developer.android.com. Retrieved from https://developer.android.com/studio/releases/sdk-tools.html

[15] Windows 10 SDK - Windows app development. (2017). Developer.microsoft.com. Retrieved https://developer.microsoft.com/en-us/windows/downloads/windows-10-sdk

[16] Gandy, D. (2017). Font Awesome, the iconic font and CSS toolkit. Fontawesome.io. Retrieved from http://fontawesome.io/

[17] Paper.js — About. (2017). Paperjs.org. Retrieved from http://paperjs.org/about/

[18] jakiestfu/Snap.js. (2017). GitHub. Retrieved from https://github.com/jakiestfu/Snap.js/

[19] jquery.org, j. (2017). jQuery. Jquery.com. Retrieved from https://jquery.com/

[20]

## Bibliography

[1]

[2]

# Appendices

## System manual

## User manual

## Supporting documentation

## Use Cases

The full list of use cases for the app can be seen below.

|  |  |
| --- | --- |
| USE CASE | DownloadImage |
| ID | UC1 |
| BRIEF DESCRIPTION | A member of hospital staff wants to download post-surgery images to determine success of surgery. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must be connected to the internet. |
| MAIN FLOW | 1) The user selects the option to view patients on the cloud.  2) The user selects an image to download locally.  3) The image is saved locally on the device. |
| POST CONDITIONS | Image stored on device. |
| ALTERNATIVE FLOWS | DownloadImage:Error |

|  |  |
| --- | --- |
| USE CASE | DownloadImage:Error |
| ID | UC1.1 |
| BRIEF DESCRIPTION | A member of hospital staff unsuccessfully attempts to download an image. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must be connected to the internet. |
| MAIN FLOW | 1) The user selects the option to view patients on the cloud.  2) The user selects an image to download locally.  3) The system displays a notification to check the internet connection. |
| POST CONDITIONS | None |
| ALTERNATIVE FLOWS | None |

|  |  |
| --- | --- |
| USE CASE | CloudImage |
| ID | UC2 |
| BRIEF DESCRIPTION | A member of hospital staff wants to view a post-surgery image of a patient. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must be connected to the internet. |
| MAIN FLOW | 1) The user selects the option to view patients on the cloud.  2) The user selects an image to view.  3) Selected image is shown to the user. |
| POST CONDITIONS | Image is displayed on screen. |
| ALTERNATIVE FLOWS | CloudImage:Error |

|  |  |
| --- | --- |
| USE CASE | CloudImage:Error |
| ID | UC2.1 |
| BRIEF DESCRIPTION | A member of hospital staff unsuccessfully attempts to view a patient’s image. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must be connected to the internet. |
| MAIN FLOW | 1) The user selects the option to view patients on the cloud.  2) The user selects an image to view.  3) The system displays a notification to check the internet connection. |
| POST CONDITIONS | None |
| ALTERNATIVE FLOWS | None |

|  |  |
| --- | --- |
| USE CASE | DownloadDrawing |
| ID | UC3 |
| BRIEF DESCRIPTION | A member of hospital staff wants to download a patient’s lip drawing and corresponding symmetry score to store locally. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must be connected to the internet. |
| MAIN FLOW | 1) The user selects the option to view patients on the cloud.  2) The user selects a drawing to download locally.  3) The drawing is saved locally on the device. |
| POST CONDITIONS | Drawing stored on device. |
| ALTERNATIVE FLOWS | DownloadDrawing:Error |

|  |  |
| --- | --- |
| USE CASE | DownloadDrawing:Error |
| ID | UC3.1 |
| BRIEF DESCRIPTION | A member of hospital staff unsuccessfully attempts to download a drawing. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must be connected to the internet. |
| MAIN FLOW | 1) The user selects the option to view patients on the cloud.  2) The user selects a drawing to download locally.  3) The system displays a notification to check the internet connection. |
| POST CONDITIONS | None |
| ALTERNATIVE FLOWS | None |

|  |  |
| --- | --- |
| USE CASE | CloudDrawing |
| ID | UC4 |
| BRIEF DESCRIPTION | A member of hospital staff wants to view a drawing of a patient’s lip region. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must be connected to the internet. |
| MAIN FLOW | 1) The user selects the option to view patients on the cloud.  2) The user selects a drawing to view.  3) Selected drawing is shown to the user. |
| POST CONDITIONS | Drawing is displayed on screen. |
| ALTERNATIVE FLOWS | CloudDrawing:Error |

|  |  |
| --- | --- |
| USE CASE | CloudDrawing:Error |
| ID | UC4.1 |
| BRIEF DESCRIPTION | A member of hospital staff unsuccessfully attempts to view a patient’s lip drawing. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must be connected to the internet. |
| MAIN FLOW | 1) The user selects the option to view patients on the cloud.  2) The user selects a drawing to view.  3) The system displays a notification to check the internet connection. |
| POST CONDITIONS | None |
| ALTERNATIVE FLOWS | None |

|  |  |
| --- | --- |
| USE CASE | LocalImage |
| ID | UC5 |
| BRIEF DESCRIPTION | A member of hospital staff wants to view a post-surgery image of a patient. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must have already downloaded the image. |
| MAIN FLOW | 1) The user selects the option to view patients stored locally.  2) The user selects an image to view.  3) Selected image is shown to the user. |
| POST CONDITIONS | Image is displayed on screen. |
| ALTERNATIVE FLOWS | None |

|  |  |
| --- | --- |
| USE CASE | DeleteImage |
| ID | UC6 |
| BRIEF DESCRIPTION | A member of hospital staff wants to delete a patient stored locally. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must have already downloaded the image. |
| MAIN FLOW | 1) The user selects the option to view patients stored locally.  2) The user selects an image to delete.  3) Selected image is deleted. |
| POST CONDITIONS | Updated list of locally stored patients are displayed. |
| ALTERNATIVE FLOWS | None |

|  |  |
| --- | --- |
| USE CASE | SyncImage |
| ID | UC7 |
| BRIEF DESCRIPTION | A member of hospital staff wants to upload an image’s drawings and symmetry scores to the cloud. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must be connected to the internet. |
| MAIN FLOW | 1) The user selects the option to view patients stored locally.  2) The user selects an image to sync.  3) The system displays a notification that sync has completed. |
| POST CONDITIONS | Drawings and symmetry scores are uploaded to the cloud. |
| ALTERNATIVE FLOWS | SyncImage:Error |

|  |  |
| --- | --- |
| USE CASE | SyncImage:Error |
| ID | UC7.1 |
| BRIEF DESCRIPTION | A member of hospital staff unsuccessfully uploads an image’s drawings and symmetry scores to the cloud. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must be connected to the internet. |
| MAIN FLOW | 1) The user selects the option to view patients stored locally.  2) The user selects an image to sync.  3) The system displays a notification to check the internet connection. |
| POST CONDITIONS | None |
| ALTERNATIVE FLOWS | None |

|  |  |
| --- | --- |
| USE CASE | CreateDrawing |
| ID | UC8 |
| BRIEF DESCRIPTION | A member of hospital staff wants to determine the success of a surgery from an image. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must have saved an image locally. |
| MAIN FLOW | 1) The user selects a locally stored image.  2) The user selects the option to create a drawing of lip regions.  3) The user draws around the lip regions.  4) The user selects the option to generate symmetry scores.  5) The system displays the symmetry score. |
| POST CONDITIONS | Symmetry score is generated and displayed. |
| ALTERNATIVE FLOWS | CreateDrawing:Redo |

|  |  |
| --- | --- |
| USE CASE | CreateDrawing:Redo |
| ID | UC8.1 |
| BRIEF DESCRIPTION | A member of hospital staff drew around the lip regions incorrectly. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user wants to redraw around the lip regions. |
| MAIN FLOW | 1) The user draws around the lip regions incorrectly.  2) The user selects the option to redraw around the lip regions.  3) The use case is repeated until the user is satisfied with the drawing and selects the option to generate symmetry scores.  4) The system displays the symmetry score. |
| POST CONDITIONS | Symmetry score is generated and displayed. |
| ALTERNATIVE FLOWS | CreateDrawing:Redo |

|  |  |
| --- | --- |
| USE CASE | LocalDrawing |
| ID | UC9 |
| BRIEF DESCRIPTION | A member of hospital staff wants to view a drawing of a patient’s lip region. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The image must have drawings associated with it. |
| MAIN FLOW | 1) The user selects the option to view patients stored locally.  2) The user selects the option to view drawings of an image.  3) The user selects a drawing to view.  4) Selected drawing is shown to the user. |
| POST CONDITIONS | Drawing is displayed on screen. |
| ALTERNATIVE FLOWS | None |

|  |  |
| --- | --- |
| USE CASE | DeleteDrawing |
| ID | UC10 |
| BRIEF DESCRIPTION | A member of hospital staff wants to delete a drawing stored locally. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must have drawings stored locally. |
| MAIN FLOW | 1) The user selects the option to view patients stored locally.  2) The user selects the option to view drawings of an image.  3) The user selects a drawing to delete.  4) Selected drawing is deleted. |
| POST CONDITIONS | Updated list of locally stored drawings are displayed. |
| ALTERNATIVE FLOWS | None |

|  |  |
| --- | --- |
| USE CASE | SyncDrawing |
| ID | UC11 |
| BRIEF DESCRIPTION | A member of hospital staff wants to upload a single drawing of an image’s lip regions and symmetry scores to the cloud. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must be connected to the internet. |
| MAIN FLOW | 1) The user selects the option to view patients stored locally.  2) The user selects the option to view drawings of an image.  3) The user selects a drawing to sync.  4) The system displays a notification that sync has completed. |
| POST CONDITIONS | Drawings and symmetry score is uploaded to the cloud. |
| ALTERNATIVE FLOWS | SyncDrawing:Error |

|  |  |
| --- | --- |
| USE CASE | SyncDrawing:Error |
| ID | UC11.1 |
| BRIEF DESCRIPTION | A member of hospital staff unsuccessfully uploads an image’s drawing and symmetry score to the cloud. |
| PRIMARY ACTORS | Clinician |
| SECONDARY ACTORS | Researcher |
| PRECONDITIONS | The user must be connected to the internet. |
| MAIN FLOW | 1) The user selects the option to view patients stored locally.  2) The user selects the option to view drawings of an image.  3) The user selects a drawing to sync.  4) The system displays a notification to check the internet connection. |
| POST CONDITIONS | None |
| ALTERNATIVE FLOWS | None |

## Test results

## Evaluation data

## Project plan

## Interim report

## Code listing