

# Verification and Validation Report: Sayyara

Team 3, Tiny Coders

Arkin Modi

Joy Xiao

Leon So

Timothy Choy

March 6, 2023

# 1 Revision History

Table 1: Revision History

Date	Developer(s)	Change
February 22, 2023	Arkin Modi	Add Automated Testing Section
March 3, 2023	Joy Xiao	Add Tests Evaluations for Legal and Cultural Requirements
March 4, 2023	Leon So	Add Introduction
March 4, 2023	Arkin Modi	Add Tests Evaluation for Security Requirements
March 4, 2023	Arkin Modi	Update Look and Feel Requirements Tests
March 4, 2023	Arkin Modi	Add Unit Testing Summary Table
March 4, 2023	Leon So	Add NFR Tests for Operational and Environmental Requirements
March 4, 2023	Timothy Choy	Add Test Evaluations for Usability and Humanity Requirements
March 4, 2023	Timothy Choy	Add Test Evaluations for Performance Requirements
March 5, 2023	Leon So	Add Functional Requirement Tests for Shop Profile
March 6, 2023	Leon So	Add Functional Requirement Tests for Authentication

## 2 Symbols, Abbreviations and Acronyms

symbol	description
API	Application Programming Interface
CD	Continuous Deployment
CI	Continuous Integration
CLS	Cumulative Layout Shift
FCP	First Contentful Paint
FID	First Input Delay
LCP	Largest Contentful Paint
ms	Milliseconds
PII	Personal Identifiable Information
PWA	Progressive Web Application
s	seconds
SRS	Software Requirements Specification
T	Test
TTFB	Time to First Byte
VnV	Verification and Validation

# Contents

<b>1</b>	<b>Revision History</b>	<b>i</b>
<b>2</b>	<b>Symbols, Abbreviations and Acronyms</b>	<b>ii</b>
<b>3</b>	<b>Introduction</b>	<b>1</b>
3.1	Project Summary . . . . .	1
3.2	Purpose of Document . . . . .	1
3.3	Scope of Testing . . . . .	1
<b>4</b>	<b>Functional Requirements Evaluation</b>	<b>1</b>
4.0.1	Authentication . . . . .	1
4.1	Shop Profile . . . . .	4
<b>5</b>	<b>Nonfunctional Requirements Evaluation</b>	<b>5</b>
5.1	Look and Feel Requirements . . . . .	5
5.2	Usability and Humanity Requirements . . . . .	6
5.3	Performance Requirements . . . . .	7
5.4	Operational and Environmental Requirements . . . . .	7
5.5	Maintainability and Support Requirements . . . . .	7
5.6	Security Requirements . . . . .	7
5.7	Cultural Requirements . . . . .	8
5.8	Legal Requirements . . . . .	9
<b>6</b>	<b>Unit Testing</b>	<b>10</b>
<b>7</b>	<b>Changes Due to Testing</b>	<b>11</b>
<b>8</b>	<b>Automated Testing</b>	<b>11</b>
<b>9</b>	<b>Trace to Requirements</b>	<b>11</b>
<b>10</b>	<b>Trace to Modules</b>	<b>11</b>
<b>11</b>	<b>Code Coverage Metrics</b>	<b>11</b>
<b>12</b>	<b>Appendix</b>	<b>12</b>
12.1	Reflection . . . . .	12

## List of Tables

1	Revision History . . . . .	i
2	Unit Tests Results Summary . . . . .	10

## List of Figures

## 3 Introduction

### 3.1 Project Summary

Sayyara is a Progressive Web Application (PWA) which acts as a single platform for independent auto repair shops and vehicle owners. This platform allows independent auto repair shops and vehicle owners to interact in various ways. Using Sayyara, vehicle owners can search for auto repair shops and services based on a variety of search filters; request quotes for service; book, view, and manage service appointments. On the application, auto repair shop owners have full shop management capabilities such as: adding and managing a list of employees; managing a list of service types and corresponding service appointment availabilities; managing store information such as location, hours of operation, and contact information. Auto repair shop owners and employees will be able to manage quotes, service appointments, and work orders from a single application.

### 3.2 Purpose of Document

This document outlines the validation and verification process for the application, Sayyara, and a detailed summary of results for the planned tests defined in the [VnV Plan](#).

### 3.3 Scope of Testing

As defined in the [VnV Plan](#), the tests reported aim to verify and validate functional and nonfunctional requirements listed in the [SRS](#).

In addition, the testing reported in this document aims to validate that the application provides adequate usability and to verify that system is in a functional state for the end users. The testing and validation will aid in ensuring that the product fulfills the system requirements, intended use, and goals of stakeholders.

## 4 Functional Requirements Evaluation

### 4.0.1 Authentication

This section will contains tests covering the “Authentication” requirements defined in the [SRS](#).

#### 1. FRT-BE1-1

Control: Manual

Initial State: Customer with no existing account

Input: Customer submits sign up form with valid email, password, name, and phone number

Output: Redirect to vehicle owner landing page upon successful registration

Test Case Derivation: New customer wants to sign up for a new account with valid sign up information provided

How test will be performed: The tester will register and sign up for a new customer account using valid sign up information

Results: Passed

**2. FRT-BE1-2**

Control: Manual

Initial State: Shop owner with no existing account

Input: Shop owner submits sign up form with valid email, password, name, phone number, shop name, shop address, shop phone number, and shop email

Output: Redirect to shop owner landing page upon successful registration

Test Case Derivation: New shop owner user wants to sign up for a new account with valid sign up information provided

How test will be performed: The tester will register and sign up for a new shop owner account using valid sign up information

Results: Passed

**3. FRT-BE1-3**

Control: Manual

Initial State: Employee with no existing account

Input: Employee submits sign up form with valid email, password, name, phone number, and shop ID

Output: Redirect to employee landing page upon successful registration

Test Case Derivation: New employee user wants to sign up for a new account with valid sign up information provided

How test will be performed: The tester will register and sign up for a new employee account using valid sign up information

Results: Passed

**4. FRT-BE2-1**

Control: Manual

Initial State: Customer with existing account and not logged in

Input: Customer submits login form with valid email, password, and vehicle information

Output: Redirect to vehicle owner landing page upon successful login

Test Case Derivation: Customer wants to login to existing with valid login credentials provided

How test will be performed: The tester will login to an existing customer account using valid login credentials

Results: Passed

5. **FRT-BE2-2**

Control: Manual

Initial State: Shop owner with existing account and not logged in

Input: Shop owner submits login form with valid email and password

Output: Redirect to shop owner landing page upon successful login

Test Case Derivation: Shop owner wants to login to existing with valid login credentials provided

How test will be performed: The tester will login to an existing shop owner account using valid login credentials

Results: Passed

6. **FRT-BE2-3**

Control: Manual

Initial State: Employee with existing account and not logged in

Input: Employee submits login form with valid email and password

Output: Redirect to employee landing page upon successful login

Test Case Derivation: Employee wants to login to existing with valid login credentials provided

How test will be performed: The tester will login to an existing employee account using valid login credentials

Results: Passed

7. **FRT-BE3-1**

Control: Manual

Initial State: Customer with existing account and logged in

Input: Customer requests to logout

Output: System logs the user out and transitions to home page

Test Case Derivation: Customer wants to logout

How test will be performed: The tester will request to logout on a customer account

Results: Passed

8. **FRT-BE3-2** Control: Manual

Initial State: Shop owner with existing account and logged in

Input: Shop owner requests to logout

Output: System logs the user out and transitions to home page

Test Case Derivation: Shop owner wants to logout

How test will be performed: The tester will request to logout on a shop owner account

Results: Passed

#### 9. **FRT-BE3-3**

Control: Manual

Initial State: Employee with existing account and logged in

Input: Employee requests to logout

Output: System logs the user out and transitions to home page

Test Case Derivation: Employee wants to logout

How test will be performed: The tester will request to logout on an employee account

Results: Passed

### 4.1 **Shop Profile**

This section will contain tests covering the “Shop Profile” requirements defined in the [SRS](#).

#### 1. **FRT-BE33-1**

Control: Manual

Initial State: Shop Owner/Employee Account with a registered shop

Input: Request to view shop profile

Output: Information regarding the shop, which includes the shop’s address, phone number, and email

Test Case Derivation: The system shall display information about the shop upon request

How test will be performed: The tester will go through the user interface to view a shop profile

Results: Passed

#### 2. **FRT-BE34-1**

Control: Manual

Initial State: User wants to view a shop’s profile

Input: Request to view shop profile

Output: Information regarding the shop, which includes the shop’s address, phone number, email, and hours of operation (if any)

Test Case Derivation: The system shall display information about the shop upon request



How test will be performed: The tester will go through the user interface to view a shop profile

Results: Passed

### 3. **FRT-BE35-1**

Control: Manual

Initial State: Shop Owner wants to add/update hours of operation of their shop

Input: Request to add and edit hours of operation with valid hours of operation

Output: Updated hours of operation

Test Case Derivation: The system shall allow the shop owner to add/edit their hours of operation

How test will be performed: The tester will go through the user interface to add/edit a shop

Results: Passed

## 5 Nonfunctional Requirements Evaluation

### 5.1 Look and Feel Requirements

#### 1. **NFRT-LF1-1**

Type: Dynamic, Manual

Initial State: The application is accessible through the Google Chrome web browser

Input/Condition: The window size is changed

Output/Result: The application shall adjust and scale to fit the new window size

How test will be performed: The tester will access the application through Google Chrome on their desktop/laptop and change the windows through the use of Google Chrome DevTools Device Toolbar

Results: Passed

#### 2. **NFRT-LF2-1**

Type: Dynamic, Manual

Initial State: The application is accessible through the web browser

Input: The application is opened in a full screened web browser window

Output: All text on the screen shall be readable

How test will be performed: The tester open the application in a full screened web browser window, navigate to all pages, and judge if all text on the screen is readable from sitting 50 centimeters away from the monitor

Results: Passed

### 3. **NFRT-LF3-1**

Type: Dynamic, Manual

Initial State: The application is accessible through the web browser and there is a completed work order and quote on the user's account

Input: The user opens the work order details and the quote details

Output: All currency shall be rounded to two decimal places

How test will be performed: The tester will navigate to the work order and quote and verify that all values of currency are rounded to two decimal places

Results: Passed

## 5.2 Usability and Humanity Requirements

### 1. **NFRT-UH1-1**

Type: Dynamic, Manual

How test will be performed: The testers will complete the manual system tests for functional requirements on a MacOS desktop/laptop device, a Windows desktop/laptop, an iOS mobile device, and an android mobile device.

Results: Passed. Testers were asked to write down which device they used to test to ensure all devices and operating systems were covered.

### 2. **NFRT-UH2-1**

Type: Dynamic, Manual

Initial State: Device is connected to the internet and application is not open.

Input/Condition: The user launches the application on a web browser.

Output/Result: The system can be assessed through the web browser.

How test will be performed: The testers will attempt to launch the application on a web browser, using a device that is connected to the internet.

Results: Passed. The testers were able to access the application using the web browser while connected to the internet.

### 3. **NFRT-UH3-1**

Type: Dynamic, Manual

Initial State: Device is connected to the internet and application is open.

Input/Condition: User disconnects from the internet.

Output/Result: The system notifies the user that there is no network connection.

How test will be performed: The testers disconnects from the internet while the application is open.

Results: Passed. Testers were greeted with a toast when they disconnected from the internet.

## 5.3 Performance Requirements

### 1. NFRT-PR1-1

Type: Dynamic, Manual

How test will be performed: A function will be added to the application which logs web metrics of the system. Testers will go through each of the pages and functions, making sure that the metrics meet current web standards as stated [in this article](#). These criteria include:

- LCP < 2.5s
- FID < 100ms
- CLS < 0.1
- TTFB < 0.5s
- FCP < 1.8s

Results: Passed. Testers were able to view logs of their progress and record results which met or exceeded current web standards for the metrics.

## 5.4 Operational and Environmental Requirements

### 1. NFRT-OE-1

Type: Dynamic, Manual

Initial State: The application is accessible through the Google Chrome web browser

Input: The user navigates across all pages and dialogs, and performs all manual system tests for functional requirements

Output: All pages and dialogs should be fully operational on the below listed device dimensions

How test will be performed: The testers will complete the manual system tests for functional requirements on Google Chrome using various device dimension options listed in the Google DevTools Device Toolbar. At minimum, the testers will test with the following dimensions: Responsive, iPhone 12 Pro (390 × 844), iPhone SE (375 × 667), and iPad Air (820 × 1180). This set will provide a reasonable variety of different device dimensions.

Results: Failed. Manually tested to verify whether the application supports the above device dimensions. Shop lookup page does not work on iPad Air (820 × 1180) dimensions. Chat window does not work on iPhone SE (375 × 667) dimensions.

## 5.5 Maintainability and Support Requirements

## 5.6 Security Requirements

### 1. NFRT-SR1-1

Type: Dynamic, Automatic

Initial State: An account with a created appointment

Input/Condition: An HTTP request to the application's backend to get the appointment by appointment ID

Output/Result: The request is rejected with a 403 Forbidden error message

How test will be performed: Through Jest an HTTP request will be sent to the server

Results: Passed.

## 2. **NFRT-SR2-1**

Type: Dynamic, Manual

Initial State: One Shop Owner account and two Customer accounts with one customer having an appointment at the shop registered under the same Shop Owner account

Input/Condition: Request available time slots at the Shop Owner's store using the Customer without an appointment's account

Output/Result: The Customer should only be able to view the time slot taken by the other customer and no other extra information

How test will be performed: The tester will view the available time slots page as the Customer without an appointment

Results: Failed. Extra information is returned from the API call that is not needed for this endpoints and exposes data that does not belong to user. None of the data contains PII.

## 5.7 Cultural Requirements

### 1. **NFRT-CR1-1**

Type: Static, Manual

Initial State: The application is accessible through the web browser

Input: The user navigates across all pages and dialogs, and performs all end-to-end tests

Output: There should be no offensive texts or images

How test will be performed: The testers will go through end-to-end tests for the entire program and will confirm that any texts or images are not offensive.

Results: Passed. Manually tested to verify that there are no offensive images or text.

### 2. **NFRT-CR2-1** Type: Static, Manual

Initial State: The application is accessible through the web browser

Input: The user navigates across all pages and dialogs, and performs all end-to-end tests

Output: All text should be displayed in English

How test will be performed: The testers will go through end-to-end tests for the entire program and will confirm that the text displayed is in English.

Results: Passed. Checked that everything is written in English in the implementation, documentation, and anything visible to the user.

## 5.8 Legal Requirements

### 1. NFRT-LR1-1

Type: Static, Manual

Input: The user navigates all source code and project documents

Output: Appropriate credits and/or copyright licenses are used, and MIT License requirements should be adhered to

How test will be performed: The testers will go through the source code of the project and project documents to determine if any open source resources were used and check that appropriate credits and/or copyright licenses are used.

Results: Passed. Checked the MIT License requirements and that the project adhered to all the requirements.

## 6 Unit Testing

Table 2: Unit Tests Results Summary

$\checkmark$ = Pass, X = Fail					
Test ID	Result	Test ID	Result	Test ID	Result
FRT-M3-1	$\checkmark$	FRT-M3-2	$\checkmark$	FRT-M3-3	$\checkmark$
FRT-M3-4	$\checkmark$	FRT-M3-5	$\checkmark$	FRT-M3-6	$\checkmark$
FRT-M3-7	$\checkmark$	FRT-M3-8	$\checkmark$	FRT-M3-9	$\checkmark$
FRT-M3-10	$\checkmark$	FRT-M4-1	$\checkmark$	FRT-M4-2	$\checkmark$
FRT-M4-3	$\checkmark$	FRT-M4-4	$\checkmark$	FRT-M4-5	$\checkmark$
FRT-M3-6	$\checkmark$	FRT-M4-7	$\checkmark$	FRT-M4-8	$\checkmark$
FRT-M4-9	$\checkmark$	FRT-M4-10	$\checkmark$	FRT-M4-11	$\checkmark$
FRT-M4-12	$\checkmark$	FRT-M4-13	$\checkmark$	FRT-M4-14	$\checkmark$
FRT-M4-15	$\checkmark$	FRT-M4-16	$\checkmark$	FRT-M4-17	$\checkmark$
FRT-M5-1	$\checkmark$	FRT-M5-2	$\checkmark$	FRT-M5-3	$\checkmark$
FRT-M5-4	$\checkmark$	FRT-M5-5	$\checkmark$	FRT-M5-6	$\checkmark$
FRT-M5-7	$\checkmark$	FRT-M5-8	$\checkmark$	FRT-M5-9	$\checkmark$
FRT-M5-10	$\checkmark$	FRT-M5-11	$\checkmark$	FRT-M5-12	$\checkmark$
FRT-M5-13	$\checkmark$	FRT-M5-14	$\checkmark$	FRT-M6-1	$\checkmark$
FRT-M6-2	$\checkmark$	FRT-M6-3	$\checkmark$	FRT-M6-4	$\checkmark$
FRT-M6-5	$\checkmark$	FRT-M6-6	$\checkmark$	FRT-M6-7	$\checkmark$
FRT-M7-1	$\checkmark$	FRT-M7-2	$\checkmark$	FRT-M7-3	$\checkmark$
FRT-M7-4	$\checkmark$	FRT-M7-5	$\checkmark$	FRT-M7-6	$\checkmark$
FRT-M8-1	$\checkmark$	FRT-M8-2	$\checkmark$	FRT-M8-3	$\checkmark$
FRT-M8-4	$\checkmark$	FRT-M8-5	$\checkmark$	FRT-M8-6	$\checkmark$
FRT-M8-7	$\checkmark$	FRT-M8-8	$\checkmark$	FRT-M8-9	$\checkmark$
FRT-M8-10	$\checkmark$	FRT-M8-11	$\checkmark$	FRT-M8-12	$\checkmark$
FRT-M8-13	$\checkmark$	FRT-M8-14	$\checkmark$	FRT-M8-15	$\checkmark$
FRT-M9-1	?				

## 7 Changes Due to Testing

[This section should highlight how feedback from the users and from the supervisor (when one exists) shaped the final product. In particular the feedback from the Rev 0 demo to the supervisor (or to potential users) should be highlighted. —SS]

## 8 Automated Testing

All automated testing was carried out by Jest, a JavaScript testing framework. Each module has its own corresponding test file located within the “test” folder. The “test” folder has the same structure as the “src” folder, creating a one-to-one mapping of a module and its corresponding test. For example, the module “src/server/services/userService.ts” has its tests located in “test/server/services/userService.test.ts”. There are two Jest test runs, with the only difference being whether the testing environment is connected to a real database or not (i.e., whether the database needs to be seeded or mocked in each test case). As part of the CI/CD pipeline, the full test suite is run and information regarding pass/fail and code coverage is reported on every pull request to the main branch and to every push to the main branch.

## 9 Trace to Requirements

## 10 Trace to Modules

## 11 Code Coverage Metrics

## 12 Appendix

### 12.1 Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Reflection. Please answer the following question:

1. In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)