Hairdressing Project

**Test Plan**

*Purpose: This document outlines the types of tests to be carried out for this project, along with the scope of what needs to be tested, standards and how each test relates to required features described in the Software Requirements Specification and the Requirements Prioritisation document.*

**Team members: Diego G., Gerardo G.**

**Date/Version:**

**31/08/2020 - Version 0.1 (Draft)**

**04/09/2020 - Version 1.0 (Reviewed draft / Initial release)**

# Overview

## Purpose

The purpose of this document is to define a detailed document that describes the test strategy, objectives, schedule, estimation and deliverables, and resources required for testing.

The sections of the document are:

* The context of the testing
* The test methodology
* The test environment
* The types of tests
* The standards that will apply
* The test deliverables
* The test scope, including functional and non-functional tests.

## Context

This project involves the following stakeholders:

|  |  |
| --- | --- |
| Name | Role |
| Diego C. | Developer |
| Gerardo G. | Developer |
| Delia Stanley | Client |
| Dawn Hetherington | Client |
| Jade Uhrbom | Client |
| Keith Critchett | Client liaison |

The requirements requested by each client have been specified in the Project Plan and all requirements have been listed in the Software Requirements Specification and sorted in the Requirements Prioritisation (MoSCoW) document.

Developers will perform functional and non-functional tests, along with unit tests and performance tests, as well as black box tests (testing features from the user’s point of view, without taking the code in consideration). Clients may validate the tests after a report has been submitted including test results.

The application to be tested consists of a Python API and a Flutter mobile app.

The scope of the tests is focused on the main features of the application under development, namely:

* Uploading pictures
* Selecting hair characteristics, including hair style and colour
* Detecting face shapes
* Browsing the history of pictures submitted, along with their changes
* Validating data, including expected results from parameters chosen to change hair features and sign in/sign up forms
* Saving pictures locally
* Comparing original pictures to their respective changes

Such tests will ensure that all actions and features required to be implemented work as expected and with an acceptable degree of accuracy (particularly in the case of changing hair characteristics and detecting face shapes).

## 1.3. Methodology

This section explains the testing methodology that will be performed by the project team for the **Hair dress project**. It defines the overall testing approach and test activities. Its purpose is to document:

Since we only have a developer team of two, both developers will fill the Test manager, and Tester roles. System resources includes computer, network, server and several test tools.

Testing the mobile app and the API will be performed in parallel. All functionalities that fails to pass their respective test should be addressed and fixed within one week since the test was performed and before the next test cycle.

Risks

|  |  |
| --- | --- |
| Risk | Mitigation |
| Team member lack the required skills | Plan training time to skill up |
| The project schedule is too tight | Set Test Priority for each of the test activity |

Scrum in Agile methodology will be followed as it provides developers with the flexibility to adjust to changes of requirements, provide better estimates and be more in control of the project schedule and state.

## Test environment

The app will be developed primarily for Android phones, despite the code base being cross-platform. It connects to a Python API, which handles processing pictures uploaded by users and the business logic to change them according to user input. It is a pure REST API, so it does not serve web pages.

Provisioning hardware, software or network resources has already been taken care of by the developers, since no additional expenses will be necessary for the development of this project aside from AWS (which will not be an issue, as the lecturer Ivone has provided the development team $50 in credits to use such service) and Asana (again, not an issue because the lecturer Paul has granted premium access for the developers to collaborate on their tasks).

As for the platforms, tools, frameworks and development environment used, they are:

* Windows 10
* GNU/Linux (cloud environment)
* Visual Studio
* Visual Studio Code
* Android Studio
* Flutter (mobile app framework)
* Fast API (Python REST API framework)
* AWS (EC2, S3, Route53, RDS, DynamoDB)
* MariaDB (current relational database containing user data)
* ASP.NET Core (current REST API that communicates with the MariaDB database and the Admin Portal)

## Types of Tests

Test to be performed are:

* Functional test
* Non-functional test
* Unit test – UI Test
* API test
* Performance test
* Black box test
* White box test

As mentioned before, developers are responsible for performing all tests. Black box testing can be performed by clients and users.

## Test standards

The testing techniques and methodologies adopted in this project include:

* White box testing, with unit tests and several test cases to validate branching logic
* Black box testing, with equivalence partitioning (representative test cases) and boundary value analysis (extreme test cases)
* Basic performance tests, analysing and reporting CPU and RAM usage and response times

Although not in a strict manner, the five current international test standards will be taken into consideration, which constitute:

* ISO/IEC/IEEE 29119-1:2013: adopting common testing definitions
* ISO/IEC/IEEE 29119-2:2013: integrating test processes into the Agile model used in this project
* ISO/IEC/IEEE 29119-3:2013: including required documentation such as Test Plan, Test Results and Test Reports
* ISO/IEC/IEEE 29119-4:2015: using several test techniques as previously described, such as equivalence partitioning
* ISO/IEC/IEEE 29119-5:2016: implementing some keyword-driven tests

In order to ensure that the tests to be executed meet the quality standards of this project, they will be documented with a matching requirement from the Software Requirements Specification document and several inputs as test cases, then validated by another developer and the results will be recorded in a report document.

Critical failing tests will be prioritised over those that do not disrupt the user flow of the application.

## Test deliverables

Test performed will be reported in several documents. Deliverables includes:

* Test plan
* Bug reports
* Test Metrics

Developers will be signing off at the end of test cycles.

# Testing Plan

## Scope of Testing

### Functional tests

Functional tests with their respective requirements linked from the Software Requirements Specification are available in the following link:

<https://github.com/HairdressingProject/styleme/blob/master/Documentation/Tests.xlsx>

### Performance tests

Some of the performance metrics to be used while testing are:

* **Response time:** total time to send a request and get a response.
* **Wait time:** how long does it takes to receive the first byte after a request is sent.
* **Average load time:** the average amount of time it takes to deliver every request.
* **Peak response time:** the longest amount of time it takes to fulfill a request. A peak response time way longer than the Average load time may indicate an anomaly.
* **Error rate:** percentage of requests resulting in errors compared to all requests.
* **Concurrent users:** how many active users at any point (also known as load size).
* **Requests per second:** how many requests are handled
* **Throughput:** Measured by kilobytes per second, throughput shows the amount of bandwidth used during the test.
* **CPU, Memory, Disk, Input/Output:** It is the measure of the CPU, memory, disk and I/O usage when the application operates in a production-like environment.

**Load testing** measures system performance as the workload increases. That workload could mean concurrent users or transactions. The system is monitored to measure response time and system staying power as workload increases. That workload falls within the parameters of normal working conditions.

### Other non-functional tests

Non-functional tests have also been added to the previously linked spreadsheet:

<https://github.com/HairdressingProject/styleme/blob/master/Documentation/Tests.xlsx>

In order to capture performance metrics, the Flutter extension for VSCode will be used to run the app in profile mode (see <https://flutter.dev/docs/perf/rendering/ui-performance#run-in-profile-mode>). Response times, CPU and RAM usage of each action executed in the app will be registered in a separate document (to be disclosed by the developers once the results are gathered).

As for the Python API, the same approach will be used by implementing logging and analysing response times by making test requests through Postman.

### Out of scope

As 3D imagery (rotating heads) are out of scope, that feature is also out of scope from testing. Same goes for changing hair porosity, density and texture of pictures uploaded, although a “consultation” section containing sample images for each property will be provided and tested.