**TEST PLAN FOR**

**Skin N Sense**



*ChangeLog*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| version number | Date of Change | Name of person who made changes | Description of the changes made |
| 001 | 03.11.2023 | Tryamb Sachan | Initial Draft |
|  |  |  |  |
|  |  |  |  |

[1 INTRODUCTION 2](#_Toc8301)

[1.1 SCOPE 2](#_Toc8302)

[1.1.1 In Scope 2](#_Toc8303)

[1.1.2 Out of Scope 2](#_Toc8304)

[1.2 QUALITY OBJECTIVE 3](#_Toc8305)

[1.3 ROLES AND RESPONSIBILITIES 3](#_Toc8306)

[2 TEST METHODOLOGY 3](#_Toc8307)

[2.1 OVERVIEW 3](#_Toc8308)

[2.2 TEST LEVELS 4](#_Toc8309)

[2.3 TEST COMPLETENESS 4](#_Toc8310)

[3 TEST DELIVERABLES 4](#_Toc8311)

[4 RESOURCE & ENVIRONMENT NEEDS 5](#_Toc8312)

[4.1 TESTING TOOLS 5](#_Toc8313)

[4.2 TEST ENVIRONMENT 5](#_Toc8314)

[5 TERMS/ACRONYMS 6](#_Toc8315)

# 1 Introduction

Skin N Sense presents the development and implementation of an Android application aimed at enhancing healthcare through the integration of machine learning models, image segmentation, and location-based services. The application leverages TensorFlow Lite (TFLite) models for two critical tasks: skin lesion analysis and mood and emotion recognition. For skin lesion analysis, a Convolutional Neural Network (CNN) is employed for image segmentation, allowing for precise detection and classification of skin conditions. When a skin image is provided, the model accurately predicts potential skin diseases, enabling early diagnosis. In the domain of mental health, facial images are processed to assess mood and emotion. The application employs a TFLite model to analyze facial expressions and provide valuable insights into the user's emotional state. This holistic approach recognizes the intrinsic connection between physical and emotional well-being. Furthermore, the system utilizes custom-created API to identify nearby healthcare providers, allowing users to locate doctors specializing in relevant medical fields.

## 1.1 Scope

1.1.1 In Scope

Scope defines the features, functional or non-functional requirements of the software that **will be** tested. Features of the Project:

1. Image Upload and Recognition: Functional Requirements - Test the functionality of uploading images.

* Including Image Verification (in case of facial images).
* Ensure the recognition and identification accuracy of medicinal plants and materials.

2. Detailed Descriptions: Non-Functional Requirements

- Verify that detailed descriptions of recognized plants/materials are provided accurately.

3. Deep Learning Recognition: Functional Requirements

- Test the deep learning recognition algorithms for accuracy and reliability.

4. Machine Learning Model: Functional Requirements

- Ensure the machine learning model (e.g., CNN with segmentation) performs accurately for image classification.

5. User-Friendly Interface: Non-Functional Requirements

- Usability testing to evaluate the user interface for intuitiveness and ease of interaction.

6. Early Stopping Mechanism: Functional Requirements

- Test the early stopping mechanism in the machine learning model to prevent overfitting.

1.1.2 Out of Scope

Out Of Scope defines the features, functional or non-functional requirements of the software that **will NOT be** tested:

1. Scalability and Cloud Hosting: While the project design accounts for scalability, extensive testing and validation for large-scale cloud hosting environments are not included in this scope. This means that the application's performance and resource utilization under cloud hosting conditions are beyond the immediate testing objectives.
2. Custom Knowledge Database: The project does not involve the development of a custom knowledge database for the Android application. While the app utilizes pre-existing knowledge and models for skin disease detection and mood prediction, the creation, management, and expansion of a custom knowledge database for other purposes are outside the current scope.

## 1.2 Quality Objective

## The project underwent rigorous quality assurance measures to ensure its accuracy, performance, usability, and overall effectiveness. These efforts encompassed various aspects:

## Accuracy Assessment: The accuracy of both the skin disease detection and mood prediction models was meticulously evaluated. This involved extensive testing with diverse datasets to measure the models' ability to provide precise results. Multiple evaluation metrics, including classification accuracy were considered to ensure accurate predictions.

## Performance Optimization: The application's performance was a top priority. Extensive testing and optimization were conducted to enhance the app's responsiveness, reduce latency, and minimize computational resource usage. Performance testing was carried out under various conditions to ensure that the app runs smoothly on a wide range of Android devices.

## Usability Testing: The application underwent usability testing to gauge its user-friendliness and overall user experience. Feedback from real users was collected to make necessary improvements to the user interface, navigation, and interaction design. This process aimed to create an intuitive and accessible user experience.

## Security and Privacy: Stringent security and privacy measures were implemented to safeguard user data and maintain data confidentiality. User data protection, secure data transmission, and adherence to privacy regulations were of paramount importance throughout the development process.

## Robustness and Reliability: The application was stress-tested to ensure its robustness and reliability under various scenarios. This included testing for potential errors, crashes, and unexpected behaviors, with the goal of delivering a stable and dependable application.

## Scalability: The architecture and design of the application were developed with scalability in mind. The application was tested to ensure it can accommodate potential future expansions, including the integration of additional features and models.

## User Feedback Incorporation: User feedback was actively solicited and taken into account for further enhancements. Continuous improvement was an integral part of the quality assurance process.

## 1.3 Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

* QA Analyst: Tryamb Sachan
* Test Manager: Prof. Shreela Pareek
* Configuration Manager: Prof. Neha Shukla
* Developers: Tryamb Sachan, Suryansh Awasthi, Shivam Singh
* Installation Team: Prof. Shreela Pareek, Prof. Neha Shukla, Tryamb Sachan, Suryansh Awasthi, Shivam Singh

# 2 Test Methodology

## 2.1 Overview

We are using an iterative testing approach to make sure our project works well. This means we test it in small steps, starting with checking if each part works on its own. Then, we see how different parts work together.

We keep testing as we make changes and add new things. This way, we make sure our project is always working well, even after modification.

## 2.2 Test Levels

**Test Levels define the Types of Testing to be executed on the Application Under Test (AUT**).

We aim to test our project at the following levels:

1. Unit Testing: This is the lowest level of testing and focuses on individual components or functions within the software. Developers often perform unit tests to verify that specific parts of the code work correctly.

1. Integration Testing: This level of testing checks how different components or modules of the software work together. It ensures that integrated parts of the software function as intended.

1. System Testing: At this level, the entire system is tested as a whole. It verifies that the software meets its specified requirements and functions properly in its intended environment.

## 2.3 Test Completeness

Here you define the criterias that will deem your testing complete.

For instance, a few criteria to check Test Completeness would be

* 100% test coverage
* All Manual & Automated Test cases executed
* All open bugs are fixed or will be fixed in next release

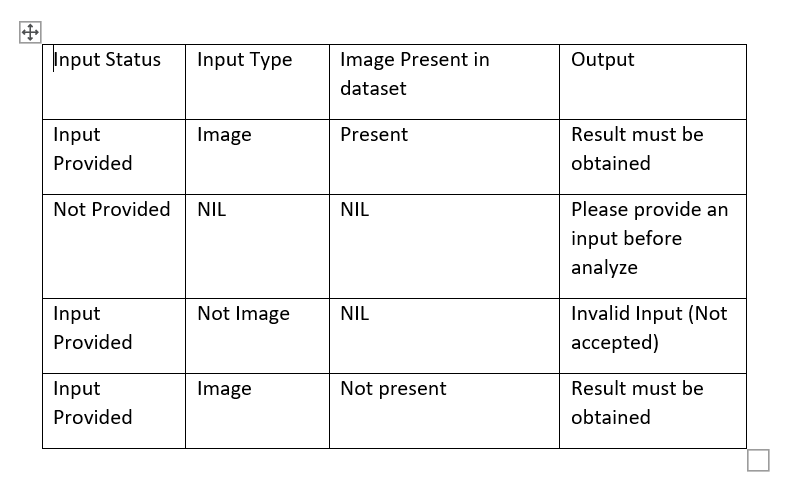
# 3 Test Deliverables

Here are the deliverables

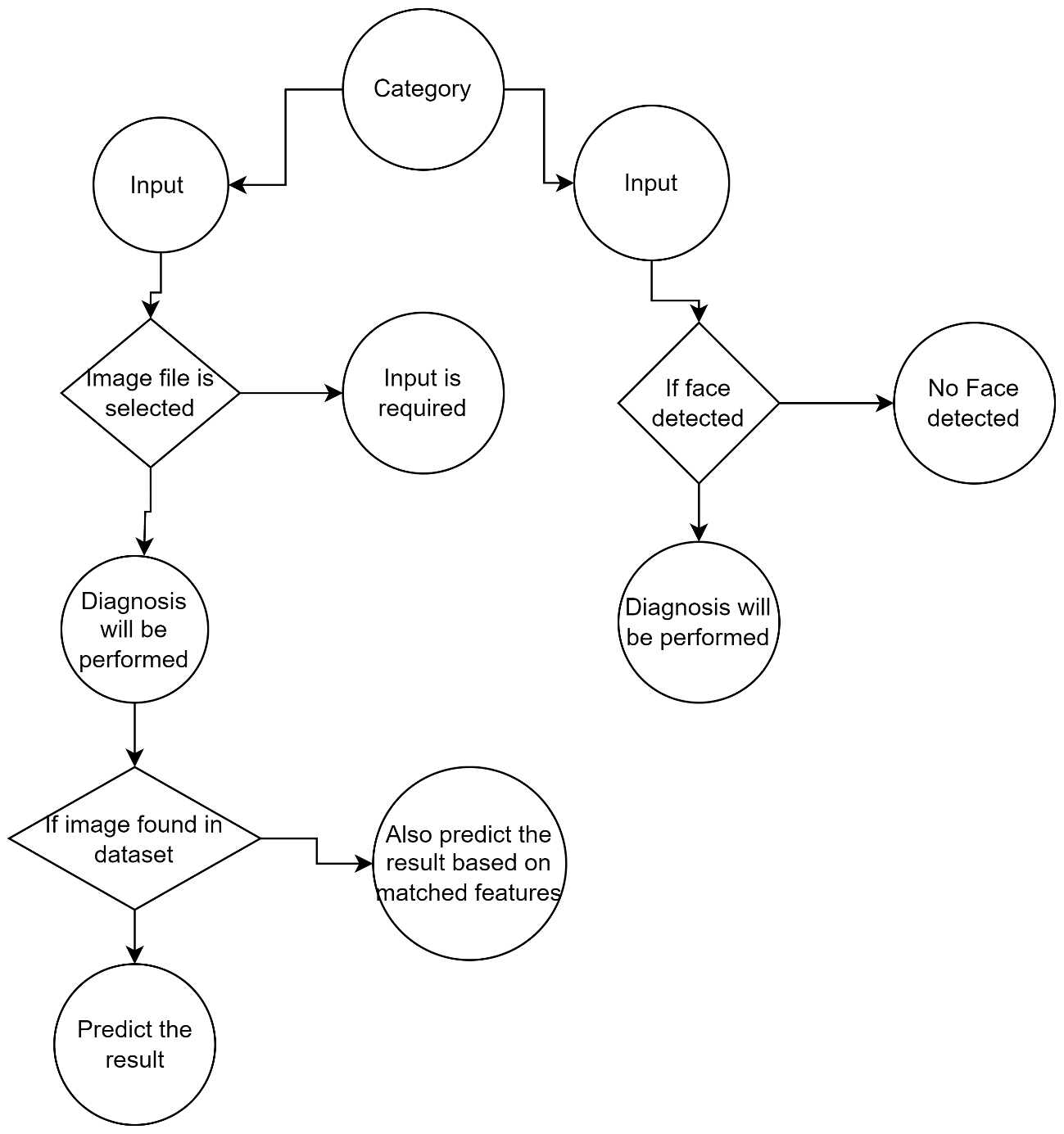
* Test Plan
* Test Cases
* Bug Reports
* Test Strategy

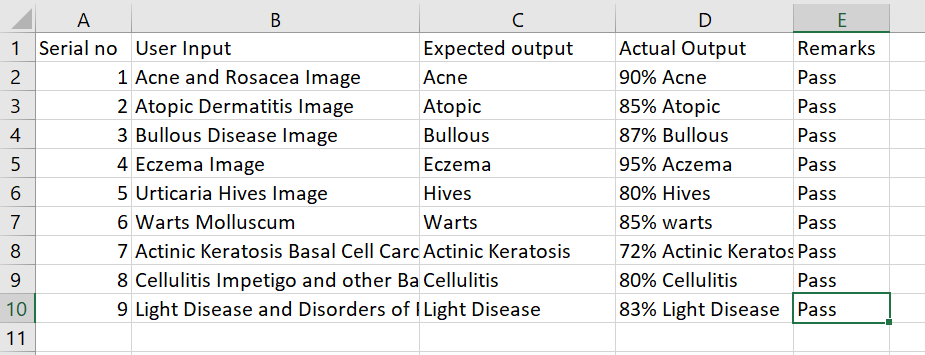
**4 Test Cases**

**Equivalence Class Partitioning:**

****

**Decision Tree:**





# 5 Resource & Environment Needs

## 5.1 Testing Tools

**Black-box testing:** I focus on the application's external behavior from a user's perspective. I interact with the app's user interface to verify that it functions as intended, checking for usability, navigation, and overall performance. This testing approach helps identify issues that users might encounter while using the application.

**White-box testing**: On the other hand, allows me to dive into the internal structure of the software. I analyze the code, algorithms, and system architecture to uncover any vulnerabilities or bugs that may exist within the application. By examining the codebase and logic, I aim to ensure that the application is not only functional but also secure and reliable, providing a comprehensive assessment of its overall quality.

List of Tools like

* Postman API

## 5.2 Test Environment

It mentions the minimum **hardware** requirements that will be used to test the Application.

Following **software’s** are required in addition to client-specific software.

* Windows 10 and above preferred
* VSCode 2022 or above preferred
* Android Studio Dolphin | 2021.3.1 (Sep 2022) or above
* Android device running Android version 8.0 (Oreo) or above.

# 6 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

|  |  |
| --- | --- |
| **TERM/ACRONYM** | **DEFINITION** |
| API | Application Program Interface |
| AUT | Application Under Test |