

System Design for Sayyara

Team 3, Tiny Coders

Arkin Modi

Joy Xiao

Leon So

Timothy Choy

January 16, 2023

1 Revision History

Table 1: Revision History

Date	Developer(s)	Change
December 28, 2022	Arkin Modi	Revision History & Mark Not Applicable Sections
January 7, 2023	Joy Xiao	Introduction & Purpose
January 11, 2023	Leon So	Undesired Event Handling
January 12, 2023	Leon So	Normal Behaviour & Introduction
January 16, 2023	Joy Xiao	Component Diagram

2 Reference Material

This section records information for easy reference.

2.1 Abbreviations and Acronyms

symbol	description
Sayyara	Explanation of program name
MIS	Module Interface Specifications
MG	Module Guide

Contents

1	Revision History	i
2	Reference Material	ii
2.1	Abbreviations and Acronyms	ii
3	Introduction	1
4	Purpose	1
5	Scope	1
6	Project Overview	1
6.1	Normal Behaviour	1
6.2	Undesired Event Handling	1
6.3	Component Diagram	2
6.4	Connection Between Requirements and Design	2
7	System Variables	2
7.1	Monitored Variables	2
7.2	Controlled Variables	2
7.3	Constants Variables	3
8	User Interfaces	3
9	Design of Hardware	3
10	Design of Electrical Components	3
11	Design of Communication Protocols	3
12	Timeline	3
13	Appendix	4
13.1	Interface	4
13.2	Reflection	4

List of Tables

1	Revision History	i
---	----------------------------	---

List of Figures

1	Component Diagram	2
---	-----------------------------	---

3 Introduction

The following document details the System Design for project Sayyara. Sayyara is a progressive web application (PWA) which will act as a single platform for independent auto repair shops and vehicle owners. This platform will allow independent auto repair shops and vehicle owners to interact in a more efficient and effective manner.

Complementary documents include the Module Interface Specifications and Module Guide. The full documentation and implementation can be found at <https://github.com/arkinmodi/project-sayyara/>.

4 Purpose

The purpose of this document is to display the component decomposition of the system and provide the user interface designs of the software being built. The implementation of the software will be based off of the designs within this document. The MIS <https://github.com/arkinmodi/project-sayyara/blob/main/docs/Design/SoftDetailedDes/MIS.pdf> and MG <https://github.com/arkinmodi/project-sayyara/blob/main/docs/Design/SoftArchitecture/MG.pdf> are also created to give details to the software architecture and detailed component breakdowns for the project.

5 Scope

[Include a figure that show the System Context (showing the boundary between your system and the environment around it.) —SS]

6 Project Overview

6.1 Normal Behaviour

Sayyara is an event-driven application which handles inputs from the intended users including: vehicle owners, and independent auto repair shop owners and employees. The application will accept various inputs through a variety of input forms and controls. Under normal behaviour where valid inputs are entered and valid events are triggered, the application will: update the appropriate local and global application states, trigger the corresponding side-effects, and/or update the database accordingly.

Vehicle owners can search for auto repair shops and services; request quotes for service; book, view, and manage service appointments and work orders. On the application, auto repair shop owners will be able to manage a list of employees; manage a list of service types and corresponding service appointment availabilities; manage store information such as location, hours of operation, and contact information. Auto repair shop owners and employees will be able to view and manage quotes, service appointments, and work orders.

6.2 Undesired Event Handling

Undesired events will be handled both client-side and server-side.

On the client-side, if an unexpected event arises or the application enters a bad state, the application will reset to a safe state. For example, if a user attempts to access a route that they are not authorized to access, they will be either redirected to an appropriate route, prompted to login, or an error page will be displayed with instructions to return to the home page. Input forms will also include input validation to ensure only properly formed data is handled. If the user attempts to input invalid data, the form field will reset and form submission will be blocked. The user will be prompted to enter a valid input value in the form field. Similarly, various user actions and inputs that may pose cause that the application to enter an undesirable state will be validated before updating the application state.

On the server-side, each API will return a response with the appropriate error status code and message. Subsequently, the client will have logic to gracefully handle unsuccessful responses and status codes, preventing the system from entering an undesirable state. Inputs will also be validated on the server-side by parsing the input data using defined schemas. This will ensure data integrity and prevents the undesirable data from entering the workflows or database.

6.3 Component Diagram

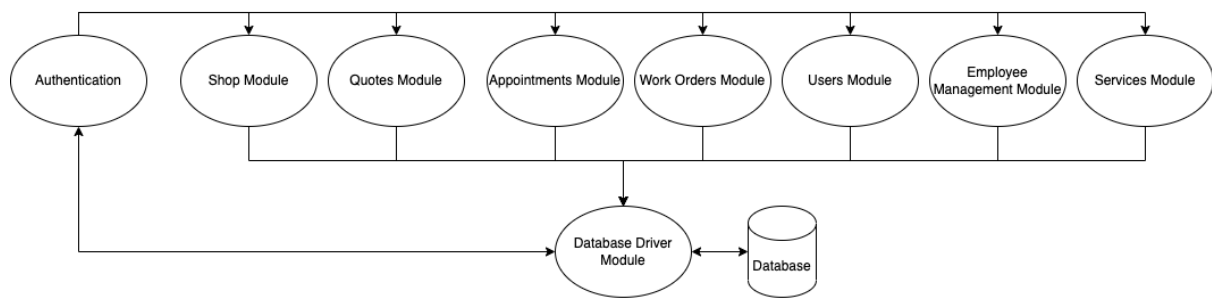


Figure 1: Component Diagram

6.4 Connection Between Requirements and Design

[The intention of this section is to document decisions that are made “between” the requirements and the design. To satisfy some requirements, design decisions need to be made. Rather than make these decisions implicit, they are explicitly recorded here. For instance, if a program has security requirements, a specific design decision may be made to satisfy those requirements with a password. —SS]

7 System Variables

7.1 Monitored Variables

N/A

7.2 Controlled Variables

N/A

7.3 Constants Variables

N/A

8 User Interfaces

[Design of user interface for software and hardware. Attach an appendix if needed.
Drawings, Sketches, Figma —SS]

9 Design of Hardware

N/A

10 Design of Electrical Components

N/A

11 Design of Communication Protocols

N/A

12 Timeline

[Schedule of tasks and who is responsible —SS]

13 Appendix

13.1 Interface

[Include additional information related to the appearance of, and interaction with, the user interface —SS]

13.2 Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Problem Analysis and Design. Please answer the following questions:

1. What are the limitations of your solution? Put another way, given unlimited resources, what could you do to make the project better? (LO_ProbSolutions)
2. Give a brief overview of other design solutions you considered. What are the benefits and tradeoffs of those other designs compared with the chosen design? From all the potential options, why did you select documented design? (LO_Explores)