System Requirements Specification

for

Digitized Rhinoplasty

Version 1.0 approved

Prepared by Priscilla Carbo, Marvin Cazeau, Jared Curtis, Maree Kelly, Alessandra Oo

Embry Riddle Aeronautical University, Daytona Beach Campus

16 September 2021

Table of Contents

Revision History

Version	Name	Date	Reason for Changes
v1.0	P.C., M.C., J.C., M.K., A.O.	16 September 2021	Formatted template, added introduction
v1.1	P.C., M.C., J.C., M.K., A.O.	28 September 2021	Completed introduction, added references.
v1.2	J.C., M.K, M.C	26 October 2021	Added document conventions, added overall description, reformatted document

1. Introduction

1.1 Purpose

The software requirements specification for Digitized Rhinoplasty v1.1 covers the general system requirements, scope of the project, and intended use for all applicable audiences.

1.2 Document Conventions

We did not use any document conventions while writing this System Requirements Specification.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, users, testers, customers, and product owners alike to provide insight into the specifications and scope of the project. The suggested sequence for reading the document is beginning with the overall description and overview sections then proceeding through the sections that are most pertinent to the user.

1.4 Product Scope

The product being designed is software that will have the capability to take in the 3-D input of a client's face and add that to a database. It will then use an algorithm that will give the client x amount of options for a new face. This will be done by setting anchor points on the clients face that can move these specified areas around to generate several looks. When the client chooses an option, the selection will be sent to Blender. Blender will then output a 3-D image of the new face completely and compare both the old and new versions. If the client likes and confirms their decision, the relevant data that Rhinoplasty Surgeons would need will be output for their use so they can make the exact changes necessary.

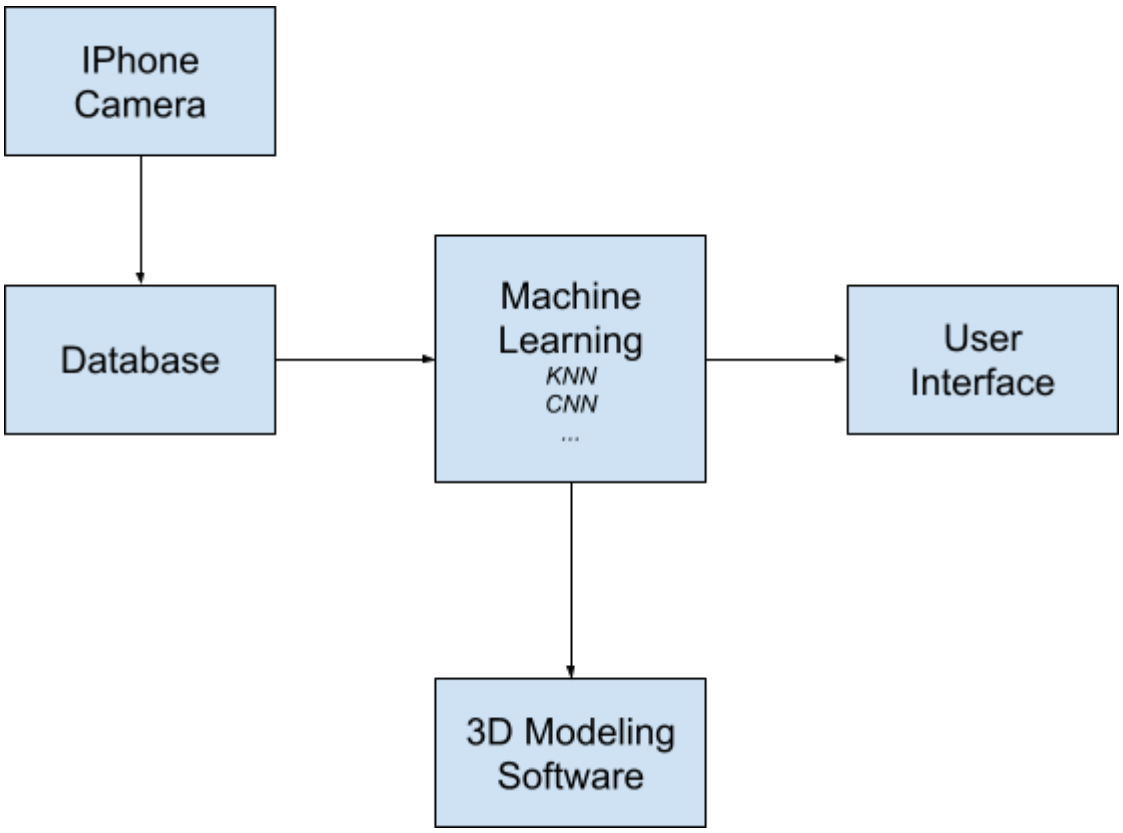
1.5 References

I. "Digitized Rhinoplasty." *Digitalization of Rhinoplasty Research*, https://digitized-rhinoplasty.com/.
    A. This is a web application intended to digitize the rhinoplasty by novel methodologies for pre-surgery planning, intraoperative guidance, and post-surgery evaluations.

2. Overall Description

2.1 Product Perspective

Digitized Rhinoplasty is a system designed to aid surgeons and patients who are about to undergo rhinoplasty surgery. It is an update to the Nova Initia system which aimed to do the same thing. This system requires some hardware and multiple pieces of software in order to function, which is modeled below.



2.2 Product Functions

- Interpret 3D scan of a user's face.
- Compare the 3D scan of a user's face to a database of other 3D Face Scans.
- Use machine learning to choose similar faces and suggest changes to the nose.
- Allow a user to open and edit a 3D model of a face in Blender.
- Export new nose mesh.

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

The first assumption about this product is that the software will run on Windows 10.0 and Mac OS 10.15.6 or later operating systems.

The second is that the surgeon will access the program as a local software.

The third assumption is about the software needed to run with the application. The user must have Blender 2.90 or later installed on their system

The fourth assumption is that the openpyxl module is installed to Blender's python module directory (C:\Program Files\Blender Foundation\Blender 2.90\2.90\scripts\modules). If openpyxl is not installed there is a tutorial on the github called "Installing openpyxl to Blender"

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:
REQ-2:

4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable

design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

The only authorized users for Nova Initia are licensed rhinoplasty surgeons and the developers of the system. The plastic surgeon will be required to verify their license by providing their certification to the system developers before accessing the system.

5.3.1 The system shall only allow access to authorized users.

5.3.2 The system shall save patient data in a secure SQL database.

5.3.3 The system shall isolate patient data from each individual patient file

5.3.4 The system shall backup every time new patient data is uploaded.

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>