System Design Document

For

Digitized Rhinoplasty

Team member: Priscilla Carbo, Marvin Cazeau, Jared Curtis, Maree Kelly, Alessandra Oo

Version	Date
v1.0	16 September 2021
v1.1	28 September 2021
v1.2	26 October 2021

TABLE OF CONTENT

SYSTEM DESIGN DOCUMENT

Overview

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

1 INTRODUCTION

1.1 Purpose and Scope

The purpose of the Systems Design Document is to track the necessary information required to effectively define the architecture and system design for Digitized Rhinoplasty v1.0. This is done to accurately detail and provide guidance for the development team of the system architecture implementation. Information such as the operating environment, system architecture, interface design, and machine learning techniques included will be found within this document.

1.2 Project Executive Summary

This section provides a description of the Digitized Rhinoplasty project from a management perspective and an overview of the framework for which the conceptual system design was prepared.

1.2.1 System Overview

The goal for the Digitized Rhinoplasty project is to provide a patient with noses of others similar to their own, which is chosen from a database by the k nearest neighbors (KNN) algorithm or the convolutional neural network (CNN) algorithm. The images that the algorithm outputs are the people that have similar facial structure as the patient and the patient will choose a nose that they like the most. Using Blender, the patient will be able to see a 3D image of their face with their current nose, and then after their measurements go through the algorithm, they will be able to see a 3D image of the new nose on their face.

The following diagram (Figure 1) gives an overview of the methodology of the system:

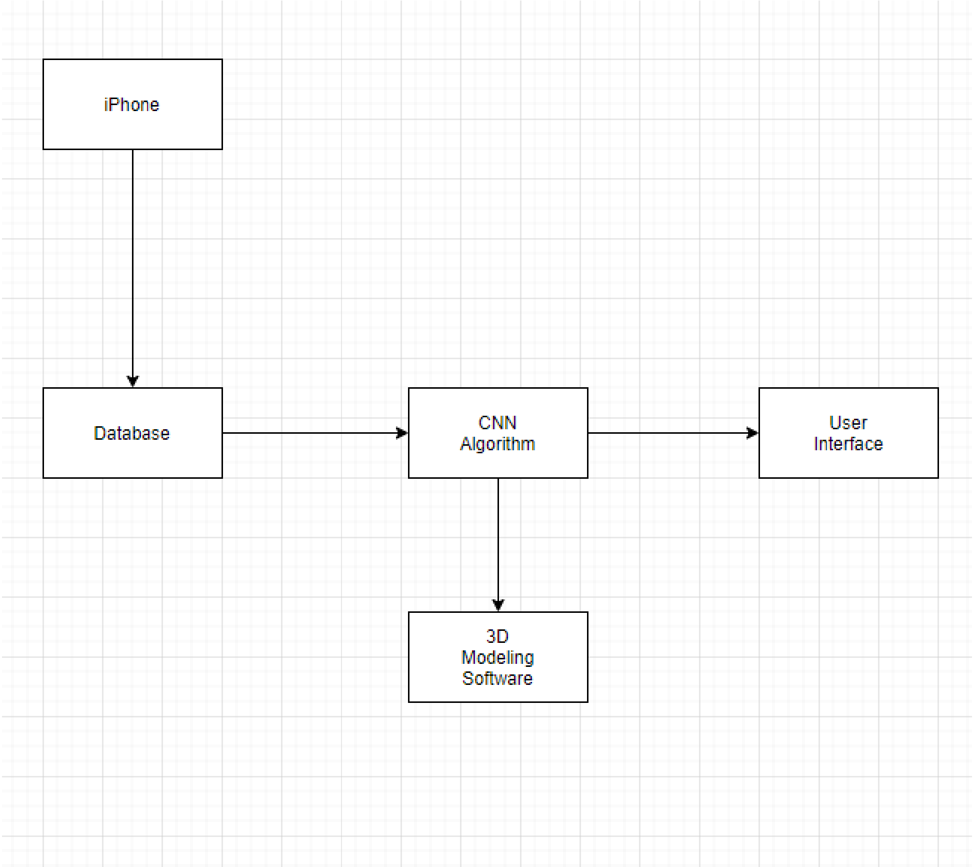

Figure 1: Overview of the methodology of the entire system

1.2.2 Design Constraints

In order to actually be used in a medical setting, the system will need to be approved by the FDA to guarantee that the results it produces are reliable and accurate. Additionally, the system relies on an app called “bellus3D” to produce 3D scans of the patient, which can only be downloaded using an Apple device with the app store.

1.2.3 Future Contingencies

Contingencies that may arrive in the future in Unsuitable machine learning AI suggestions, Machine learning Incompatibility, and one that is occurring at the time this document is being produced is lack or difficulty with implementing and meshing systems. Unsuitable machine learning AI (or ML AI for short) is the possibility that a suggestion is made that is either clearly bugged or just very off of the set perimeters so the image created is incorrect in various ways. This would require rigorous testing to identify areas of weakness in the code that could cause this issue. Otherwise, we could implement a quick rerun option for the client/surgeon if this issue does occur after such testing. ML incompatibility is the possibility that the ML we’re trying to implement just won’t work for our architecture in its current state. Solutions range from selecting an AI that works with our architecture or redesigning our architecture to cater to the new ML AI. An issue occurring right now and most likely in the future is the difficulty with implementing and meshing systems. Our architecture currently has issues meshing blender and a python script from a previous design. A workaround for this issue and future contingencies would be to build new code from the ground up that caters to our current architecture.

1.3 Document Organization

This document is designed to give the reader an overview and in-depth look on the system design and the documentation to understand the systems, process, and architecture. The following will include the system architecture, the different interfaces used in this project, hardware and software designs, and the system integrity.

1.4 Glossary

Rhinoplasty: *noun*	plastic surgery performed on the nose
FDA	Food and Drug Association
KNN	K Nearest Neighbors

2 SYSTEM ARCHITECTURE

2.1 System Software Architecture

2.1.1 User Interface

- The user interface will consist of a login page and main page to navigate various enhancements to the app. This will be written in HTML, CSS and Javascript.
- This interface will use React API to implement microservices that will be used to display files from blender.
- The landing page will have a navigation bar, a container for the 3D model, and a form for the user to input patient information and upload a scanned image to the application.
- Upon uploading a 3D scan for the new patient, the user will be prompted to execute the algorithm and a 3D model from blender will appear in the left container.
- The interface will include a calendar for scheduling rhinoplasty procedures, as well as an archive of previous patients with their before and after results.
- The interface will be accessible through a webpage in order to be implemented into the original rhinoplasty application created by the customer.

2.1.2 Database

- The iPhone is used to download Bellus3D, the scanning application, from the App Store.
- The application is then used to scan the patients' and volunteers' faces to create a database. The scan is then sent as an object file to the Digitized Rhinoplasty web application.
- The Digitized Rhinoplasty Application is used to plot the needed points on the patients'/volunteers' faces. After positioning the specified points on the nose the JSON files are exported from the application.
- The JSON files are read by read_in.py using Pandas, after the code is run the user is prompted to enter the file name on the console. After entering the file name of the desired file, the program creates data frames and generates a .CSV file of the final data frame into the same directory where the file is being run from. The naming convention uses the patient's original file name and "_df" added at the end.

3 HUMAN-MACHINE INTERFACE

This section provides the detailed design of the system and subsystem inputs and outputs relative to the user/operator. Any additional information may be added to this section and may be organized according to whatever structure best presents the operator input and output designs. Depending on the particular nature of the project, it may be appropriate to repeat these sections at both the subsystem and design module levels. Additional information may be added to the subsections if the suggested lists are inadequate to describe the project inputs and outputs.

3.1 Inputs

- Each volunteer, with assistance from a Digitized Rhinoplasty project team member, will input their face scan into the Bellus3D application.
- The input to the Digitized Rhinoplasty application is the scanned image as an object.
- The system operator will input the patient’s name, ID number, and a 3D scan of the patient's face as a JSON file.
- The Blender application takes in old and new points into Blender via .XLSX file as well as the patient's 3D .OBJ file.

3.2 Outputs

- The Machine Learning Algorithm will update the SQL database with the changed points of the patient's nose.
- The system will then pull the .JSON files from SQL and convert them to .XLSX files so that Blender may interact with it.
- The system shall allow Blender to output the new nose .OBJ file when it finishes executing the script in Blender.
- The measurements of change in nose shape are output to an XML file for the surgeon and can be displayed in the UI.

4 DETAILED DESIGN

4.1 Hardware Detailed Design

The only hardware component needed for the Digitized Rhinoplasty project is an iPhone with iOS 11.1 or later. The Bellus3D application will only work with the stated phones to ensure the scan is high definition. Reference Figure 1 in order to see how the scanning application is connected to the Digitized Rhinoplasty project system.

4.2 Software Detailed Design

- The user will scan their image and upload it to the digitized app and add all needed points.
- Then it will allow the user to upload the JSON file and that file will turn into a CSV file, that CSV file will be used for blender.
- The UI will allow the user to login and upload JSON files of patients.
- The UI will already have all of the users data uploaded & the measurements after going through the algorithm.
- Patients' 3D face scans will be analyzed, and data points will be collected.
- The KNN algorithm will input the K number of patients that are closest to the facial structure of the user.
- The user will input the K number for example 3 or 4 depending on the similarities they have with the database and the patient will then choose a nose that fits their liking.
- Blender will show the new nose on the patient's face.

4.3 Internal Communications Detailed Design

We will be using a k-NN Algorithm, first the k-NN algorithm is triggered, then the current patient’s csv file is read and the 14 needed points are pulled. Each point pair is put into the euclidean distance and ratio functions to find the ratio of the distance between the points. The next step in the algorithm is to compare the patient’s face to the database of face ratios. We decided to have k be dependent on each individual patient and how many similar faces can be found within some tbd radius of the current patient. After k similar faces are found, the current patient chooses the nose they believe suits them best and the chosen nose is sent to blender along with the current patient file.

5 EXTERNAL INTERFACES

5.1 Interface Architecture

The external system used by the Digitized Rhinoplasty project system is an SQL server. The system communicates with the SQL server to create its database of faces. Users are able to access the face database through queries. The user will be able to execute the machine learning algorithm, compare before and after 3D models of the patient's face, and display measurements for the surgery.

5.2 Interface Detailed Design

The external system used by the Digitized Rhinoplasty project system is a SQL server. The system communicates with the SQL server to create its database of faces. Users are able to access the face database through queries. The surgeon will decide what kind of query one would like in order to access information. The options include: pulling stored patient data, saving input patient face measurements, and saving the system created incision data.

6 SYSTEM INTEGRITY CONTROLS

The integrity controls for this application are designed to manage the integrity of the data. The Digitized Rhinoplasty project system will have access to the patient's facial point data. The system will save the data after the measurements have been calculated by the surgeon. Only Verified Digitized Rhinoplasty project personnel will be able to interact with the system and have access to all the data and be authorized to make modifications to the data.

To ensure that the system is safe from common threats of modern information systems the following levels of control will be implemented:

- Input controls ensure accuracy of data entered into the application. This is implemented within the Digitized Rhinoplasty project system or the database schema, or both, by assigning input control types, such as value limit controls, completeness controls, data validation controls, and field combination controls. Data is only entered by verified Digitized Rhinoplasty project personnel.
- Internal security or access controls to restrict access of sensitive data to only the surgeon, this does not include the patient.
- Ability to identify all audit information by user identification, network terminal identification, date, time, and data accessed or changed.
- Output controls are concerned with the output of the data calculated after the patient picks a nose. The output controls will appear on the display GUI.