**System Test Plan**

**For**

***Nova Initia***

Sarai Toloza

Jacob Gattuso

Hailey DeNys

Yara AlRashidi

Emily Nunez

|  |  |
| --- | --- |
| Version/Author | Date |
| 1.1 Sarai Toloza | 11/3/2021 |
| 1.1 Emily Nunez | 11/3/2021 |
|  |  |
|  |  |
|  |  |

Table of Contents

[1.](#_heading=h.30j0zll) Introduction 2

[1.1](#_heading=h.1fob9te) Purpose 2

[1.2](#_heading=h.3znysh7) Objectives 2

[2.](#_heading=h.2et92p0) Functional Scope 2

[3.](#_heading=h.tyjcwt) Overall Strategy and Approach 2

[3.1](#_heading=h.3dy6vkm) Testing Strategy 2

[3.2](#_heading=h.1t3h5sf) System Testing Entrance Criteria 4

[3.3](#_heading=h.4d34og8) Testing Types 4

[3.4](#_heading=h.2s8eyo1) Suspension Criteria and Resumption Requirements 4

[4.](#_heading=h.17dp8vu) Execution Plan 5

[4.1](#_heading=h.3rdcrjn) Execution Plan 6

[5.](#_heading=h.26in1rg) Traceability Matrix & Defect Tracking 7

[5.1](#_heading=h.lnxbz9) Traceability Matrix 7

[5.2](#_heading=h.35nkun2) Defect Severity Definitions 8

[6.](#_heading=h.1ksv4uv) Environment 8

[6.1](#_heading=h.44sinio) Environment 8

[7.](#_heading=h.z337ya) Assumptions 8

[8.](#_heading=h.3j2qqm3) Risks and Contingencies 9

# Introduction

## Purpose

This document is a test plan for Nova Initia System Testing, produced by the System Testing 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.

## Objectives

* Meets the requirements, specifications, and Business rules.
* Supports the intended business functions and achieves the required standards.
* Satisfies the Entrance Criteria for User Acceptance Testing.
* Follows internal security standards
* Complies with all health and safety regulation to ensure the well being of the patient

# Functional Scope

The Modules in the scope of testing for the Nova Initia System Testing are mentioned in the documents attached in the following path:

1. The System Requirements Specification document: <https://docs.google.com/document/d/1WSCZzfAqJp7yq9Y2FEHEJKG2nweouBcuVMc-nhQ2yb0/edit>
2. User manuals:

<https://github.com/saraitoloza/NovaInitia2/tree/main/Deliverables>

1. Section 3.1 of this document

# Overall Strategy and Approach

## Testing Strategy

Nova Initia System Testing will include testing of all functionalities that are in the scope (section 2) identified. System testing activities will include the testing of new functionalities, modified functionalities, screen level validations, workflows, functionality access, testing of internal & external interfaces.

The testing types section following this one will be discussing what needs to be tested. However, this section will describe how the types will be tested.

**3.1.1 Function Testing**

**Test Objective**: The application’s navigation data entry, processing, and retrieval work according to the specific requirements in the SRS

**Technique:** Execute use cases from the use case diagram -> when valid data is given then the corresponding result is given, when invalid data is inputted then the expected warning message should show

**Completion Criteria:** When all use cases have been tested and all defects have been mitigated

**Special Consideration**: Access to the Nova Initia system, and the corresponding System Requirements Specification document

**3.1.2 Database Testing**

**Test Objective**: Upload the image on the digitized application where the points will be placed

and saved as a JSON file.

**Technique:** Execute the Python code and convert the JSON file to XLSX for blender. Then check the source folder to ensure the XLSX file as output.

**Completion Criteria:** When there is an output XLSX file that contains all the organized correct patient face data.

**3.1.2 Performance Testing**

**Test Objective**: Ensure k-N N algorithm can read, calculate, and write values to XLSX.

**Technique:** Execute the k-N N algorithm, compute values read in XLSX, and write new values to a new XLSX.

**Completion Criteria:** New points should be able to be read in Blender. These points must also meet the requirements for a perfect nose.

## System Testing Entrance Criteria

In order to start system testing, certain requirements must be met for testing readiness. The readiness can be classified into usability testing, functional testing, and data and documentation testing.

## Testing Types

### Usability Testing

User interface attributes, cosmetic presentation, and the 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.)

System Requirements Specification, 3.2.1 The system shall display a login screen

System Requirements Specification, 3.2.2 The system shall display a home screen

System Requirements Specification, 3.2.3 The system shall display a preview a 3D face with facial points

System Requirements Specification, 3.2.4 The system shall allow input of new patient data

System Requirements Specification, 3.2.5 The system shall output new user data to database

System Requirements Specification, 3.2.6 The system shall execute Blender for manual face manipulation

System Requirements Specification, 3.2.7 The system shall execute the k-NN algorithm when the user triggers it

System Requirements Specification, 3.2.8 The system shall allow the user to input images

System Requirements Specification, 3.2.9 The system shall output the patient “before image” to Blender

### 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

System Requirements Specification, 3.1.1: “The system shall take in the quantitative measurements of the face scan”

System Requirements Specification, 3.1.2: “The system shall allow the user to create a 3D image of the new nose on the patients scanned face”

System Requirements Specification, 3.1.3: “The system shall display the 3D image of the patients face with the new nose”

System Requirements Specification, 3.1.4 The system shall use the k-NN algorithm to pick the k nearest neighbors.

System Requirements Specification, 3.1.5 The system shall display k options for the patient to choose from.

**3.3.3 Data and Documentation Testing**

Data and documentation cover all the user guides, installation guides, read me files, and set up a manual that is provided with the software to ensure that the user understands the Nova Initia system. The objectives of this type of testing: check if what is stated in the documents is available in the software and check if the explanation of the system is correctly explained in the documentation.

System Requirements Specification, 3.5.1: “The documentation will be online through the application”.

System Requirements Specification, 3.5.2: “The intended audience will be doctors and patients. Doctors should have previous background in surgical procedures”.

System Requirements Specification, 3.5.3: “The patient will receive the same feedback as the doctor as well as a blended image of their expected output”.

System Requirements Specification, 3.6.1: “Patients' face proportions will be calculated based on data calculated from the k- N N algorithm”.

System Requirements Specification, 3.6.2: “The database has ratios and XYZ points that will be used by the algorithm that are used to produce the new angles of the nose”.

System Requirements Specification, 3.6.4: “The software will then provide the surgeon with k options for the patient to choose from.”.

System Requirements Specification, 3.6.6: “The system shall send the patient’s choice to blender.”

System Requirements Specification, 3.6.7 The system shall create the patient’s new face in blender.

System Requirements Specification, 3.6.8 The system shall output the incisions for the surgeon to make on the patient’s face to give them their new nose.

## 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.

### Suspension Criteria

If the new XYZ points outputted from the algorithm do not align with the rules, then the algorithm is faulty and the test will be suspended. This could be caused by incorrect reading/writing from XLSX to an array and incorrect arithmetic.

For unit testing, we can test the values in the array to match the values in the CSV file. If this is false, the rest of the program will terminate. Before testing the system will verify the user ID and password before logging in, if ID or passwords do not match testing will be terminated.

### Resumption Requirements

The program will be able to re-execute testing once the programmer has verified array values corresponding with the correct XLSX values.

# Execution Plan

## 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 interdependency.

The test plan for the Nova Initia 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)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement (From SRS) | Test Case Identifier | Input | Expected Behavior | Pass / Fail |
| 3.2.1: “Patients' face proportions will be scanned using the Bellus3D application” | 10.1 | application is ran | The proportions are scanned | Pass |
| 3.2.2: The system shall take in a JSON file created from the patient face scan web application as an input | 1.1 | User types in the desired patient’s JSON file name when prompted after running the Python code. | The code runs without error and outputs a XLSX file with the patient’s information | Pass |
| 3.2.1: “The system shall take in the quantitative measurements of the face scan” | 5.1 | The k- N N algorithm is run along with the .XLSX files. | New angle measurements and ratios are calculated. | Pass |
| 3.2.1: “The system shall allow the user to create a 3D image of the new nose on the patients scanned face” | 6.1 | The Blender file is ran | The blender opens and the 3D model of the patients face is displayed | Pass |
| 3.1.1: “The system shall display the 3D image of the patients face with the new nose” | 7.1 | The Blender file is ran | The patient's 3D model of their face is displayed with the new nose | Pass |
| 3.1.2: “The system shall output k options for the patient to choose from” | 8.1 | k-N N Algorithm is ran | A file is created with the k options of noses similar to the patient. | Pass |
| 3.4.1: “The intended audience will be doctors and patients. Doctors should have previous background in surgical procedures”. | 9.1 | The user inputs a valid surgeon ID before accessing the software | The user is granted access to the software | Pass |
| 9.2 | The user inputs an invalid surgeon ID before accessing the software | The user is denied access to the software | Pass |
| 3.4.2: “The database has points and ratios that will be used by the k-N N algorithm that are used to produce the new angles of the nose”. | 11.1 | Algorithm is ran | New angles of the nose are produced | Pass |
| 3.4.3: “The software will then provide the surgeon new points of the 3D model of the patient's face” | 12.1 | Algorithm is ran | The file containing the new points of the 3D model is created | Pass |
| 3.4.4: “ The system shall output the patient “before image” to Blender”. | 13.1 | The add points script is run in Blender | The old points on the face are displayed with spheres and cubes respectively. | Pass |

***Table 4.1.*** This table goes into detail about how to test each specific requirement from the System Requirements Specification document and includes what the expected result of the test should return.

# Traceability Matrix & Defect Tracking

## Traceability Matrix

List of requirement, corresponding test cases

***Requirement CRITICAL:*** System Requirements Specification, 3.2.3 The system shall display a preview a 3D face with facial points

***Test Cases:***

***Requirement CRITICAL:*** System Requirements Specification, 3.2.4 The system shall allow input of new patient data

***Test Cases:***

***Requirement CRITICAL:*** System Requirements Specification, 3.2.5 The system shall output new user data to database

***Test Cases:***

***Requirement CRITICAL:*** System Requirements Specification, 3.2.6 The system shall execute Blender for manual face manipulation”

***Test Cases:***

***Requirement CRITICAL:*** System requirements Specification, 3.6.1: System Requirements Specification, 3.1.1: “The system shall take in the quantitative measurements of the face scan”

***Test Cases:***

***Requirement CRITICAL:*** System requirements Specification, 3.6.2: “The database has XYZ points that will be used by the algorithm that are used to produce the new angles of the nose”.

***Test Cases:*** Check that algorithm produces XYZ values.

***Requirement CRITICAL:*** The system shall use the k-NN algorithm to pick the k nearest neighbors.

***Test Cases:***

***Requirement CRITICAL:*** System requirements Specification, 3.6.4: “The software will then provide the surgeon with k options of patient's face for the patient to choose from”.

***Test Cases:***

***Requirement MEDIUM:*** System requirements Specification, 3.6.5: “The software will display both the original nose points and the new nose points with different colored shapes”.

***Test Cases:*** Check that both points (pre algorithm and post algorithm) are displayed in Blender.

## Defect Severity Definitions

|  |  |
| --- | --- |
| **Critical** | 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:   * System abends * Data cannot flow through a business function/lifecycle * Data is corrupted or cannot post to the database |
| **Medium** | 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:   * Form navigation is incorrect * Field labels are not consistent with global terminology |
| **Low** | 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:   * Repositioning of fields on screens * Text font on reports is incorrect |

# Environment

## 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:

* Eclipse Version 4.7.3 and up
* Nova Initia System
* Blender Version 2.90.1 and up
* Access to the database of patients
* Python Spyder Version 4.2.1

# Assumptions

This section list assumption that is made specific to this project

* The user should be a licensed surgeon
* All images and data points being stored in the database are the correct file format (.obj and .xlsx respectively)
* The user must have Blender (2.90.1 or newer) installed
* Blender must have the openpyxl module installed to its python library in order to function correctly (If openpyxl is not installed to Blender please see the “Installing Openpyxl to Blender” tutorial on the project’s on GitHub in the Deliverables folder)

# Risks and Contingencies

|  |  |  |  |
| --- | --- | --- | --- |
| Risk # | Risk | Impact | Contingency Plan |
| 1 | Calculations incompatible with customers desires | High | Testing the product for different rhinoplasty needs and scenarios to ensure the best possible results prior to deployment. |
| 2 | Incompatible compiler | High | Testing deployment can run smoothly with various IDEs including but not limited to Eclipse and VSCode. |

# 

# 