

Project Title: System Verification and Validation Plan for Sayyara

Team 3, Tiny Coders

Arkin Modi

Joy Xiao

Leon So

Timothy Choy

October 31, 2022

1 Revision History

Table 1: Revision History

| Date | Developer(s) | Change |
|------------------|--------------|--|
| October 26, 2022 | Arkin Modi | Added Software Validation Plan |
| October 27, 2022 | Joy Xiao | Add Testing Team and Design Verification |
| October 27, 2022 | Arkin Modi | Add Automated Testing and Verification Tools |
| October 28, 2022 | Leon So | Add Summary |
| October 28, 2022 | Arkin Modi | Add System Tests for Quotes |
| October 29, 2022 | Leon So | Add Objectives |
| October 29, 2022 | Arkin Modi | Add System Tests for Work Orders |
| October 30, 2022 | Arkin Modi | Add Implementation Verification Plan |
| October 30, 2022 | Leon So | Add Usability and Humanity Tests |

Contents

| | | |
|----------|--|-----------|
| 1 | Revision History | i |
| 2 | Symbols, Abbreviations and Acronyms | iv |
| 3 | General Information | 1 |
| 3.1 | Summary | 1 |
| 3.2 | Objectives | 1 |
| 3.3 | Relevant Documentation | 1 |
| 4 | Plan | 1 |
| 4.1 | Verification and Validation Team | 1 |
| 4.2 | SRS Verification Plan | 2 |
| 4.3 | Design Verification Plan | 2 |
| 4.4 | Verification and Validation Plan Verification Plan | 2 |
| 4.5 | Implementation Verification Plan | 2 |
| 4.6 | Automated Testing and Verification Tools | 3 |
| 4.7 | Software Validation Plan | 3 |
| 5 | System Test Description | 3 |
| 5.1 | Tests for Functional Requirements | 3 |
| 5.1.1 | Quotes | 3 |
| 5.1.2 | Work Orders | 8 |
| 5.2 | Tests for Nonfunctional Requirements | 10 |
| 5.2.1 | Area of Testing1 | 10 |
| 5.2.2 | Area of Testing2 | 11 |
| 5.3 | Traceability Between Test Cases and Requirements | 11 |
| 6 | Unit Test Description | 11 |
| 6.1 | Unit Testing Scope | 11 |
| 6.2 | Tests for Functional Requirements | 11 |
| 6.2.1 | Module 1 | 11 |
| 6.2.2 | Module 2 | 12 |
| 6.3 | Tests for Nonfunctional Requirements | 12 |
| 6.3.1 | Module ? | 12 |
| 6.3.2 | Usability and Humanity Requirements | 13 |
| 6.4 | Traceability Between Test Cases and Modules | 13 |
| 7 | Appendix | 14 |
| 7.1 | Symbolic Parameters | 14 |
| 7.2 | Usability Survey Questions? | 14 |

List of Tables

| | | |
|---|--|---|
| 1 | Revision History | i |
| 2 | Manual System Testing Responsibilities | 2 |

[Remove this section if it isn't needed —SS]

List of Figures

[Remove this section if it isn't needed —SS]

2 Symbols, Abbreviations and Acronyms

| symbol | description |
|--------|-------------------------------------|
| CI | Continuous Integration |
| SRS | Software Requirements Specification |
| PWA | Progressive Web Application |

This document ... [provide an introductory blurb and roadmap of the Verification and Validation plan —SS]

3 General Information

3.1 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 Objectives

The objective of the testing outlined in this document is to demonstrate adequate usability and ensure that the system is in a functional state for the end users. The testing and validation will also aid in ensuring that the product fulfills the system requirements, intended use, and goals of stakeholders.

3.3 Relevant Documentation

[Reference relevant documentation. This will definitely include your SRS and your other project documents (design documents, like MG, MIS, etc). You can include these even before they are written, since by the time the project is done, they will be written. —SS]

Author (2019)

4 Plan

[Introduce this section. You can provide a roadmap of the sections to come. —SS]

4.1 Verification and Validation Team

The verification and validation team will consist of the core developers (Joy Xiao, Tim Choy, Leon So, Arkin Modi), as well as the course instructor and TAs.

The developers are responsible for coming up with tests with suitable edge cases to evaluate the correctness of Sayyara. The developers will all be responsible for writing and executing all test cases listed in the document and taking note of the results. The developers will ensure that Sayyara passes all tests after performing the tests and making any necessary updates.

4.2 SRS Verification Plan

[List any approaches you intend to use for SRS verification. This may include ad hoc feedback from reviewers, like your classmates, or you may plan for something more rigorous/systematic. —SS]

[Maybe create an SRS checklist? —SS]

4.3 Design Verification Plan

Design verification will be done by the core developers of the project. The design will also be reviewed by the TAs of the course. The design of the system will be verified by going through the requirements from the [Software Requirements Specification \(SRS\)](#) and determining whether the outputs correspond with the expected inputs. The verification will also be done by going through the [Module Guide \(MG\)](#) and [Module Interface Specification \(MIS\)](#) checklist and ensure that all the modules are completed and fulfill the corresponding requirements.

4.4 Verification and Validation Plan Verification Plan

[The verification and validation plan is an artifact that should also be verified. —SS]

[The review will include reviews by your classmates —SS]

[Create a checklists? —SS]

4.5 Implementation Verification Plan

Automated integration tests will be created to fulfill the system tests defined in section 5 and automated unit tests will be created to fulfill the tests defined in section 6. Static analysis will be performed through code review by at least one core team member per pull request and automation tools as stated in 4.6. A pull request will cover at most one module.

Manual System Tests defined in section 5 shall take place from January 16, 2023 to January 30, 2023. The areas of responsibility will be divided as described in table 2.

Table 2: Manual System Testing Responsibilities

| Area of Testing | Team Member Responsible |
|---------------------|-------------------------|
| Authentication | Leon So |
| Appointments | Joy Xiao |
| Quotes | Arkin Modi |
| Work Orders | Arkin Modi |
| Employee Management | Leon So |
| Services | Joy Xiao |
| Shop Lookup | Timothy Choy |
| Shop Profile | Timothy Choy |
| Employee Profile | Timothy Choy |

4.6 Automated Testing and Verification Tools

All tooling and automations that will be used for testing and verification have been documented in the [Development Plan](#) under the “Technology” and “Code Standard” sections.

As part of our CI, we will have a pipeline running the automated tests against every commit to the main branch and every pull request opened against the main branch that contains a source code or testing code change. There will also be a pipeline running against every pull request asserting adherence to the documented code style standards. Code coverage reports will be commented directly to the pull request or the commit to the main branch using a pipeline.

4.7 Software Validation Plan

The plan for validating the software and the requirements shall be to conduct review session with the stakeholders. These review sessions shall focus on the business events and user flows as defined in the [SRS](#).

5 System Test Description

5.1 Tests for Functional Requirements

[Subsets of the tests may be in related, so this section is divided into different areas. If there are no identifiable subsets for the tests, this level of document structure can be removed. —SS]

[Include a blurb here to explain why the subsections below cover the requirements. References to the SRS would be good here. —SS]

5.1.1 Quotes

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

1. FRT-BE8-1

Control: Manual

Initial State: Customer Account with no quote requests initiated from the account

Input: Request past quotes

Output: A message saying there are no past quotes

Test Case Derivation: There are no past quotes to display

How test will be performed: The tester will request for past quotes through the user interface

2. FRT-BE8-2

Control: Manual

Initial State: Customer Account with quotes requests which have been closed and were initiated by the account

Input: Request past quotes

Output: A table with all the quotes

Test Case Derivation: There are past quotes to display

How test will be performed: The tester will request for past quotes through the user interface

3. **FRT-BE8-3**

Control: Manual

Initial State: Shop Owner/Employee Account with no quotes requests assigned to the shop

Input: Request past quotes with quote ID, customer phone number, and/or customer name

Output: A message saying there are no past quotes

Test Case Derivation: There are no past quotes to display matching the inputted criteria

How test will be performed: The tester will request for past quotes through the user interface

4. **FRT-BE8-4**

Control: Manual

Initial State: Shop Owner/Employee Account with closed quotes requests that were assigned to the shop

Input: Request past quotes with quote ID, customer phone number, and/or customer name

Output: A table with all the quotes

Test Case Derivation: There are past quotes to display matching the inputted criteria

How test will be performed: The tester will request for past quotes through the user interface

5. **FRT-BE9-1**

Control: Manual

Initial State: Customer Account with quotes requests which have been closed and were initiated by the account

Input: Request quote details

Output: The quote details

Test Case Derivation: Display the quote details requested

How test will be performed: The tester will request for quote details through the user interface

6. FRT-BE9-2

Control: Manual

Initial State: Shop Owner/Employee Account with closed quotes requests that were assigned to the shop

Input: Request quote details

Output: The quote details

Test Case Derivation: Display the quote details requested

How test will be performed: The tester will request for quote details through the user interface

7. FRT-BE10-1

Control: Manual

Initial State: Customer Account

Input: A filled in quote request form

Output: A confirmation messaging indicating that the quote was requested

Test Case Derivation: The system shall verify that the quote was created successfully

How test will be performed: The tester will create a quote through the user interface

8. FRT-BE10-2

Control: Manual

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

Input: A quote is requested from the shop

Output: A notification for the open quote request

Test Case Derivation: The system shall notify that a quote was created

How test will be performed: The tester will create a quote through the user interface

9. FRT-BE11

Control: Manual

Initial State: Customer Account with a created quote

Input: Request to delete quote with the quote's ID

Output: A quote deleted confirmation message

Test Case Derivation: The system shall delete the quote upon request

How test will be performed: The tester will delete a quote through the user interface

10. FRT-BE12-1

Control: Manual

Initial State: Customer Account with a created quote

Input: Request to update quote with the quote's new details

Output: A quote updated confirmation message

Test Case Derivation: The system shall update the quote upon request

How test will be performed: The tester will update a quote through the user interface

11. FRT-BE12-2

Control: Manual

Initial State: Shop Owner/Employee Account with a registered shop and an active quote

Input: A quote's details are updated

Output: A notification for the updated quote

Test Case Derivation: The system shall notify that a quote was updated

How test will be performed: The tester will update a quote through the user interface

12. FRT-BE13

Control: Manual

Initial State: Customer Account with a created quote

Input: A request to create a new quote using details from another quote which is identified by quote ID

Output: A confirmation messaging indicating that the quote was requested

Test Case Derivation: The system shall verify that the quote was created successfully with an existing quote's details

How test will be performed: The tester will create a quote through the user interface

13. FRT-BE14-1

Control: Manual

Initial State: Customer Account with a created quote

Input: A response to a quote

Output: A notification for the responded to quote

Test Case Derivation: The system shall notify that a quote was responded to

How test will be performed: The tester will create a response to a quote through the user interface

14. FRT-BE14-2

Control: Manual

Initial State: Shop Owner/Employee Account with a registered shop and an active quote

Input: A response to a quote

Output: A quote responded confirmation message

Test Case Derivation: The system shall verify that the quote was responded to successfully

How test will be performed: The tester will create a response to a quote through the user interface

15. FRT-BE15

Control: Manual

Initial State: Customer Account with a responded to quote

Input: Accept a responded to quote

Output: The system shall request the user to create an appointment with the shop that responded

Test Case Derivation: The system shall have the user create an appointment upon accepting a quote

How test will be performed: The tester will accept a quote through the user interface

16. FRT-BE16-1

Control: Manual

Initial State: Customer Account with a created quote

Input: A message to request additional information

Output: A message sent confirmation

Test Case Derivation: The system shall confirm to the user that the message was sent

How test will be performed: The tester will send a message for additional information through the user interface

17. FRT-BE16-2

Control: Manual

Initial State: Shop Owner/Employee Account with a registered shop and an active quote

Input: A message to request additional information

Output: A message sent confirmation

Test Case Derivation: The system shall confirm to the user that the message was sent

How test will be performed: The tester will send a message for additional information through the user interface

5.1.2 Work Orders

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

1. **FRT-BE17-1**

Control: Manual

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

Input: A new appointment created at the shop from the quote

Output: A new work order is created with details from the quote

Test Case Derivation: The system shall automatically create a work order for new appointment with information from the quote

How test will be performed: The tester will create an appointment through the user interface

2. **FRT-BE17-2**

Control: Manual

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

Input: A new appointment created at the shop

Output: A new work order is created

Test Case Derivation: The system shall automatically create a work order for new appointment

How test will be performed: The tester will create an appointment through the user interface

3. **FRT-BE18**

Control: Manual

Initial State: Shop Owner/Employee Account with a registered shop, an upcoming appointment, and a work order associated to the appointment

Input: Appointment is deleted

Output: Appointment and Work Order deleted confirmation message

Test Case Derivation: The system shall delete appointment and the associated work order upon request

How test will be performed: The tester will delete an appointment through the user interface

4. **FRT-BE19**

Control: Manual

Initial State: Shop Owner/Employee Account with a registered shop, an upcoming appointment, and a work order associated to the appointment

Input: Work Order is searched by the customer name, assigned employee, service type, and/or a date range

Output: The work orders matching the inputted criteria in a table format

Test Case Derivation: The system shall display work order upon request

How test will be performed: The tester will search for a work order through the user interface

5. **FRT-BE20**

Control: Manual

Initial State: Shop Owner/Employee Account with a registered shop, an upcoming appointment, and a work order associated to the appointment

Input: Request to view past work orders

Output: Past work orders in a table format

Test Case Derivation: The system shall display work order upon request

How test will be performed: The tester will search for past work orders through the user interface

6. **FRT-BE21**

Control: Manual

Initial State: Shop Owner/Employee Account with a registered shop, an upcoming appointment, and a work order associated to the appointment

Input: Request to update a work order with new details

Output: A work order updated confirmation message

Test Case Derivation: The system shall update work order upon request

How test will be performed: The tester will update a work order through the user interface

7. **FRT-BE22**

Control: Manual

Initial State: Shop Owner/Employee Account with a registered shop, an appointment, and a work order associated to the appointment

Input: A customer has paid for their appointment

Output: The work order is emailed to the customer, and the appointment and work order are marked as “completed”

Test Case Derivation: The system shall mark work orders and appointments as completed when the customer has paid for the word done

How test will be performed: The tester will mark a work order as completed through the user interface

8. **FRT-BE23**

Control: Manual

Initial State: Shop Owner/Employee Account with a registered shop, an appointment, and a work order associated to the appointment

Input: Request work order details

Output: The work order details

Test Case Derivation: The system shall display work orders details upon request

How test will be performed: The tester will view a work order's details through the user interface

5.2 Tests for Nonfunctional Requirements

[The nonfunctional requirements for accuracy will likely just reference the appropriate functional tests from above. The test cases should mention reporting the relative error for these tests. Not all projects will necessarily have nonfunctional requirements related to accuracy —SS]

[Tests related to usability could include conducting a usability test and survey. The survey will be in the Appendix. —SS]

[Static tests, review, inspections, and walkthroughs, will not follow the format for the tests given below. —SS]

5.2.1 Area of Testing1

Title for Test

1. test-id1

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input/Condition:

Output/Result:

How test will be performed:

2. test-id2

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

5.2.2 Area of Testing2

...

5.3 Traceability Between Test Cases and Requirements

[Provide a table that shows which test cases are supporting which requirements. —SS]

6 Unit Test Description

[Reference your MIS (detailed design document) and explain your overall philosophy for test case selection. —SS] [This section should not be filled in until after the MIS (detailed design document) has been completed. —SS]

6.1 Unit Testing Scope

[What modules are outside of the scope. If there are modules that are developed by someone else, then you would say here if you aren't planning on verifying them. There may also be modules that are part of your software, but have a lower priority for verification than others. If this is the case, explain your rationale for the ranking of module importance. —SS]

6.2 Tests for Functional Requirements

[Most of the verification will be through automated unit testing. If appropriate specific modules can be verified by a non-testing based technique. That can also be documented in this section. —SS]

6.2.1 Module 1

[Include a blurb here to explain why the subsections below cover the module. References to the MIS would be good. You will want tests from a black box perspective and from a white box perspective. Explain to the reader how the tests were selected. —SS]

1. test-id1

Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

Initial State:

Input:

Output: [The expected result for the given inputs —SS]

Test Case Derivation: [Justify the expected value given in the Output field —SS]

How test will be performed:

2. test-id2

Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

Initial State:

Input:

Output: [The expected result for the given inputs —SS]

Test Case Derivation: [Justify the expected value given in the Output field —SS]

How test will be performed:

3. ...

6.2.2 Module 2

...

6.3 Tests for Nonfunctional Requirements

[If there is a module that needs to be independently assessed for performance, those test cases can go here. In some projects, planning for nonfunctional tests of units will not be that relevant. —SS]

[These tests may involve collecting performance data from previously mentioned functional tests. —SS]

6.3.1 Module ?

1. test-id1

Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

Initial State:

Input/Condition:

Output/Result:

How test will be performed:

2. test-id2

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

6.3.2 Usability and Humanity Requirements

1. NFRT-UH1

Type: Dynamic, Manual

How test will be performed: The testers will complete the manual system tests for functional requirements listed section 5.1 on a MacOS desktop/laptop device, a Windows desktop/laptop, an iOS mobile device, and an android mobile device.

2. NFRT-UH2

Type: Dynamic, Manual

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

Input/Condition: The user launches the application on Google Chrome

Output/Result: The system can be assessed through Google Chrome

How test will be performed: The tester will attempt to launch the application on the Google Chrome web browser and using a device that is connected to the internet.

3. NFRT-UH3

Type: Dynamic, Manual

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

Input/Condition: User disconnects from the internet

Output/Result: The system notifies the user that there is no network connection and functionality will be limited.

How test will be performed: The tester disconnect from the internet while the application is open.

6.4 Traceability Between Test Cases and Modules

[Provide evidence that all of the modules have been considered. —SS]

References

Author Author. System requirements specification. <https://github.com/...>, 2019.

7 Appendix

This is where you can place additional information.

7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

7.2 Usability Survey Questions?

[This is a section that would be appropriate for some projects. —SS]

Appendix — Reflection

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

1. What knowledge and skills will the team collectively need to acquire to successfully complete the verification and validation of your project? Examples of possible knowledge and skills include dynamic testing knowledge, static testing knowledge, specific tool usage etc. You should look to identify at least one item for each team member.
2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?