Engineering Notebook

Monday, September 21, 2020 5:15 PM

Engineering Notebook:

The Engineering notebook is an ongoing electronic journal of your activities on the project. Content that needs to be covered in this document includes (but not limited):

- The specific component of the project you worked on (code, hardware component, etc.)
- The specific problem you solved
- The specific document you worked on
- The components you tested

Each week, we will randomly select one or two people per team and review their notebook for our evaluation. Therefore, there is a possibility that throughout the semester, your notebook will be evaluated 2-4 times. In addition, you will be required to submit your final version of the notebook on November 25th. *If the notebook is not available at the moment of random selection, it will be consider as a no submission*. These will be reviewed at the end of any class meeting times.

Initial Research

Friday, September 11, 2020 10:04 PM

KNN

- KNN stands for k-nearest neighbors
- Is a supervised machine learning algorithm that relies on labeled input data to learn a function that produces an appropriate output when given new unlabeled data.
- The KNN algorithm assumes that similar things exist in close proximity.
- Calculations
  - Distance between two points
    - C = $\sqrt{a^2 + b^2}$
    - C=$\sqrt{(x_A - x_B)^2+(y_A - y_B)^2}$
- The KNN Algorithm
  - Load the data
  - Initialize K to your chosen number of neighbors
  - For each example in the data
    - Calculate distance between the query example and the current example from the data.
    - Add the distance and the index of the example to an ordered collection
  - Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
  - Pick the first K entries from the sorted collection
  - Get the labels of the selected K entries
  - If regression, return the mean of the K labels
  - If classification, return the mode of the K labels
- Note:
  - As we decrease the value of K to 1, our predictions become less stable.
  - As we increase the value of K, our predictions become more stable due to majority voting / averaging, and thus, more likely to make more accurate predictions (up to a certain point).

From <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

SVM

- SVM stands for Support Vector Machines
- Is a supervised learning models
- SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.
  - The objective of applying SVMs is to find the best line in two dimensions or the best hyperplane in more than two dimensions in order to help us separate our space into classes.
- Looks at the extremes and draws a decision boundary (aka hyperplane)
- D- and D+ the section in the middle (0) is the margin or hyperplane

Vision Statement/Meeting 9/12

Saturday, September 12, 2020 11:58 AM

Vision Statement

To create a system that provides a patient with representative faces of different nose types that align with the measurements of their face, therefore allowing the patient to select which nose is attractive for their rhinoplasty procedure.

I created - vision statement

Notes: Faces similar to their own / representative different type of noses and select which one is attractive will fit on a face that looks like your face

http://openimaj.org/tutorial/intro-facial-feature-extraction.html

Stand up 9/15

Tuesday, September 15, 2020 2:31 PM

Trying to

Doing now:

Me & Chris - working up uploading photo using python script, get that photo and display it to you using script

We are not trying to pick most attractive nose

We want face that matches the patients face. We want to show these options to the customer.

Database w/ actors/models - **not our main goal to define beauty'(Akbas)**

We just want to find bodies that match (jacob)

Akbas to do - send us some more librarys
3d scans from web application to use as our patient

2d face scan (input) the output needs to be matching faces

Backlog
We made changes for the app that we want to do application through

We are using python app

What the product will be at the end of the semester?(Akbas -

Display 5 faces to show patient.
These noses look good (select 2?_

Say they select 1 we should be able to tell what changes need to be done to meet that selected nose.

*****We need to tell them what will be different (size - space from bridge,)

Summary of how its different - giving hints to the surgeon to tell them how to change the users nose to the new one.

(the difference between the n

Python Scripting

Friday, September 18, 2020 7:32 PM

Task 1: create python script
script 1: name file yourself and save to my repository file location
Script 2: don't ask user for name it saves a s default and pushes it to the
Note: No longer needed to do script 2 (9/18/20 and 9/19/20)

Useful links:

https://www.geeksforgeeks.org/python-pil-image-open-method/

Python Scripts

Created on Wed Sep 16 09:39:52 2020

@author: Vjord

script 1: name file yourself and save to my repository file location
Script 2: don't ask user for name it saves a s default and pushes it to the

note: repositoryjo-victoria/imagedatabase

"""

"script 1"

```
#!/usr/bin/python
import cgi, os
import cgitb; cgitb.enable()
form = cgi.FieldStorage()
# Get filename here.
fileitem = form['filename']
# Test if the file was uploaded
if fileitem.filename:
   # strip leading path from file name to avoid
   # directory traversal attacks
   fn = os.path.basename(fileitem.filename)
   open('/tmp/' + fn, 'wb').write(fileitem.file.read())
   message = 'The file "' + fn + '" was uploaded successfully'
else:
   message = 'No file was uploaded'
print(message)
```

```
# -*- coding: utf-8 -*-
"""
Created on Fri Sep 18 19:58:39 2020

@author: Vjord
"""

# "Base code to open an image at a fixed location"
# from PIL import Image

#read the image
# im1 = Image.open(r"C:\Users\Vjord\Pictures\avionlogo.PNG")

#show images
# im1.show()


"code to allow user to select a photo"
import cv2
import os

def load_images_from_folder(folder):
    images = []
    for filename in os.listdir(folder):
        img = cv2.imread(os.path.join(folder,filename))
        if img is not None:
            images.append(img)
    return images
```

JSON Files Task

Monday, September 21, 2020 5:11 PM

Task: Open JSON files and find the functionality
GitHub Info: Research javascript files sent by akbas #11 (From <https://app.zenhub.com/workspaces/se-450-5f53121d05c784001ae8505a/issues/king-of-kong/machine-learning-nose-jobs/11>)

How to open JSON research
https://medium.com/@antonmedv/how-to-work-with-json-from--f5a10cbe2121
https://stackoverflow.com/questions/8553200/how-do-i-install-and-use-a-json-editor-in-eclipse
https://sourceforge.net/projects/eclipsejsonedit/
https://marketplace.eclipse.org/marketplace-client-intro?mpc_install=945

Features.json (sample script)

```
var fpoints = [
 {
  "id": "FirstPoint",
  "name": "1st Point",
  "imageFile": "feature_point_first_point-frontal.png",
  "description": "Mark any point needed to calculate a distance, an angle or a ratio.",
       "usedInMeasurements": ["Distance_First_Second", "Angle_First_Second_Third",
       "Ratio_First_Second_Third_Fourth"],
       "xVal": "",
       "yVal": "",
       "zVal": "",
  "type": "Points"
 },"usedInMeasurements": ["Angle_First_Second_Third", "Ratio_First_Second_Third_Fourth"],
       "xVal": "",
       "yVal": "",
       "zVal": "",
  "type": "Points"
 },
```

Description:

- Features.json
  - Labels:
    - Id - same as name
    - Name - same as id
    - imageFile - image name/ can classify what position the photo is in.
    - Description - Mark any point needed to calculate a distance, an angle or a ratio
    - usedInMeasurements - values for distance, angle, and ratio
    - xVal
    - yVal
    - zVal
    - Type

- Female_Head_3-2006
    - Contains data/info for the features.json file
- Measuremaents.json
    - Filled in information
    - Formula to calculate: Pythagorean's theorem. C^2=a^2 +b^2

Standup 9/22

Tuesday, September 22, 2020 2:33 PM

W/ Akbas

Vision statement critique:
Add to the vision statement something that relates to:

- What changes need to be done on this version
- What nose they can select
- Since you have measurements you can say you can

Calculate difference on the nose bridge
Goal for this semester - what we find about the nose is calculate measurements and print them

Need to match proportions of the face (like generating a whole new image)

Suggestion (Akbas) : give option on diff faces

The faces look alike, but there will be proportion differences.

Just want draft of changes that need to be done. [Akbas]

Product Backlog critique

- He only sees 4 tasks as backlog items
- Backlog needs to show everything we are planning to do
- Then look at a few issues
- Github tutorial - need to put what is on which was sprint
- Need to fix backlog and show how to display it.

Anton

- Face clustering

Chintan

- Cv2
- pip3 install opencv-python

Standup 9/24

Thursday, September 24, 2020 2:33 PM

67-8 = want to match the features and not the nose (== total of 59 features to detect items)

Match points that they have and put them in the json files (****DO THIS***)
JSON files contain the point

What measurements we take will match those in the measurements.json file
8 (features on nose)
59 (on general face)

Standup 9/29

Tuesday, September 29, 2020 2:32 PM

Chintan
- Able to spit out code for each face in CSV file for 1600 pictures

Victoria
- JSON
  - Helpful links:
    - https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON#:~:text=JavaScript%20Object%20Notation%20(JSON)%20is,page%2C%20or%20vice%20versa).
    - https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Basics

Jacob
- Nothing now, will integrate stuff to python

Anton
- Created algorithm

Chris
- Working/researching algorithm
- Learning how to do python app

What will we show during demo (Akbas)?
- Show dlib/algorithm to show how measurements will be captured
- Consider as --- skeleton
- Okay with seeing what we want to show
- For the sprint we were working on this and we want to show this
- After this sprint he will see facial points

We changed structure of design class
- This semester he wants us to focus on implementation
  - Focus on the product

Best tool is ratios

After Team Meeting
- I'll work on SDS
- Jacob and Chris will look into the python app
  - Are assigning themselves it in github
- Anton will host demo on his computer for Thursday's demo

SDD

Wednesday, September 30, 2020 11:40 AM

Updated the following sections of the SDD

1. INTRODUCTION

2. Purpose and Scope

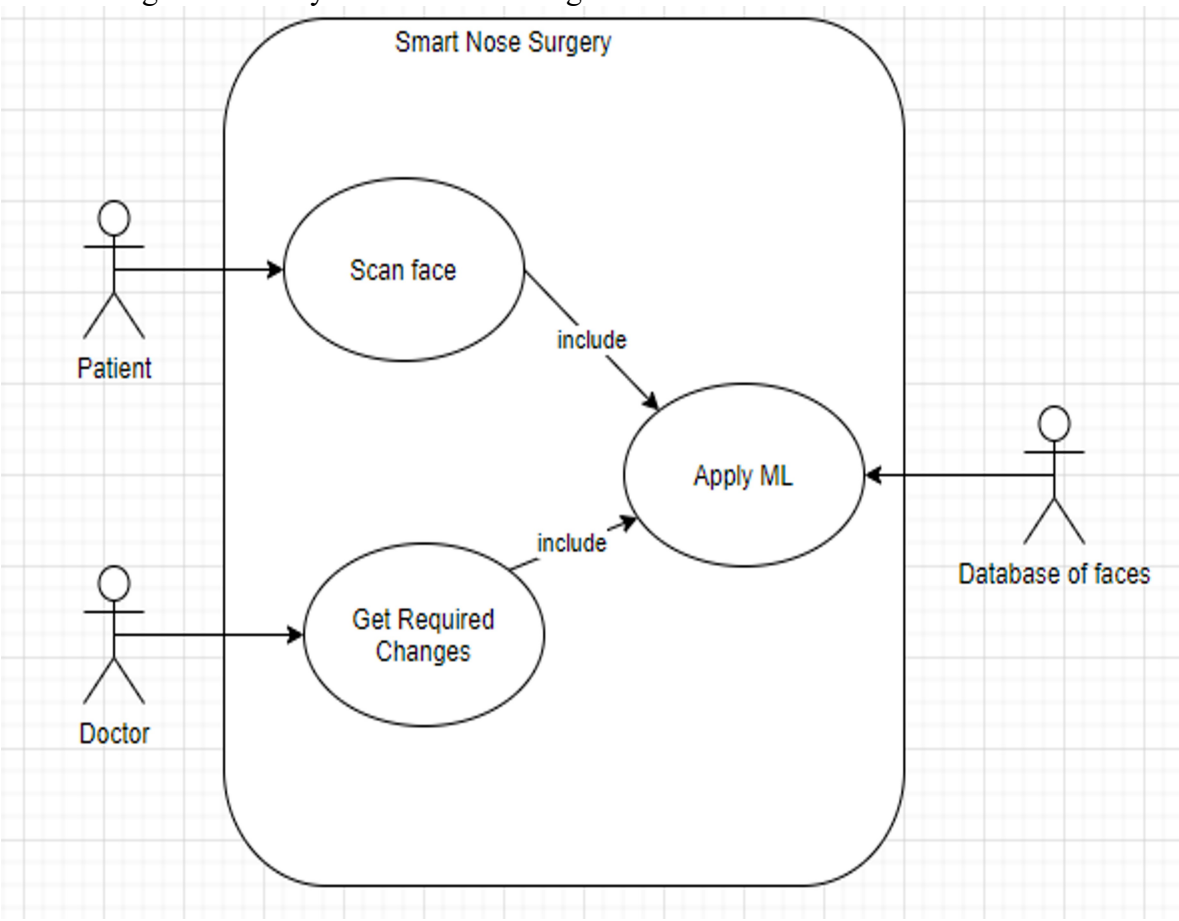
This document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, and external interfaces for Machine Learning Algorithm for Rhinoplasty (M-LAR) system.

3. Project Executive Summary

This section provides a description of the Machine Learning Algorithm for Rhinoplasty (M-LAR) system from a management perspective and an overview of the framework within which the conceptual system design was prepared.

4. System Overview

The system will provide a patient with representative similar faces with different nose types that align with the measurements of their face, therefore allowing the patient to select which nose is preferred for their rhinoplasty procedure. Below is the high-level architecture and context diagram of the system's overall design.



5. Document Organization

This document is designed to describe the structure of the Machine Learning Algorithm for Rhinoplasty (M-LAR) system.The following sections will provide information on what the product does, limitations, interactions, interfaces, hardware and software designs, and security.

6. Project References

1. System Requirements Specification

7. Glossary

1. M-LAR - Machine Learning Algorithm for Rhinoplasty (M-LAR) system
2. "Patient" and "User" are used as interchangeable terms.
3. KNN - "k-nearest neighbors", a machine learning algorithm that, when given a test point represented by a point on the coordinate plane, will find the k points in the data set, also in the coordinate plane, that are nearest to the test point.

4. SYSTEM ARCHITECTURE

This section describes an overview of the hardware and software architecture for the M-LAR system and subsystems.

8. System Hardware Architecture

In this section, describe the overall system hardware and organization. Include a list of hardware components (with a brief description of each item) and diagrams showing the connectivity between the components. If appropriate, use subsections to address each subsystem.

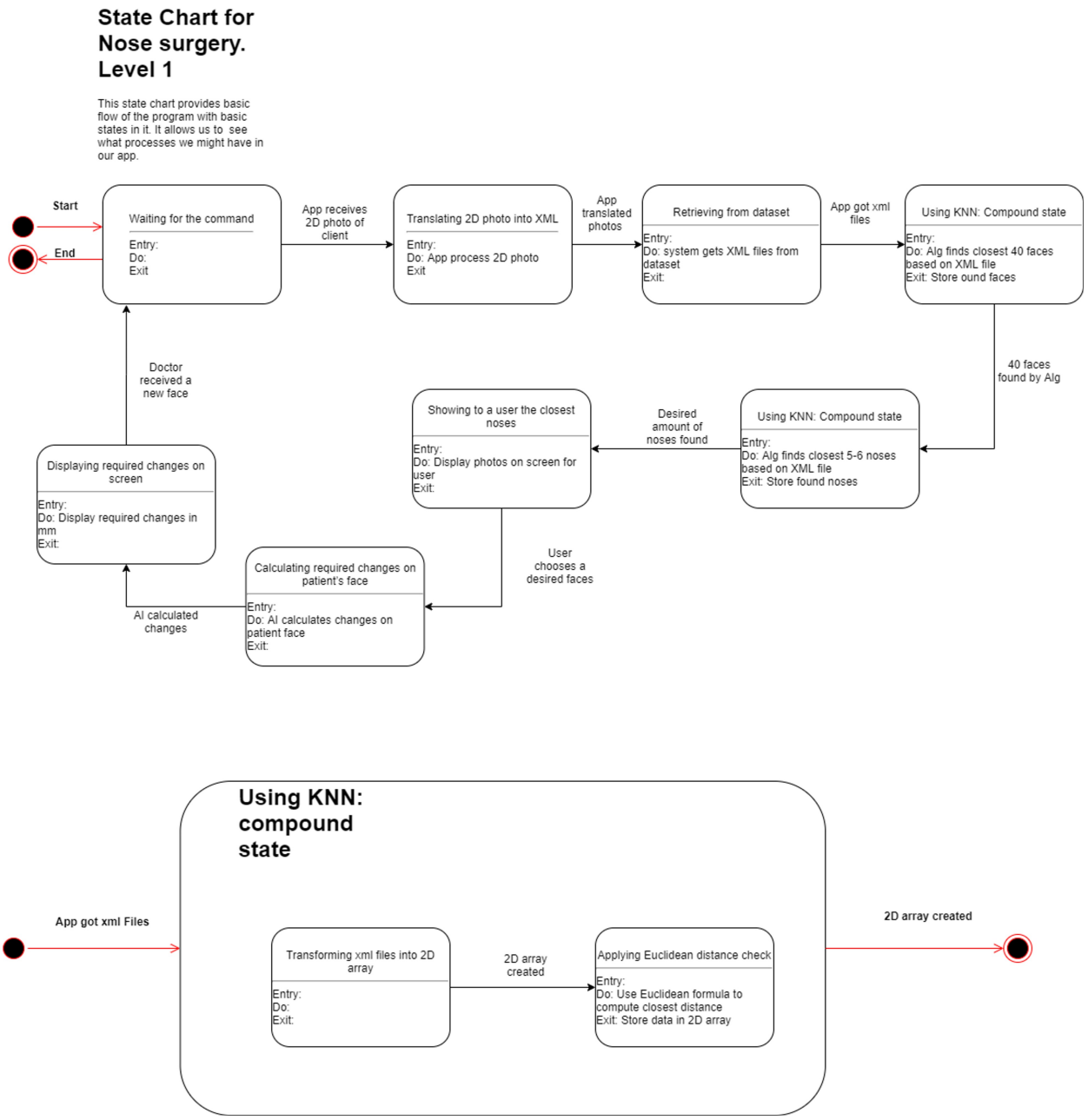
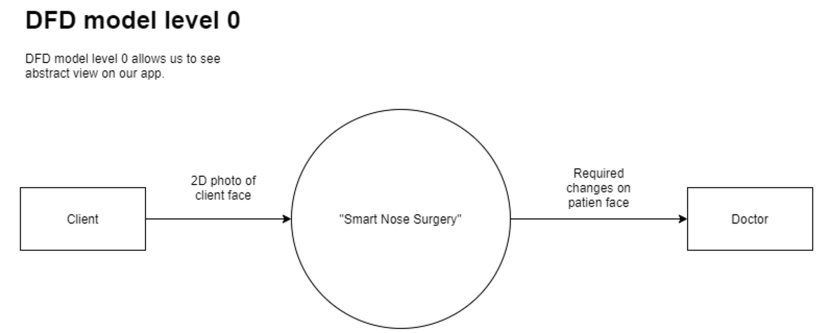


Figure 1: State Chart Diagram

Figure 1 describes what states and conditions will be cycled through in the system.



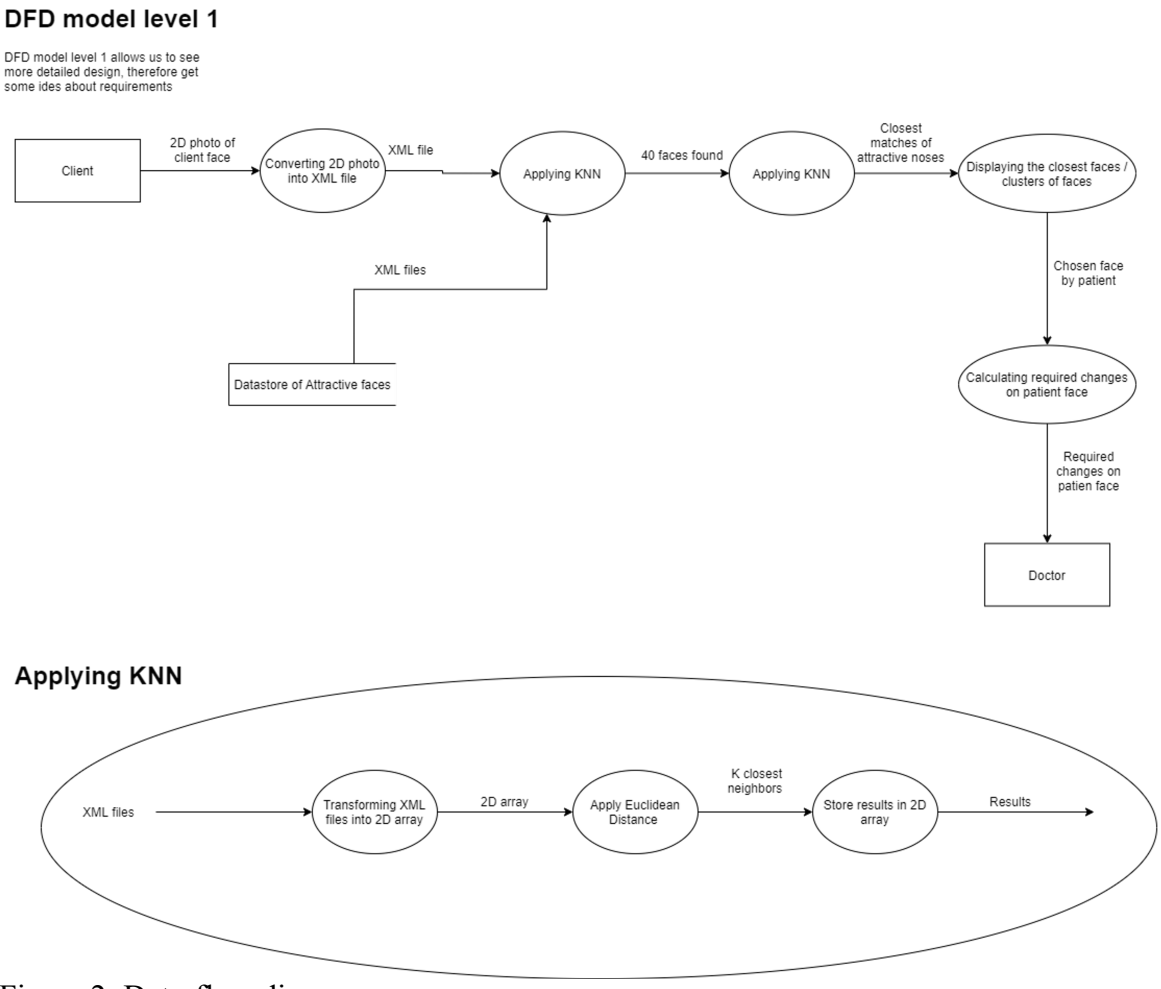


Figure 2: Data-flow diagram

Figure 2 describes the transfer of data in the system between the client and the doctor.

Demo Day - class

Thursday, October 1, 2020 2:38 PM

Notes:

Doesn't like listing 40 faces

Achieve dimensionally -
Update to 3D -
40 faces but can update
Looking at Euclidean distance is nice, now lets look at clusters.

Similar faces in groups, when you are proposing changes it doesn't have to go from location x to

After meeting
Chris - likes

Stand up

Tuesday, October 6, 2020 2:22 PM

What will the product be at the end of sprint 2

- Goals: 1)Backend algorigthm, you give an algorithm and it does knn. Assign Z score based on mean. 2)Python app - display on grid
- In a demo what will we see in the sprint
    - [Write here in the notebook]
    - Do scenario of what will happen during this time
    - Here is what we are planning
    - Rough outline of what we're doing
    - Share link w/ Akbas
    - Z-Score of the face
        - Statistical method/numerical value method

Demo Sprint 2
Steps:

- Log into designed python app (save for sprint 3)
- Ask user for a photo
    - Options:
        - Program access users' computer
        - User uploads a photo
        - ~~Define what position we would like the photo to be~~
            - Front facing
                - Profile(distant future maybe)
- Program processes the user photo
    - During process the system
        - Run dlib face detector
        - Run statistical analysis (Z-Score?) - give users' face a z-score
        - Uses KNN to define the points for nearest neighbor
            - Faces passed down for clustering
- Display a grid of faces that have been recommended for the user

  Demo KNN and clustering working on its on.

Show how many faces KNN recommends
Run script for uploading photo
Face detector using dlib
Output through KNN
Show grid of all of the faces
"cluster algorithm"
Give photo --> see representative faces

Later task - redo csv files

After standup meeting

A - look for svm, meet with akbas,

Create a program that does subtraction
Logging in and security

Standup 10/12 w/Tayler

Monday, October 12, 2020 12:07 PM

SRS:
Separate table of contents
But who did each section and put in change log
Definitions and acronyms in alphabetical order
Graphs were find
We missed a section from template
Add back shalls
Rework on formatting (numbering vs - bullets)

SDD:
Put caption under our diagrams
Bullet points in glossary

I did sprint demo outline. Todo: begin 3D faces

Not edit image but return calculations

Standup 10/8

Thursday, October 8, 2020 2:24 PM

Stick to KNN and SVM for sprint 2

Didn't like that the users

SRS

Wednesday, October 14, 2020 11:14 PM

Updated SRS for Version 2

Updates includes

- Formatting
    - Bulleting
    - Numbering
- Wording "the system shall"
-

Standup 10/20

Tuesday, October 20, 2020 2:18 PM

Class - goals for next sprint, video describe project

Video is replacement of live demo

Question: do we need to readapt out demo outline

Stand-up

Demo is done, fixing/fixed SRS, 3D library

Chris - satisficed with how the library look. Got the app working to take the picture.

Will it be ready for the demo?

- Yes - can use KNN

Anton - faces rearrange

Show how to demonstrate its working - test plans

Victoria -

Grade scale from Jafer
Consider video- spell out what we have done since sprint 1.

Instead of KNN algorithm - do virtual piece

Chintan - faces

Jacob - going into todays meeting what to work on next

DEMO IS NEXT THURSDAY
If we want to do a dry run let akbas now

But engineering notebook on wiki page??

After Meeting 2:45 - :

Main things we need to do
Clustering

Easy way to do K #

Redo the csv (height/width/nose point)

Get knn algorithm on app

Clustering - get clustering to work

Facial feature.py
Get csv reader from testerapp. (put two together

```
def calculateFacialSize(self):
    width = self.facialCords[16][0] - self.facialCords[0][0] # top right(17) - top left(1)
    height = self.facialCords[8][1] - self.facialCords[27][1] # bottom center(9) - top center(28)
    return (width , height)
```

Standup 10/22/20

Thursday, October 22, 2020 2:16 PM

Regular Class Meeting

- Sprint 2 demo is next Thursday
- Peer evaluations due Oct.28
- Put engineering notebook on Github
- SDD/SRS full document due

Standup

- Chintan
  - Thing not working
  - Need to
- Chris
  - Looking for images/
- Jacob
  - Trimming down csv
- Anton
  - Test cases
- Victoria
  - CSV
- Notes for Akbas:
  - Demo next week
  - Any problems in the team try to solve process for fixing it
  - Try to equalize/fair workload
  - Wants a trailer of the demo on Tuesday

Afternoon:

- Talked about ratios last meeting
  - Issues about using width and height originally
- How are we doing the csv files
  - By names - ratios
  - Run only one to calculate data - KNN
  - "one line code"
- Chris meet with Tayler to push back epic
- What's left
  - Demo trailer - chris needs to finish
  - CSV - chintan fix
  - Jacob - read the file
  - Trailers - make presentation (Victoria - contact Anton)

- Soon as csv is done - need to fix look at stuff
- Diagrams
- Ratio compute (jacob)
- Main app (chris)