

System Test Plan
For
Facial Analyzer for Rhinoplasty Operation

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

Version/Author	Date
Version 1	16 September 2021
Version 2	19 October 2021

Table of Contents

1.	Introduction	2
1.1	Purpose	2
1.2	Objectives	2
2.	Functional Scope	2
3.	Overall Strategy and Approach	2
3.1	Testing Strategy	2
3.2	System Testing Entrance Criteria	2
3.3	Testing Types	2
3.4	Suspension Criteria and Resumption Requirements	3
4.	Execution Plan	3
4.1	Execution Plan	3
5.	Traceability Matrix & Defect Tracking	3
5.1	Traceability Matrix	3
5.2	Defect Severity Definitions	3
6.	Environment	4
6.1	Environment	4
7.	Assumptions	4
8.	Risks and Contingencies	4
9.	Appendices	4

1. Introduction

1.1 Purpose

This document is a test plan for Facial Analyzer for Rhinoplasty Operations System Testing, produced by the development team. It describes the testing strategy and approach to testing the team will use to verify that the application meets the established requirements of the business prior to release.

1.2 Objectives

- Meets the requirements, specifications and the Business rules.
- Supports the intended business functions and achieves the required standards.
- Satisfies the Entrance Criteria for User Acceptance Testing.

2. Functional Scope

The Modules in the scope of testing for the Facial Analyzer for Rhinoplasty Operation System Testing are mentioned in the document attached in the following path:

1. The System Requirement Specification Documents:
<https://github.com/jcortis664/FacialAnalyzer/blob/main/Deliverables/System%20Requirements%20Specification%20Template.pdf>
2. User Manuals:
<https://github.com/jcortis664/FacialAnalyzer/tree/main/Deliverables>
3. Section 3.1: Testing Strategy

3. Overall Strategy and Approach

3.1 Testing Strategy

Facial Analyzer for Rhinoplasty Operation System Testing will include testing of all functionalities that are in scope (Section 2) identified. System testing activities will include the testing of new functionalities, modified functionalities, screen level validations, work flows, functionality access, testing of internal & external interfaces.

3.1.1 Function Testing

Test Objective: Confirm the functionality of the data entry, manipulation, machine learning, and output of the digitized rhinoplasty programs.

Technique: Execute multiple use cases to confirm that the given output and expected output match.

Completion Criteria: All use cases have been tested and provide the expected result.

Special Consideration: Access to the digitized rhinoplasty system and related system requirements document.

3.1.2 Database Testing

Test Objective: Confirm the functionality and usability of the database of face scans and the database of face ratios.

Technique: Upload multiple face scans and JSON files to the database and test these file's usability in the database and machine learning algorithms.

Completion Criteria: The database functions properly when utilized by all machine learning algorithms.

Special Consideration: Access to the database of faces as well as access to the website 'digitized-rhinoplasty.com'.

3.1.3 Performance Testing

Test Objective: Ensure the newly implemented algorithm can read, calculate, and write values to the XLSX file.

Technique: Execute the new algorithm, compute values read in XLSX, and write new values to a new XLSX file.

Completion Criteria: New points should be able to be read in the program Blender. These points must also meet the requirements for what the user believes will look good on their face.

Special Consideration:

3.2 System Testing Entrance Criteria

In order to start system testing, certain requirements must be met for testing readiness. The readiness can be classified into:

3.3 Testing Types

3.3.1 Usability Testing

User interface attributes, cosmetic presentation and content will be tested for accuracy and general usability. The goal of Usability Testing is to ensure that the User Interface is comfortable to use and provides the user with consistent and appropriate access and navigation through the functions of the application (e.g., access keys, consistent tab order, readable fonts etc.)

3.3.2 Functional Testing

The objective of this test is to ensure that each element of the component meets the functional requirements of the business as outlined in the:

- Business / Functional Requirements
- Business rules or conditions
- Other functional documents produced during the course of the project i.e. resolution to issues/change requests/feedback

3.4 Suspension Criteria and Resumption Requirements

This section will specify the criteria that will be used to suspend all or a portion of the testing activities on the items associated with this test plan.

3.4.1 Suspension Criteria

Testing will be suspended if the incidents found will not allow further testing of the system/application under-test. If testing is halted, and changes are made to the hardware, software or database, it is up to the Testing Manager to determine whether the test plan will be re-executed or part of the plan will be re-executed.

3.4.2 Resumption Requirements

Resumption of testing will be possible when the functionality that caused the suspension of testing has been retested successfully.

4. Execution Plan

4.1 Execution Plan

The execution plan will detail the test cases to be executed. The Execution plan will be put together to ensure that all the requirements are covered. The execution plan will be designed to accommodate some changes if necessary, if testing is incomplete on any day. All the test cases of the projects under test in this release are arranged in a logical order depending upon their inter dependency.

The test plan for the system is as follows:

4.1.1 Database Testing (See 3.1.2)

4.1.2 Function Testing (See 3.1.1)

4.1.3 Performance Testing (See 3.1.3)

5. Traceability Matrix & Defect Tracking

5.1 Traceability Matrix

List of requirement, corresponding test cases.

5.2 Defect Severity Definitions

Critical	<p>The defect causes a catastrophic or severe error that results in major problems and the functionality rendered is unavailable to the user. A manual procedure cannot be either implemented or a high effort is required to remedy the defect. Examples of a critical defect are as follows:</p> <ul style="list-style-type: none">• System abends• Data cannot flow through a business function/lifecycle• Data is corrupted or cannot post to the database
Medium	<p>The defect does not seriously impair system function and can be categorized as a medium Defect. A manual procedure requiring medium effort can be implemented to remedy the defect. Examples of a medium defect are as follows:</p> <ul style="list-style-type: none">• Form navigation is incorrect• Field labels are not consistent with global terminology
Low	<p>The defect is cosmetic or has little to no impact on system functionality. A manual procedure requiring low effort can be implemented to remedy the defect. Examples of a low defect are as follows:</p> <ul style="list-style-type: none">• Repositioning of fields on screens• Text font on reports is incorrect

6. Environment

6.1 Environment

- The System Testing environment will be used for System Testing.

In order to conduct the testing the tester needs to have the following installed onto their computer:

- Blender V2.9 & up
- Access to a database of faces
- Python V3.8 & up

7. Assumptions

This section lists the assumptions for this project.

- User is a licensed medical professional
- All files are in the correct file format
- User must use Blender
- Blender must be modified in order to function correctly

- Clients will submit pictures with no facial expressions & correctly to render a 3D image
- No face tattoos/piercings

8. Risks and Contingencies

Risk #	Risk	Impact	Contingency Plan
1	Unsuitable Suggestion	High	Making sure different scenarios are tested rigorously to insure this happens as infrequently as possible
2	Machine Learning Incompatibility	High	Testing deployment in many scenarios to ensure the ML AI can consistently produce results in the suitable environments

9. Appendices