System Design Document

For

Digitized Rhinoplasty (Option 9)

Team members: 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
v1.3	30 November 2021
v2.0	02 February 2022

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 v2.0. 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 generate realistic 3D faces through a GAN ML algorithm while automatically mapping significant facial landmarks on a patient and providing the patient with other noses, which are similar to their own. These clusters of similar looking patients will be chosen from a database by a neural network based 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.

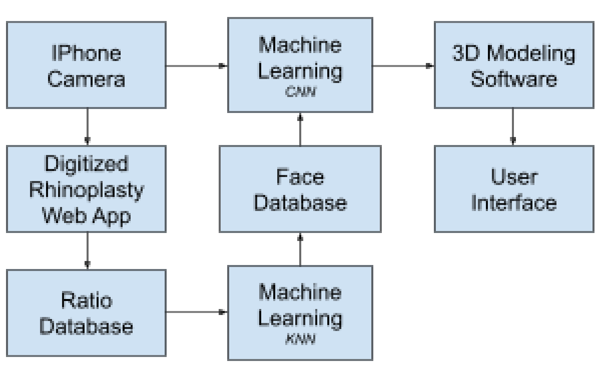


Figure 1: Overview of the methodology of the entire system

1.2.2. Design Constraints

There are certain design constraints considered given that the system is to be used in medical applications. In order to be used in an actual 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 relied on an app called "Bellus3D" to produce 3D scans of the patient, which can only be downloaded using an iOS device with the iOS App store. Additionally, the application has been removed from the Apple app store since the end of December, 2021. Previous implementations had us using the KNN (K-Nearest Neighbors) algorithm which we found did not suit what we were trying to accomplish so we had to scrap the algorithm. We are now using a neural network algorithm but we need to generate 3D images for it using a GAN (Generative Adversarial Network) and this application could constrain our product further.

1.2.3. Future Contingencies

Contingencies that may arrive in the future are in Unsuitable machine learning AI suggestions and Machine Learning Incompatibility. Unsuitable machine learning (or ML for short) is the possibility that when a suggestion, that is either bugged or off the set parameters, is made, the image created will be incorrect. This would require rigorous testing to identify areas of weaknesses in the code that caused this issue. This has a higher likelihood of occurring now that we need to use a GAN to process 3D images. The GAN could produce an error and the AI could compound on that error as well so it is important that the GAN produces images correctly consistently.

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 system, 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 on the nose usually for cosmetic purposes
FDA (*abbrev.*)	Food and Drug Association
GAN (*abbrev.*)	Generative Adversarial Network
CNN (*abbrev.*)	Convolutional Neural Network
ML (*abbrev.*)	Machine Learning
AI (*abbrev.*)	Artificial Intelligence

UI (*abbrev.*)	User Interface

2. SYSTEM ARCHITECTURE

2.1. System Software Architecture

2.1.1 User Interface

- The user will run programs from and view output in the command prompt/terminal window.
- The user will utilize Blender to load a 3D face with their chosen landmarks and manipulate the 3D model of the patient's nose.

2.1.2 Database

- An iOS device is used to download Bellus3D, the scanning application, from the Apple App Store.
- The application is then used to scan the patient's and volunteer's faces to create a database. The scan is then sent as an .obj file to the Digitized Rhinoplasty web application.
- A GAN application will be used to increase the faces in our database. This will allow the AI to create better suggestions through a much larger sample size.
- The Digitized Rhinoplasty Application is used to plot the points on the patients'/volunteers' faces manually. After positioning the specified points on the nose the JSON files are exported from the web 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 the Option 9 project team member, will input their face scan into the Bellus3D application on an Apple device with iOS 11.1 or later with TrueDepth camera working.

- The input to the Digitized Rhinoplasty Web application is the scanned image as an .obj, jpeg and .mtl file.

- On the Digitized Rhinoplasty Web application, the system operator will rename the files with patients' names and point 50 landmarks on the said face manually.

- The Blender application currently takes in 3 files: The exported JSON file from the Rhinoplasty web application for both the dataPointsFile_path and the newDataPointsFile_path, the patient's obj file for the faceObjectFile_path, and the patient's Scaninfo folder for the meshFile_path.

3.2. Outputs

- The Bellus3D application outputs the scanned face with .obj, mtl and jpeg files.

- After the landmarks are implemented manually, the Rhinoplasty Digitized Web application outputs a JSON file with landmarks.

4. DETAILED DESIGN

4.1. Hardware Detailed Design

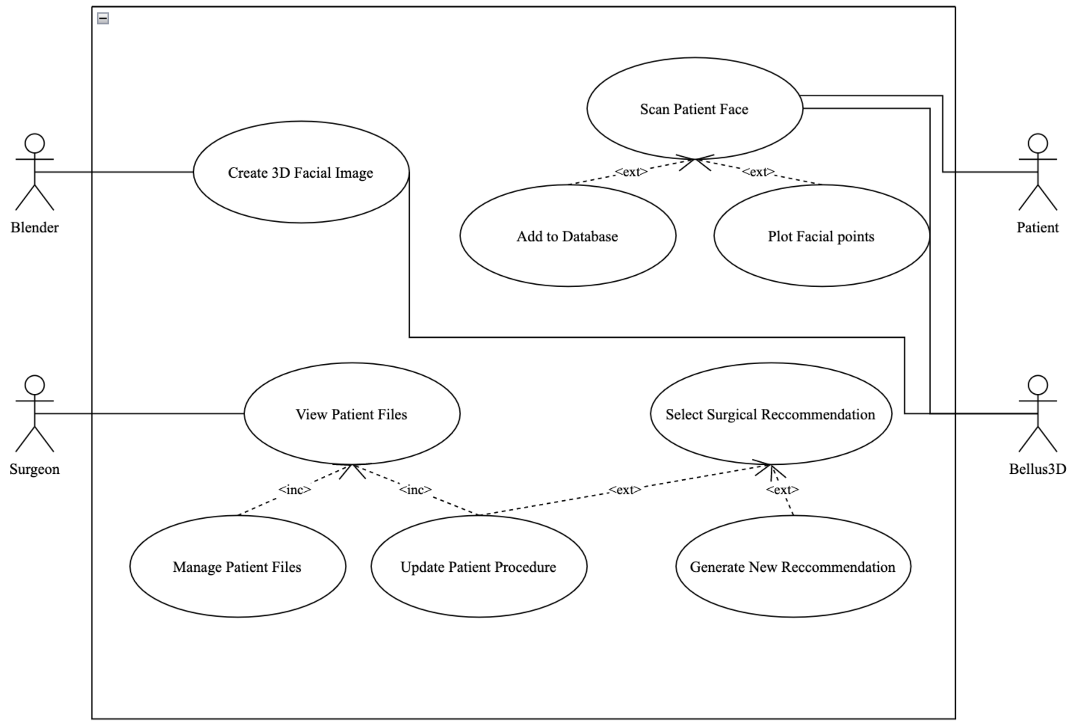
The hardware component needed for the Digitized Rhinoplasty project is an Apple device with iOS 11.1 or later and TrueDepth camera working. The scanning application, Bellus3D, is only compatible with an iOS device that has a working TrueDepth camera. This results in having a high quality 3D .obj image.

4.2. Software Detailed Design

- The scanned face with Bellus 3D application will be uploaded to the Rhinoplasty Digitized Web application as a .obj, jpeg and mtl file.
- The Rhinoplasty Digitized Web application will output a JSON file.
- The user then uploads the JSON file into the read_in.py, which will add the patient to the database and/or set the patient as the current one; as well as generate a CSV file from the JSON file.
- The software 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 CNN algorithm will be used to group a patient's 2D face with a group of similar faces that resemble their own.
- The user will choose one of these patient's noses depending on the similarities they have with the database.
- Blender will allow a doctor to alter the nose on the patient's face.

4.2.1. Software Analysis Models

In Figure 2, the system use case diagram is presented with the basic functionality design of the Digitized Rhinoplasty system. This was used throughout the requirements elicitation process.



4.3. Internal Communications Detailed Design

First, the current patient's csv file is read and the 14 needed points are pulled. The next step in the algorithm is to compare the patient's face to the database of face ratios to generate the top 10 most similar faces for each patient, and create groupings based on these results. These face groupings, along with the patient's face jpg, are then used by the CNN algorithm to display a group of faces that look similar to the patient's face for a natural looking suggestion to their nose.

5. EXTERNAL INTERFACES

5.1. Interface Architecture

The external system used by the Digitized Rhinoplasty project system is currently just an excel file, which is used to generate its database of faces. Users are able to access the face database by utilizing a program such as Excel, Notepad, Visual Studio Code, etc. The user will be able to execute the machine learning algorithm and compare before and after 3D models of the patient's face in the 3D modeling software Blender.

5.2 Interface Detailed Design

The interface utilizes the command prompt to run the python files. Additionally, the user will be able to execute the machine learning algorithm and model the patient’s face in the 3D modeling software Blender.