Hazard Analysis
Sayyara

Team 31
SFWRENG 4G06
Christopher Andrade
Alyssa Tunney
Kai Zhu
Ethan Vince-Budan
Collin Kan
Harsh Gupta

April 5, 2023

#	Date	Developer(s)	Change
	Oct. 10, 2022	Kai	Added intro, H1-1 and SR1-3
	Oct. 11, 2022	Alyssa	Added scope/purpose, H2-1 and SR4-5
	Oct. 12, 2022	Chris	Added system boundaries and components section, H3-1 to H4-1, and SR-6 to SR-9
	Oct. 13, 2022	Ethan	Added H5-1, SR-10 to SR-12
	Oct. 14, 2022	Ethan	Added H5-2, SR-13
	Oct. 15, 2022	Collin	Added H6-1, SR-14 to SR-16 and section 7 Roadmap
Rev 0	Oct. 19, 2022	All	Revision 0
Rev 1	April 5, 2022	All	Revision 1

Table 1: Revision History

Contents

List of Tables

1 Introduction

This document contains the hazard analysis for the Sayyara Automotive Service Progressive Web Application. The application is an all-in-one platform to connect customers with auto service providers for shop search, quote requests, work orders, and scheduling. Hazards in this application are vulnerabilities or potential sources of errors that can cause system failures or security risks.

2 Scope and Purpose of Hazard Analysis

The failures in the Sayyara system can result in consequences. Some of these consequences may be hazardous. The purpose of the hazard analysis is to identify the potential hazards, the failures that can cause these hazards, as well as the causes of these failures. The scope of this hazard analysis document covers potential hazards identified in Sayyara components, including what the cause of these hazards may be and how they can be resolved or mitigated with the help of safety and security requirements. System boundaries and critical assumptions are also taken into consideration for the scope of potential hazards. Hazard analysis will be continued throughout the development stages of Sayyara to ensure all potential hazards that may arise are handled appropriately.

3 System Boundaries and Components

The system of the application is made up of many working parts. These can essentially be broken down into three sections: the front-end, the back-end, and the environment for which they operate in. The three sections will contain components from low-to-high abstractions levels without over-generalizing or over-decomposing.

These are the components that make up the front-end (for the user interface, input verification, etc):

- NextJS framework for cross-platform user interface and building frontend
- Cypress testing library
- Vercel webpage host

These are the components that make up the back-end (for authentication, server requests, storage of data, etc):

- PostgreSQL relational database structure
- Django server routing
- Microsoft Azure for database and server host
- Redis for real-time chat and notification system

- unittest testing library
- Docker for building backend and frontend

These are the components that make up the environment:

- Device hardware (computer)
- Operating system
- Web browser
- GitHub code storage

In terms of interactable high-level components that will utilize all the above components, there are: ("management" here implies creation, searching, editing, and deletion):

- Authentication
- Appointment scheduling
- Chat system
- Vehicle owner shop lookup
- Quotes management
- Shop work orders management
- Shop employees management
- Shop services management
- Profile management
- Payment service

As such, the aforementioned hardware, software, browser, hosting services, building and deploy containers, user interface, testing systems, along with the high-level interactable components, will make up the system's boundary.

4 Critical Assumptions

- The user has access to the internet.
- The deployment stack supports Docker and hosts at 99.99% up time.
- The shop owners take the responsibility of the correctness and ownership of the data.
- Any server-side denial of service attacks will be mitigated by the hosting provider.

5 Failure Mode and Effect Analysis

Design Function	Failure Modes	Effects of Failure	Causes of Failure	Recommended Action	SR	Ref.
Authenticate users	Grants access to unauthorized users	Exposes user/shop data and configurations	a. Malicious brute force password attempts b. User choosing common password c. Collision in password hash	a. Limit password attempts per time period b. Enforce more secure password practices on sign up c. Use collision-resistant hashing together with salting	a. SR-1 b. SR-2 c. SR-3	H1-1
Appointment scheduling	A scheduled appointment between customer and shop is overwritten with second appointment	a. Causes first customer to have to reschedule to potentially later date b. First customer might completely cancel appointment, causing shop to lose business and money	a. Failure to update database with modified web app information	a. Ensure database updates as soon as first appointment is scheduled b. Immediately remove scheduled appointment from pool of available appointments	a. SR-4 b. SR-5	H2-1
Real-time Chat and Notifications System	Timeout on a live chat	Shop owners and vehicle owners cannot communicate through live chat	a. Redis software failure b. Server/database host service is down from spamming	a. Limit number of messages sent in a time-frame b. Notify the user that the message cannot be sent due to a server outage	a. SR-6 b. SR-7	H3-1
Shop lookup	Not all shops appear to the vehicle owner	Vehicle owner cannot find a shop they are looking for even though it does exist	a. Query too large for database	a. Break large queries into smaller queries through pagination b. Use loading spinners to notify the user if a query is still occurring despite some results being received	a. SR-8 b. SR-9	H4-1
Quotes management	Quote request contains incorrect information	Confusion for ASP; potential over/under paying for service; more/less service done than necessary	a. User input errors b. Misleading or incorrect details provided	a. Force vehicle owners to provide proof of ownership and vehicle details b. Provide vehicle owners with option to upload photos/videos along with service request c. Give service providers a warning when vehicle owners do not provide photo/video evidence	a. SR-10 b. SR-11 c. SR-12	H5-1
	Large number of quotes received for the same vehicle	Degradation of database performance; frustration or confusion for automotive service providers	a. Denial of service attacks b. Malicious use of quote submission system	a. Restrict vehicle owners to submitting one quote request per service provider.	a. SR-13	H5-2
Work Orders	Work order does not auto create or delete upon appointment booking or cancellation	Missing work orders (could cause some required parts to not be acquired), or unnecessary work orders (could case wasted expenses on unneeded parts).	a. Error while running create or delete operations on the work order database	a. Retry database operations multiple times, with increasing intervals between retries. b. Send information about these work orders to the shop owners (email or app notification)	a. SR-14 b. SR-15 c. SR-16	H6-1

Table 2: Failure Mode and Effect Analysis

6 Safety and Security Requirements

SR-1:

The system shall block account login for LOCKOUT_TIME after LOGIN_TRIES failed attempts.

Rationale: Unlimited login attempts could leave the system vulnerable to brute force password attacks.

Associated Hazards: H1-1.

SR-2:

The system shall enforce secure password practices during sign up or password change attempts by blocking low security passwords.

Rationale: Common passwords such as 'qwerty', 'password', '12345' or short passwords are highly vulnerable to malicious login attempts. Most modern account systems have basic password security requirements such as minimum length, capitalization, and symbols.

Associated Hazards: H1-1.

SR-3:

The system shall be reasonably safe against password hash collision.

Rationale: Improper implementation of password hashing could lead to increased chance of collision, resulting in successful login using multiple passwords and therefore increasing the risk of malicious access through brute force attack or other methods. Selecting a hashing function with a low risk of collision improves security.

Associated Hazards: H1-1.

SR-4:

The system shall communicate with the database immediately upon an appointment creation between a customer and a shop.

Rationale: If the database is configured to update only a few times daily, there is the potential that a second interaction to schedule an appointment at a taken time may occur, causing an overwrite of the first appointment.

Associated Hazards: H2-1.

SR-5:

Users will only be able to view available appointments.

Rationale: If the user who is scheduling an appointment is able to view all appointments, taken or not, they may potentially be able to overwrite a scheduled appointment, either due to an exploit on the web page, or a bug in the code allowing them to schedule even though it is taken. It is simpler to just remove visibility of taken appointments to eliminate any potential hazard in double-booking.

Associated Hazards: H2-1.

SR-6:

Users will only be able to send a message once every 2 seconds.

Rationale: If a user spams multiple messages, it could potentially result in slowdown or crashing of the application's designating server/database cluster. To prevent this, it makes sense to simply block a single user from sending multiple requests at a time.

Associated Hazards: H3-1.

SR-7:

Users will be notified through an error message toast if the server/database hosting service is currently down.

Rationale: If one of the hosting services used stops working, users may be unexpectedly unable to perform basic actions such as searching for work orders, chatting with shops, creating appointments, updating profiles, etc. This could be confusing for the user as loading could take infinitely long so it should be communicated to the user.

Associated Hazards: H3-1.

SR-8:

All database requests will be split into smaller queries through pagination if the query exceeds the maximum limit defined by the subscribed database hosting package.

Rationale: If there are many shops in the database and a vehicle owner attempts to search for all shops, the query may be too large, fail, and end up returning nothing. So, it is much better to use pagination for query requests.

Associated Hazards: H4-1.

SR-9:

All server requests that take longer than 2 seconds will require a loading spinner somewhere on the actionable component to notify the user results are still loading.

Rationale: If there is still more data coming to the user's interface, they should be aware of it, otherwise they may think that there are no results from a query or too few.

Associated Hazards: H4-1.

SR-10:

The system shall provide vehicle owners a method to attach photos and videos to a service request. A maximum of one video and ten photos will be allowed per service request.

Rationale: This will make estimating cost and turnover time easier for the automotive service provider. Limits on media helps save space on any related storage devices and improves loading times.

Associated Hazards: H5-1.

SR-11:

The system shall only store media related to a service request for the lifetime of that request.

Rationale: This further saves space on related storage devices, improves database performance and reduces the chance of personal data breaches.

Associated Hazards: H5-1.

SR-12:

The system shall provide shop employees and shop owners with a warning message when viewing a service request with no attached media.

Rationale: Without photo or video evidence, it will be harder to estimate cost/turnover time of required service to the vehicle.

Associated Hazards: H5-1.

SR-13:

The system shall limit vehicle owners to submitting only one quote request per automotive service provider, per vehicle.

Rationale: In a normal use case, users should need no more than one quote request per automotive service provider for each of their vehicles.

Associated Hazards: H5-2.

SR-14:

The system shall retry database operations a set number of times, with increasing time intervals between them.

Rationale: The system could be temporarily down/experiencing outages, in which case a retry and a later time could succeed.

Associated Hazards: H6-1.

SR-15:

The system shall notify shop owners of work orders that should be ignored or of work orders that have not been created, but an appointment has been made, in the event that all retries fail.

Rationale: If all retries for a database operation has failed, then something is probably fundamentally wrong with the system. If so, then no database operations will be made, so shop owners must be notified directly

Associated Hazards: H6-1.

SR-16:

The system shall notify developers and maintainers if all retries have failed.

Rationale: The same rationale as SR-15, however developers and maintainers will need to actually identify the issues and fix it.

Associated Hazards: H6-1.

7 Roadmap

In order to actually meet the requirements of the MVP, we will need to implement most of the safety requirements. Notable exception would be those pertaining to H3-1 (SR-8, SR-9) and H5-2 (SR-13). We do not expect a large number of shop owners or customers, therefore a query that is too large for the database will not be a scenario that we will run into. DOS/DDOS attacks and other malicious uses of the system can also be largely ignored in order to meet the MVP. These features will be postponed until later, however they could prove to be necessary in the future.