

Software Requirements Specification for Sayyara Automotive Matcher: Sayyara Automotive Matches

Team 27, Kappastone
Tevis Doe, doet
Caitlin Bridel, bridelc
Gilbert Cherrie, cherrieg
Rachel Johnson, johnsr12
Harkeerat Kanwal, kanwalh
Himanshu Aggarwal, aggarwah

October 6, 2022

Contents

1	Reference Material	v
1.1	Table of Units	v
1.2	Table of Symbols	v
1.3	Abbreviations and Acronyms	v
1.4	Mathematical Notation	v
2	Introduction	1
2.1	Purpose of Document	1
2.2	Characteristics of Intended Reader	1
2.3	Organization of Document	1
3	General System Description	2
3.1	System Context	2
3.2	User Characteristics	2
3.3	System Constraints	3
4	Specific System Description	3
4.1	Problem Description	3
4.1.1	Terminology and Definitions	3
4.2	Solution Characteristics Specification	4
4.2.1	Assumptions	4
5	Functional Requirements	4
5.1	The Scope of the Work and the Product	4
5.1.1	The Context of the Work	4
5.1.2	Work Partitioning	4
5.1.3	Individual Product Use Cases	5
5.2	Functional Requirements	9
5.3	Formalized Requirements	11
5.4	Phase in Plan	12
6	Nonfunctional Requirements	13
6.1	Look and Feel Requirements	13
6.1.1	Appearance Requirements	13
6.1.2	Style Requirements	13
6.2	Usability and Humanity Requirements	13
6.2.1	Ease of Use Requirements	13
6.2.2	Personalization and Internationalization Requirements	14
6.2.3	Learning Requirements	14
6.2.4	Understandability and Politeness Requirements	14
6.3	Performance Requirements	14
6.3.1	Speed and Latency Requirements	14

6.3.2	Safety-Critical Requirements	15
6.3.3	Precision or Accuracy Requirements	15
6.3.4	Reliability and Availability Requirements	15
6.3.5	Robustness or Fault-Tolerance Requirements	15
6.3.6	Capacity Requirements	15
6.3.7	Scalability or Extensibility Requirements	16
6.3.8	Longevity Requirements	16
6.4	Operational and Environmental Requirements	16
6.4.1	Expected Physical Environment	16
6.4.2	Requirements for Interfacing with Adjacent Systems	16
6.4.3	Productization Requirements	16
6.4.4	Release Requirements	16
6.5	Maintainability and Support Requirements	17
6.5.1	Maintenance Requirements	17
6.5.2	Supportability Requirements	17
6.5.3	Adaptability Requirements	17
6.6	Security Requirements	17
6.6.1	Access Requirements	17
6.6.2	Integrity Requirements	17
6.6.3	Privacy Requirements	17
6.6.4	Audit Requirements	18
6.6.5	Immunity Requirements	18
6.7	Cultural Requirements	18
6.7.1	Cultural Requirements	18
6.7.2	Political Requirements	18
6.8	Legal Requirements	18
6.8.1	Compliance Requirements	18
6.8.2	Standards Requirements	18
6.9	Health and Safety Requirements	19
7	Project Issues	19
7.1	Open Issues	19
7.2	Off-the-Shelf Solutions	19
7.3	Tasks	19
7.4	Migration to the New Product	20
7.5	Risks	20
7.6	Costs	20
7.7	User Documentation and Training	20
8	Appendix	21
8.1	Reflection	21
8.1.1	Approaches	22
8.1.2	Chosen Strategies	22

Revision History

Date	Version	Notes
5 October 2022	1.0	First iteration of SRS

This SRS closely follows the Volere template provided by the course instructor, however it uses some parts of the initial template as well. However, it covers the same sections as both templates, so no big changes have actually occurred. Some of the subsections that were not relevant to the project were omitted from these templates.

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

N/A

1.2 Table of Symbols

N/A

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
FR	Functional Requirement
NFR	Non-Functional Requirement
SRS	Software Requirements Specification
PWA	Progressive Web App
MVP	Minimal Viable Product
AWS	Amazon Web Services
URL	Uniform Resource Locator

1.4 Mathematical Notation

N/A

2 Introduction

This section provides an overview of the rest of the document. It describes the purpose of this document, outlines the expected knowledge of the reader and provides a roadmap of the SRS.

2.1 Purpose of Document

The purpose of this document is to present a detailed description of Sayyara Automotive Matcher and outline the expectations of the program for functionality and performance. The document will describe the purpose of the system and its features and constraints. This SRS will also indicate conditions under which the system will be developed. This document is intended to be read by stakeholders and developers to achieve a better understanding of the system.

2.2 Characteristics of Intended Reader

The intended audience for this document are developers and other stakeholders such as project managers, quality assurance and product experts. It is assumed that these readers have a basic understanding of Progressive Web App (PWA) development and server hosting. It is also assumed that the reader understands the process for car maintenance, including quotes, work orders and billing.

2.3 Organization of Document

The document is divided into sections that each give an in-depth look into an aspect of the system. To begin, the [General System Description](#) section describes the system and identifies how it will interact with its environment and users. This section will also summarize the attributes of the expected users. The following section, [Specific System Description](#), will describe the purpose of the system through the problem statement and explain necessary terminology for complete understanding of the problem and requirements. The [Functional Requirements](#) section will outline the needs of the client and expectations of the software, in terms of functionality. This section will also describe the scope of work, formal requirements and the plan for how the team intends to implement the functional requirements. Following that, the [Nonfunctional Requirements](#) section will explain all quality features the team plans to implement in this project. The current environment for this system will be discussed in the [Project Issues](#) section, to present all factors that will contribute to the success or potential failure of Sayyara Automotive Matcher. Finally, the [Reflection Appendix](#) details the skills and knowledge each team member requires in order to complete this project and the proposed action plan accepted by each member.

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

The system context for this PWA is shown in Figure 1. This figure shows the relationship between the user inputs and the different parts of the software that the users interact with. The figure starts at the User box which shows the first stage of user input which is the user logging in as a customer, shop owner or shop employee. From this stage they are guided to the respective front end view of the PWA relevant to their access level and from these front end PWA views they have various interactions and user inputs that will interact with the database and have various outputs as shown by the arrows in the figure.

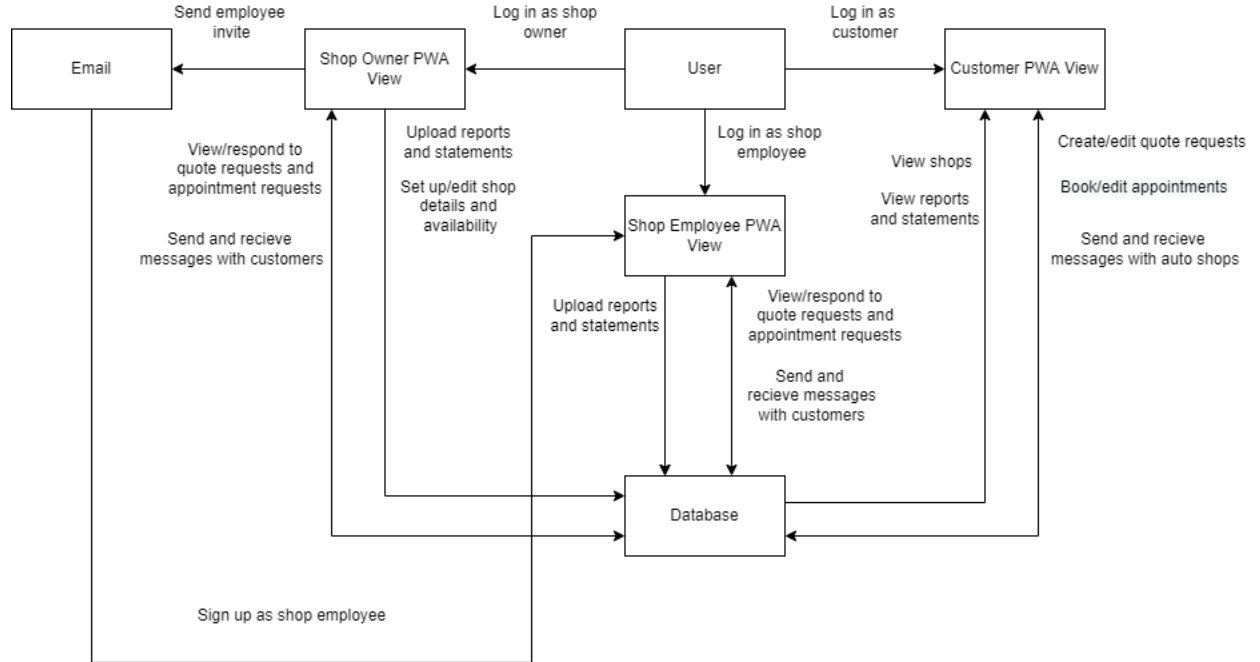


Figure 1: System context displaying user interactions with PWA views and database.

3.2 User Characteristics

The end users of the system can be of two types, a typical consumer and a typical shop owner or employee. The general consumer will not be expected to have any knowledge about auto motives, and as such it will not be required to use the software. In general, the

typical consumer is expected to know how to use a mobile app or web page (which will be the output of this project, see [System Constraints](#)) without any instruction. Finally, general consumers are assumed to have no physical disabilities that may prevent them from using the system.

Shop owners/employees are expected to have all the same general knowledge as a typical consumer but with knowledge relating to the automotive industry in addition. In particular, this type of user will have knowledge about the terminology and processes used in the automotive industry, and as such, they will not be explained to any potential users. Shop owners or employees are expected to have the knowledge to be able to assist consumers in their understanding of automotive issues to allow them to more effectively use the application.

3.3 System Constraints

This system has no constraints, barring one. The output of the project, as mandated by the supervisor of the project, will be a PWA (progressive web app). This means that the output of the project will take the form both of a web page that can be run in a browser, but also an application that will run on mobile devices.

4 Specific System Description

This section presents the problem description, which gives a high-level view of the problem to be solved. In addition, it contains a section detailing any terminology used in the document that may be unclear for readers.

4.1 Problem Description

Sayyara Automotive Matcher is intended to solve a typical issue that a consumer may run in to when attempting to service their vehicle: finding a reliable mechanic, and creating appointments with them. Consumers may have to phone many mechanics to find a mechanic that charges a reasonable price for services, and may not necessarily even be able to find a time slot that works for both the mechanic and them. Sayyara Automotive Matcher intends to solve these issues with coordination for both consumers and mechanics, making servicing vehicles easier for both of them.

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Canned job: a job provided by a shop where the pricing and timing is already listed, so there is no need for a customer to request a quote or for the shop to send a quote to the customer.

4.2 Solution Characteristics Specification

This section covers the assumptions that may affect the requirements if changed in the future.

4.2.1 Assumptions

Many assumptions about the users of the system are detailed under [User Characteristics](#), but the following assumptions are also made:

- The mobile hardware that a user has will support PWAs.
- When using the application, the user will have a valid internet connection.
- The servers that eventually host the system will be able to support the users of the system.

If there is a change in these assumptions, the functional requirements defined in the following section may be subject to change.

5 Functional Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

5.1 The Scope of the Work and the Product

5.1.1 The Context of the Work

The system context is covered in [Section 3.1](#)

5.1.2 Work Partitioning

The work partitioning is outlined in [Table 1](#).

Table 1: Work Partitioning

Event	Input and Output	Summary
Create account	User info (IN)	User creates account
User logs in	Username (IN) Password (IN)	User logs into the system
Reset password	New password (IN)	User resets password

Enroll shop	Shop name (IN) Shop address (IN) Owner info (IN)	Shop owner enrolls store
Owner invites employee	Employee (IN) Employee invite (OUT)	Owner invites employee to create account
View all users	All users (OUT)	System admin views all mechanics and users
Send quote request	Quote request (IN) Quote from shop (OUT)	Customer sends quote request to shop
Book appointment	Appointment request (IN) Appointment acceptance/rejection (OUT)	Customer books appointment with shop
Create work order	Customer and quote info (IN) Work order (OUT)	Shop creates work order
Assign work order	Employee (IN) Work order (IN)	Shop owner assigns work order to employee
Cancel appointment	Appointment (IN)	Shop owner cancels appointment
Set shop availability	Shop available hours/days (IN) Shop schedule (OUT)	Shop sets shop availability

5.1.3 Individual Product Use Cases

UC1. Name: Enroll Shop

Trigger: Shop owner attempts to enroll shop

Precondition: None

Main Success Scenario:

1. Shop owner enters required information

Success Postcondition: Shop owner successfully enrolls shop

Functional Requirement(s): FR1

UC2. Name: Create Owner Account

Trigger: Shop owner attempts to create account

Precondition: Shop owner enrolls shop

Main Success Scenario:

1. Shop owner enters required information

Success Postcondition: Shop owner successfully creates account
Functional Requirement(s): FR2

UC3. Name: Create Employee Account

Trigger: Shop employee attempts to create account

Precondition: Shop owner invites employee to create account

Main Success Scenario:

1. Shop employee accepts invitation from shop owner
2. Shop employee enters required information

Success Postcondition: Shop employee successfully creates account
Functional Requirement(s): FR3, FR4, FR13

UC4. Name: Login

Trigger: User attempts to login

Precondition: User creates account

Main Success Scenario:

1. User enters username and password
2. System authenticates username and password

Undesired Event Handling:

1. System cannot authenticate username or password
2. User repeats Step 2 until successful authentication

Success Postcondition: User successfully logs into account
Functional Requirement(s): FR8

UC5. Name: View All Users

Trigger: System admin attempts to view all mechanics and users

Precondition: System admin authenticates admin status

Success Postcondition: System admin successfully views all mechanics and users

Functional Requirements(s): FR5

UC6. Name: Edit Account

Trigger: User attempts to edit account

Precondition: User logs into account

Main Success Scenario:

1. User enters changes to account
2. User saves changes

Success Postcondition: User changes to account become visible
Functional Requirement(s): FR6

UC7. Name: Delete Account

Trigger: User attempts to delete account

Precondition: User is not a shop owner

Success Postcondition: User account is no longer visible

Functional Requirement(s): FR7

UC8. Name: Reset Password

Trigger: User attempts to reset password

Precondition: User creates account

Main Success Scenario:

1. User enters contact information
2. System authenticates that user info exists in records
3. System allows user to reset password

Undesired Event Handling:

1. System unable to authenticate user
2. User repeats Step 1 until authentication succeeds

Success Postcondition: User successfully resets password
Functional Requirement(s): FR9

UC9. Name: Customer Books Appointment

Trigger: Customer books appointment with shop

Precondition: None

Main Success Scenario:

1. Customer searches for and selects shop
2. Customer sends quote request to shop
3. Shop owner or employee creates quote
4. Shop owner or employee sends quote to customer
5. Customer accepts quote
6. Shop owner or employee accepts appointment
7. System automatically creates work order

Alternate Scenario:

1. Customer searches for and selects shop
2. Customer books a canned job (see [4.1.1](#))
3. Shop owner or employee accepts appointment
4. System automatically creates work order

Undesired Event Handling:

1. Customer rejects quote or shop rejects appointment
2. No appointment created

Success Postcondition: Customer successfully books appointment

Functional Requirement(s): [FR10](#), [FR11](#), [FR18](#), [FR20](#), [FR26](#), [FR28](#), [FR29](#), [FR30](#)

UC10. Name: Shop Cancels Appointment

Trigger: Shop owner attempts to cancel appointment

Precondition: Customer or shop books appointment

Main Success Scenario:

1. Shop owner searches for upcoming work order, quote, or appointment
2. Shop owner views appointment
3. Shop owner cancels appointment
4. System automatically deletes work order

Success Postcondition: Shop successfully cancels appointment

Functional Requirement(s): [FR12](#), [FR22](#), [FR23](#), [FR25](#)

UC11. Name: Assign Work Order

Trigger: Shop owner attempts to assign work order to employee

Precondition: Work order has been created

Main Success Scenario:

1. Shop owner searches for and selects employee

Success Postcondition: Shop owner successfully assigns work order to employee

Functional Requirement(s): [FR14](#), [FR27](#)

UC12. Name: Send Work Order

Trigger: Shop attempts to send work order to customer

Precondition: Shop owner or employee creates work order

Success Postcondition: Customer successfully receives work order

Functional Requirement(s): [FR15](#), [FR16](#)

- UC13. Name: Edit Shop Profile
Trigger: Shop owner attempts to edit shop profile
Precondition: Shop owner enrolls shop and creates account
Success Postcondition: Changes visible on shop profile
Functional Requirement(s): FR17
- UC14. Name: Set Shop Availability
Trigger: Shop owner or employee attempts to set shop availability
Precondition: Shop owner or employee creates account
Success Postcondition: Customers can view shop availability
Functional Requirement(s): FR19

5.2 Functional Requirements

- FR1. The system will allow shop owners to enroll their shops.
Use Case: UC1
- FR2. The system will allow shop owners to create individual accounts.
Use Case: UC2
- FR3. The system will allow shop owners to invite employees to create individual accounts.
Use Case: UC3
- FR4. The system will allow shop employees to create individual accounts via employer invitation.
Use Case: UC3
- FR5. The system will allow system admin to see all mechanics and users.
Use Case: UC5
- FR6. The system will allow any user to edit their own account.
Use Case: UC6
- FR7. The system will allow shop employees to delete their own account.
Use Case: UC7
- FR8. The system will allow shop owners and employees to login by entering a username and password.
Use Case: UC4
- FR9. The system will allow shop owners and employees to reset their password.
Use Case: UC8
- FR10. The system will allow shop owners to create quotes.
Use Case: UC9

- FR11. The system will allow shop owners and employees to send quotes to customers.
Use Case: UC9
- FR12. The system will allow shop owners to accept, reject, and assign appointments.
Use Case: UC10
- FR13. The system will allow shop owners to add, remove, and view all employees.
Use Case: UC3
- FR14. The system will allow shop owners and employees to view, search for, and filter employees.
Use Case: UC11
- FR15. The system will allow shop owners and employees to create work orders.
Use Case: UC12
- FR16. The system will allow shop owners to send work orders to customers.
Use Case: UC12
- FR17. The system will allow shop owners to edit their shop profile.
Use Case: UC13
- FR18. The system will allow shop employees to accept and view appointments.
Use Case: UC9
- FR19. The system will allow shop employees to set shop availability.
Use Case: UC14
- FR20. The system will allow any user to view shop profiles.
Use Case: UC9
- FR21. The system will allow customers to submit rework requests.
- FR22. The system will allow shop owners and employees to view, search for, and filter quotes.
Use Case: UC10
- FR23. The system will allow shop owners and employees to view, search for, and filter past and upcoming work orders.
Use Case: UC10
- FR24. The system will allow shop owners and employees to call and chat with customers.
- FR25. The system will automatically delete a work order if the appointment is cancelled.
Use Case: UC10
- FR26. The system will automatically create a work order if an appointment is created.
Use Case: UC9

FR27. The system will allow shop owners and employees to assign work orders to employees.
Use Case: UC11

FR28. The system will allow customers to book appointments.
Use Case: UC9

FR29. The system will allow customers to submit quote requests.
Use Case: UC9

FR30. The system will allow customers to search for and filter shops.
Use Case: UC9

Note: These functional requirements were derived from a Confluence board [1] with all user features created by our supervisor.

5.3 Formalized Requirements

The following Parnas table, Table 2, will formalize the utilization of the system between customers and shop owners/employees, one of the most important parts of the system. It will use the following variables to represent interactions, with abbreviations as necessary:

- Stimulus:
 - In_Chat: The customer or shop opens a chat with each other.
 - In_CustomerR: The customer makes an appointment with a shop.
 - In_ShopR: The shop owner/employee accepts a customer request.
 - In_Complete: The appointment with a customer is marked as complete.
- State:
 - Init (I): Initial state, can be considered to be a landing page.
 - ChatWin (CW): The customer can chat with the shop and vice versa.
 - AppShop (AS): The shop can see an appointment request from the customer.
 - AppAcc (AA): The customer can see their request has been accepted by the shop.

Table 2: Formalized subsection of the requirements space.

	I	CW	AS	AA	CW & AS	CW & AA
In.Chat	CW	-	CW & AS	CW & AA	-	-
In.CustomerR	AS	CW & AS	-	-	-	-
In.ShopR	-	-	AA	-	CW & AA	-
In.Complete	-	-	-	I	-	I

5.4 Phase in Plan

In this section, the functional requirements have been divided into batches of work based on the applications user stories. The division of batches and proposed dates are detailed in Table 3.

Table 3: Schedule of batches of work

Batch	Description	Functional Requirements	Dates
1	Shop Owner Account Creation and Shop Enrolment	FR1, FR2, FR8, FR9	Oct. 24, 2022
2	Basic Customer Services	FR10, FR11, FR12, FR15, FR16, FR18, FR19, FR20, FR22, FR23, FR24, FR25, FR26, FR28, FR29, FR30	Nov. 21, 2022
3	Employee Account Creation	FR3, FR4	Jan. 5, 2023
4	Shop Owner Capabilities	FR13, FR14, FR27	Feb. 7, 2023
5	Finishing Touches	FR5, FR6, FR7, FR17, FR21	Mar. 20, 2023

6 Nonfunctional Requirements

6.1 Look and Feel Requirements

6.1.1 Appearance Requirements

NFR-1 The product shall be visually appealing.

Rationale: To attract new shop owners and customers to sign up, it is important that the product appears visually appealing.

Fit Criterion: Without any external prompt, 90% of a sample group of representative users shall start using the product within a few minutes of their encounter.

NFR-2 The product shall have a consistent appearance on all of the devices.

Rationale: It is important for the product to have a consistent appearance so that users can use their experience with the product on other devices.

Fit Criterion: 95% of a sample group of users shall agree that the product looks identical in appearance on different devices.

NFR-3 The product shall allow the users to customize the appearance of the product.

Rationale: Since everyone has a difference taste, it is important for the product to offer customization options so that users can have a more personalized experience.

Fit Criterion: The supervisor shall certify that the product is customizable.

6.1.2 Style Requirements

NFR-4 The product shall feel professional and trustworthy.

Rationale: It is essential that the product appear professional and trustworthy in order to build up the trust of the users.

Fit Criterion: 85% of a representative sample of users shall agree that the product feels professional and trustworthy,

NFR-5 The font and theme used shall be consistent throughout the app.

Rationale: It is important that the components adhere to the same fonts and styles to ensure that they complement one another.

Fit Criterion: 90% of a representative sample of users shall not be able to discern any variations in font and theme of any of the elements of the product.

6.2 Usability and Humanity Requirements

6.2.1 Ease of Use Requirements

NFR-6 The product shall be easy to use by anyone above the age of 16.

Rationale: Since the minimum driving age in most provinces is 16, the product

shall be user friendly for everyone above this age.

Fit Criterion: 80% of the representative users who are 16 or older shall be able to do the assigned tasks in the allotted time.

NFR-7 The product shall be usable by people with a basic comprehension of the English language, without any prior training.

Rationale: To target a diverse group of users from various educational backgrounds, the product must not require prior training or knowledge to operate.

Fit Criterion: Without any training, 90% of a sample English-speaking group shall be able to navigate through the different services offered by the app.

6.2.2 Personalization and Internationalization Requirements

NFR-8 The product shall automatically save user preferences.

Rationale: Users should be able to exit the web app and return later or on a different device with the same settings applied. This saves them time from making modifications everytime they launch the web app.

Fit Criterion: The settings applied by a user shall automatically apply on other devices at least 90% of the times.

6.2.3 Learning Requirements

NFR-9 The product shall require no external training to learn.

Rationale: Users should not have to spend time learning how to use the different services offered by the product.

Fit Criterion: 95% of a sample group shall be able to use the different services offered by the product without any prior experience or training with the product.

6.2.4 Understandability and Politeness Requirements

NFR-10 The product shall use icons and symbols that are widely recognized.

Rationale: Widely recognizable icons and symbols will make the product more understandable.

Fit Criterion: In a wide and diverse sample group, 90% of the users shall be able to comprehend what the icons and symbols represent.

6.3 Performance Requirements

6.3.1 Speed and Latency Requirements

NFR-11 Messages sent by users shall be received by the recipients within 2 seconds after being sent.

Rationale: Messages should be sent and received in a timely manner to prevent any waiting time for the users.

Fit Criterion: The product shall be able to deliver messages to the recipients within 2 seconds for 90 percent of the times, and less than 5 seconds for the rest of the times.

NFR-12 The product's loading time between different pages shall be no more than 5 seconds.

Rationale: A website with long loading times can lose the trust of its users.

Fit Criterion: When navigating between different pages, the product shall take less than 2 seconds 95% of times, and less than 4 seconds rest of the times.

NFR-13 The product's response times shall be quick enough to prevent the disruption of the user's train of thought.

Rationale: Users will not be able to use the product's services efficiently if the response time is slow.

Fit Criterion: A survey of a group of users shall indicate that 90% of the users found the response times to be fast.

6.3.2 Safety-Critical Requirements

N/A

6.3.3 Precision or Accuracy Requirements

N/A

6.3.4 Reliability and Availability Requirements

NFR-14 The product shall achieve 98% up time.

Rationale: A product with long downtime will lose the trust of its users.

Fit Criterion: The product shall not be unavailable for more than 2% of the times.

6.3.5 Robustness or Fault-Tolerance Requirements

NFR-15 Errors shall be logged and handled gracefully.

Rationale: Logging errors will allow developers to find potential bugs in the product. Handling errors gracefully will allow users to continue to use the app even in case of errors.

Fit Criterion: A survey of users conducted after a few weeks of launching the product shall indicate that no more than 5% of users encountered any crashes.

6.3.6 Capacity Requirements

N/A

6.3.7 Scalability or Extensibility Requirements

NFR-16 The product shall be able to scale to handle more than 50,000 users within the 6 months after its release.

Rationale: A product that is designed with scalability in mind will require less changes in the future.

Fit Criterion: Logging reports shall not show an increase in failure rate when more than 50,000 users have signed up on the product.

6.3.8 Longevity Requirements

NFR-17 The product shall be expected to operate for at least 5 years without requiring significant maintenance.

Rationale: A product that is designed to operate with low maintainence will last longer.

Fit Criterion: Internal surveys shall show that the majority of engineers working on the product believe it can operate without significant maintainence.

6.4 Operational and Environmental Requirements

6.4.1 Expected Physical Environment

N/A

6.4.2 Requirements for Interfacing with Adjacent Systems

NFR-18 The frontend application shall be able to communicate with the backend server.

Rationale: If the product is designed to allow interfacing with a backend server, developers can prevent future rework.

Fit Criterion: The frontend application shall be capable of fetching and sending data to the backend server.

6.4.3 Productization Requirements

NFR-19 The product shall be hosted on a remote server and be accessible to customers through a URL.

Rationale: It is important for the product to be deployed on a server so that users can access it quickly and on the device of their choice.

Fit Criterion: A diverse group of users from around the globe shall verify that they can access the product using the URL.

6.4.4 Release Requirements

N/A

6.5 Maintainability and Support Requirements

6.5.1 Maintenance Requirements

NFR-20 New updates shall be installed overnight.

Rationale: Updating the product overnight will avoid downtimes during business hours.

Fit Criterion: Log reports shall verify that updates are only occurring during the night.

6.5.2 Supportability Requirements

N/A

6.5.3 Adaptability Requirements

NFR-21 The product shall be installable on mobile and desktop devices, and run on modern web browsers.

Rationale: An app that can operate on multiple platforms has a greater market reach.

Fit Criterion: A survey of the users shall indicate that the app is compatible on at least 80% of their modern devices.

6.6 Security Requirements

6.6.1 Access Requirements

NFR-22 Only the developers and other members of the company shall be able to access the internal bug and crash logs.

Rationale: It is a security risk if a log report with sensitive data is made accessible to all users.

Fit Criterion: A security firm shall certify that only the company employees and developers have access to the internal bug and crash logs.

6.6.2 Integrity Requirements

NFR-23 Users shall not be able to directly query data stored in the database.

Rationale: It is a huge security and privacy risk if users are able to query the database directly.

Fit Criterion: A security firm shall certify that users have no way to query data stored in the database.

6.6.3 Privacy Requirements

NFR-24 Private data shall be protected in compliance with applicable data protection laws and the company's information policy.

Rationale: A product that complies with data protection laws is more trusted by the users.

Fit Criterion: A security firm shall certify that all data protection laws are being followed.

NFR-25 Users shall be notified before storing cookies on their devices.

Rationale: To comply with the privacy laws, it is necessary to alert users before storing cookies on their computer.

Fit Criterion: A survey shall indicate that at least 99% of the users received a prompt to allow cookies to be stored on their devices when they first accessed the product.

6.6.4 Audit Requirements

N/A

6.6.5 Immunity Requirements

N/A

6.7 Cultural Requirements

6.7.1 Cultural Requirements

NFR-26 The product shall not include any images or content that might be perceived as offensive to the users.

Rationale: Customers would not want to use a product that contains any offensive imagery or content.

Fit Criterion: There shall be a consensus among a diverse group of potential customers that the product doesn't include anything that may be offensive.

6.7.2 Political Requirements

N/A

6.8 Legal Requirements

6.8.1 Compliance Requirements

N/A

6.8.2 Standards Requirements

N/A

6.9 Health and Safety Requirements

N/A

7 Project Issues

7.1 Open Issues

In the current automotive ecosystem, it is a hassle for the average motorist to find an auto service professional they can trust and that provides the most affordable services. The process requires many phone calls to be made and the consumer is required to describe the same problem multiple times.

From the opposite perspective, it is difficult for shop owners to accurately provide estimates on when they would be available to service your vehicle.

7.2 Off-the-Shelf Solutions

There are many automotive repair related apps already on the market, but even the best of them are only able to fulfill a subset of the aforementioned problems. One such app, GoMechanic Car Services, provides price estimates, appointment booking, and an accessory shop, but it is limited to a single automotive repair shop franchise, and does not allow independent repair shop owners to leverage the platform. Furthermore, it is only available in India, and such an alternative does not yet exist in North America.

Another app, Simply Auto: Car Maintenance, focuses more on providing an accurate maintenance schedule and providing cost summaries and other cost related features, such as attaching receipts, and calculating taxes. This app fails, however, to provide easy appointment booking, and other helpful features that allow the consumer to stay connected with repair technicians.

7.3 Tasks

1. Complete documentation as required.
2. Setup repository, initialize project codebases, and setup continuous integration pipeline.
3. Receive the product requirements from our industry partner.
4. Create the relational database model.
5. Create the UI design on Figma.
6. Implement features.
7. Demo features to partner and interview our partner for clarification on the requirements.

8. Repeat previous two steps until the final product is created.

7.4 Migration to the New Product

The app will be designed to be easy to learn for both users that are completely new to these types of apps, as well as for users that have already used a similar solution in the past.

7.5 Risks

The back-end server could go down, but this has a low likelihood of occurring if we host it on a popular cloud provider such as AWS or Google Cloud. In general, this project is not safety critical, so there are no other major risks.

7.6 Costs

This project has no costs for development other than the cost incurred by the students to take this course.

7.7 User Documentation and Training

Documentation is not required for this project as our industry partner only desires a MVP to showcase to prospective investors. There is no training required to use this system.

8 Appendix

8.1 Reflection

The successful completion of this software project is dependent on the team's collective acquisition of necessary skills. The skills and/or knowledge that each team member will need are listed in Table A1.

Table A1: Breakdown of team member required knowledge

Team Member	Required Knowledge
Tevis Doe	<ul style="list-style-type: none">• Front-end specific knowledge: NextJS, PWA technologies• Back-end specific knowledge: Django• General knowledge: General long-term project management
Caitlin Bridel	<ul style="list-style-type: none">• Knowledge: Next.js, Django, and PWA technologies• Skills: Time management
Gilbert Cherrie	<ul style="list-style-type: none">• Front-end specific knowledge: Learn NextJS and about PWA technologies, enhance React skills• Back-end specific knowledge: Learn Django and creating restful APIs
Rachel Johnson	<ul style="list-style-type: none">• Front-end specific knowledge: React, NextJS• Back-end specific knowledge: Django
Continued on next page	

Table A1 – continued from previous page

Team Member	Required Knowledge
Harkeerat Kanwal	<ul style="list-style-type: none">• Front-end specific knowledge: React, NextJS• Back-end specific knowledge: Django, PostgreSQL• Project management using GitHub.
Himanshu Aggarwal	<ul style="list-style-type: none">• Front-end specific knowledge: React, NextJS, TailwindCSS• Back-end specific knowledge: Django, PostgreSQL

8.1.1 Approaches

A possible approach to gain knowledge and enhance skills in front-end and back-end technologies including React, NextJS, TailwindCSS, Django, Postgresql and PWA technologies is by taking online courses through Udemy or Coursera. These courses are usually paid but will give an in-depth tutorial and overview about the specific front end technologies. Another approach is to use free sources such as Youtube and written tutorials on websites such as geeksforgeeks or w3schools. This option will not cost the team but will require the team members to create their own syllabus and do more self-teaching.

To learn project management skills, team members could study project management methodologies through online documentation. This could also include Github projects and issue documentation, to learn how Github will be used to manage project work. Another strategy would be to meet with a project manager to shadow his work or learn from a 1-on-1 on how to implement project management skills in this project effectively.

Enhancing time management skills could be done by adopting a tight schedule for this project and using calendars, agendas, and to-do lists to track progress. Another way to develop this skill is to set SMART goals and work with another team member to check-in on individual progress to ensure goals are being met on a regular basis.

8.1.2 Chosen Strategies

For learning domain-specific technologies, all team members have decided to use free outlets such as Youtube and tutorial websites to source relevant information for the technologies they require more knowledge in. This approach was chosen to keep costs for each team member low and also the team was able to find helpful free resources that will allow them to gain necessary knowledge.

To enhance project management skills, Tevis and Harkeerat will review methodologies and documentation online. Harkeerat will specifically focus on Github documentation. This method was chosen by both team members as it will be much less time consuming than reaching out to a project manager and both members only require small refreshers to contribute to the project effectively.

Caitlin has chosen to improve her time management skills by adding our project schedule to her calendar and making sure to stay on track by consistently tracking progress with to-do and priority lists. Caitlin has chosen this strategy as she already keeps a calendar for other courses and can easily apply it to this project on a more rigid scale to increase her time management.

9 Resources

- [1] Ibrahim, Nabil. "Sayyara Confluence." *Confluence*, <https://www.atlassian.com/>. We cannot link to the confluence directly, as it contains information we cannot disclose, however, it was used to create this document.
- [2] James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.