

Software Requirements Specification (SRS) Automotive Paint Defect Reporting System

Authors: Andrew Astakhov, Brandon Burt, Collin Skonieski, Jeremy Specht, Nick Frederick

Customer: General Motors Paint Defect Client Analysts

Instructor: Dr. James Daly

1 Introduction

This introductory section describes the purpose of the document, scope of the project, definition of the various terms used in the document, and the organization of the following sections.

1.1 Purpose

This document will describe the full requirements of the Automotive Paint Defect Reporting Software. It will describe constraints and a prototype for this system, including an outline of how it will be used. This document is meant mainly for the team to ensure that project requirements are satisfied, but also for the customer to review.

1.2 Scope

We will be developing the “Automotive Paint Defect Report System”. It is an application that allows a user to create detailed reports about paint defects on automobiles using a computer. Compared to the original report-writing method, this will save paper and time. The goal is to make the entire report-writing process easier and more efficient. The software will also store the reports in a database to be accessed or changed later.

1.3 Definitions, acronyms, and abbreviations

Defect: A flaw or imperfection in a vehicle’s paint job, which can have three possible severity levels: 1, 5, and Special.

Paint Defect Audit: The examination of an individual vehicle for paint defects at a given checkpoint. The location, type, and severity of any defects found are recorded.

Polish Deck: One of the checkpoints where the vehicles are audited for paint defects.

E-Coat: One of the checkpoints where the vehicles are audited for paint defects.

Prime Review: One of the checkpoints where the vehicles are audited for paint defects.

Quality Analysis Report: A detailed summary of the paint defects discovered at a specific checkpoint on a given day, compiled with the results of selected individual audits for that checkpoint.

Defect Analysis Report: A brief summary of the defects discovered at all checkpoints at a specific GM plant on a given day.

DPU: Defects Per Unit.

Daily Report: A summary of the paint defect statistics for a single day at a given GM plant. The report emphasizes defects per unit.

Weekly Report: A summary of how the paint defect statistics have changed over the course of a week at a given GM plant. The report emphasizes defects per surface.

Monthly Report: A summary of how the paint defect statistics have changed over the course of a month at a given GM plant. The report emphasizes finding potential trends in the data.

Horizontal: Refers to defects on the roof or hood of the vehicle.

Vertical: Refers to defects on the side of the vehicle.

1.4 Organization

Including the introduction, there are seven sections in this document. The second section contains an overall description of the project. The third section looks at the specific requirements that the project must achieve. The fourth section contains models to help visualize how the system will run from different perspectives. The fifth section gives details about the prototype that will be developed and how to use it. The sixth section is a short reference list for any other documents that may be used. The seventh section is a simple contact information list for the developers of this project.

2 Overall Description

This section gives an in-depth description of the perspective and functions of the product, the user characteristics, constraints, assumptions, dependencies and the apportioning of requirements

2.1 Product Perspective

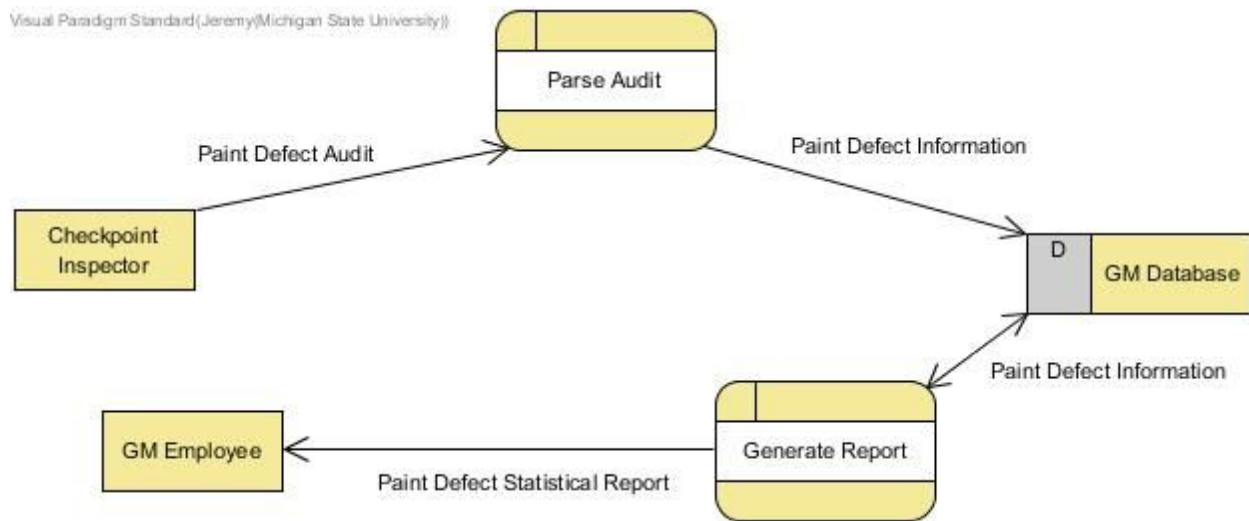


Figure 1: DFD Diagram of related systems

The Automotive Paint Defect Reporting Software is an interface for creating paint defect audits and storing them on a database, as well as retrieving them from the database and doing data analysis on those audits to create various reports.

The user will have a simple interface to choose between making different kinds of audits, creating reports based on audits in the database, or searching for audits/reports on the database.

2.2 Product Functions

The product will allow users to create detailed audits about paint defects by entering information into the software at a computer. The audits may come from different checkpoints in the assembly line. These audits will include a chart of the car showing the location and type of defect, and numeric information about the number of defects found such as the DPU.

Users will also be able to save these audits to a database, and later access, change, or print the audit as desired. Users can also enter certain days, weeks, or month-long periods and to get an automatically compiled report on all the audits that were made during that time. These reports can also be stored on a database, and accessed and printed later.

2.3 User Characteristics

The user of the system should be a client analyst that has experience with the analysis and reporting of defects on General Motors vehicles. This includes knowledge of paint defect terminology. The user should also have prior experience with the system. In the case that the user has no experience with the system, they shall be given a training session with a more experienced analyst.

2.4 Constraints

The saving of documents is constrained by the size of the database. If the database reaches capacity, storing new reports will become impossible. Saving documents to the database is also constrained by network connections. If the network cannot connect to the database, the system will be unable to save the file on the database.

Additionally, this system is created for a set number of vehicles. In the event that one of these plants began producing different vehicles, the system would potentially need to be modified to handle the new type of vehicle.

2.5 Assumptions and Dependencies

For our system we will assume that the checkpoint users are knowledgeable about paint defects. When inputting paint defect type/categorization data they will be entering this information directly without the help of a drop down menu. We are also acting under the assumption that the checkpoint consoles where the data is entered will be connected to remote GM database servers.

2.6 Apportioning of Requirements

Based on negotiations with our customers, the following requirements were determined to be addressed in future releases of the system:

- Ability to link between all GM automotive plants.
- Biometrics to log on to and use the system.
- Reports that can be distributed via the cloud to other locations.
- Reports based on other vehicles in the GM lineup.

- A generalized report that contains data from both the QA report and Defect Analysis.
- A visually appealing and interactive way to select where defects are on the vehicle.
- An area where the type of defect is described and labeled in the reports.

3 Specific Requirements

1. When the software is run on a system, the first thing a user will see is a login page, asking for a username and password. Authorized users will be able to log in using a username and password.
 - 1.1. Users will have different permission levels. Basic users will be able to use all the features of the software, administrative users will be able to create and delete basic users, and a developer user will be a single account that can create and delete administrative users.
2. Once logged in, a user can then choose to create several kinds of audits.
 - 2.1. These include Paint Defect Audits for each of the checkpoints, Polish, E-deck, and Prime.
 - 2.2. These audits can be saved and stored on a database, which will record the user submitting the audit and the time it was submitted. These can be viewed and edited later by any user
3. Users will be able to generate a daily/weekly/monthly report using all the audits logged on a certain day/week/month
 - 3.1. Reports will be compiled from audits on the day/week/month entered by the user
 - 3.2. Users can submit the reports to a database. The stored report will record the inputted information, as well as the user's name and date of the report.
 - 3.3. Users can later search and find these reports and open them again.
4. Users can print reports out

4 Modeling Requirements

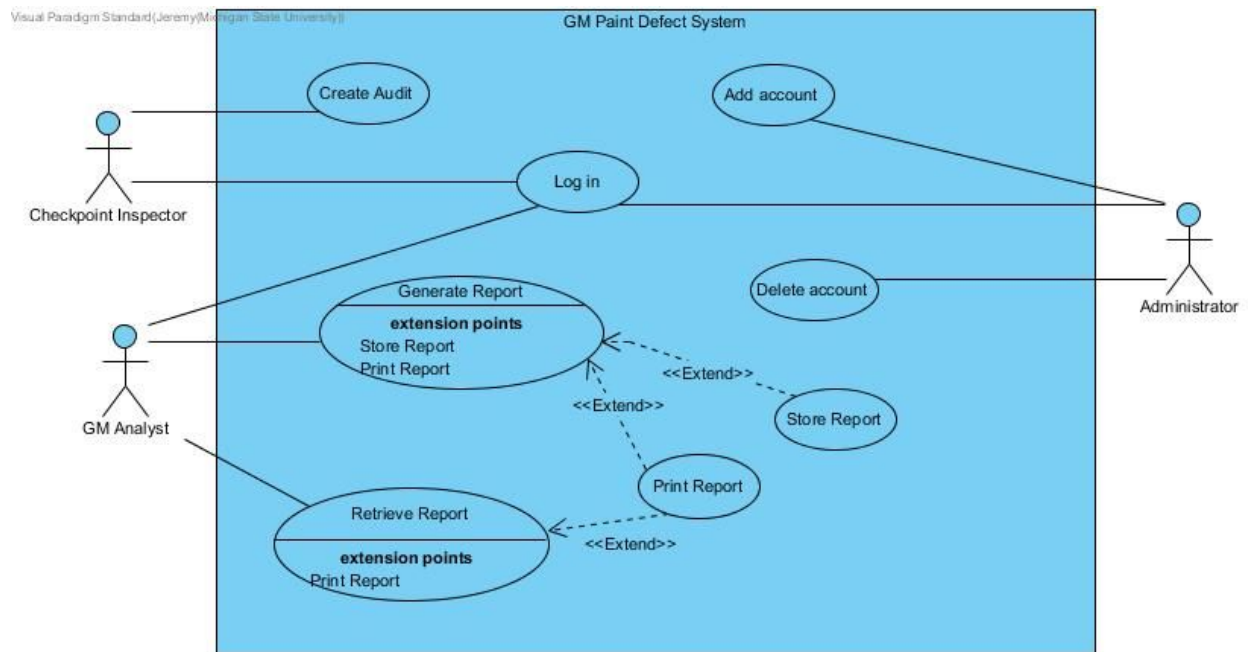


Figure 2: Use Case Diagram

This use case diagram demonstrates how users will need to log in, satisfying requirement 1. For requirement 2, the use case diagram shows checkpoint inspectors can create audits (which will be stored). For requirement 3, the use case diagram shows users (GM Analysts) can make reports, and then store them and access it later. For requirement 4, the use case diagram shows how GM analysts can print the reports.

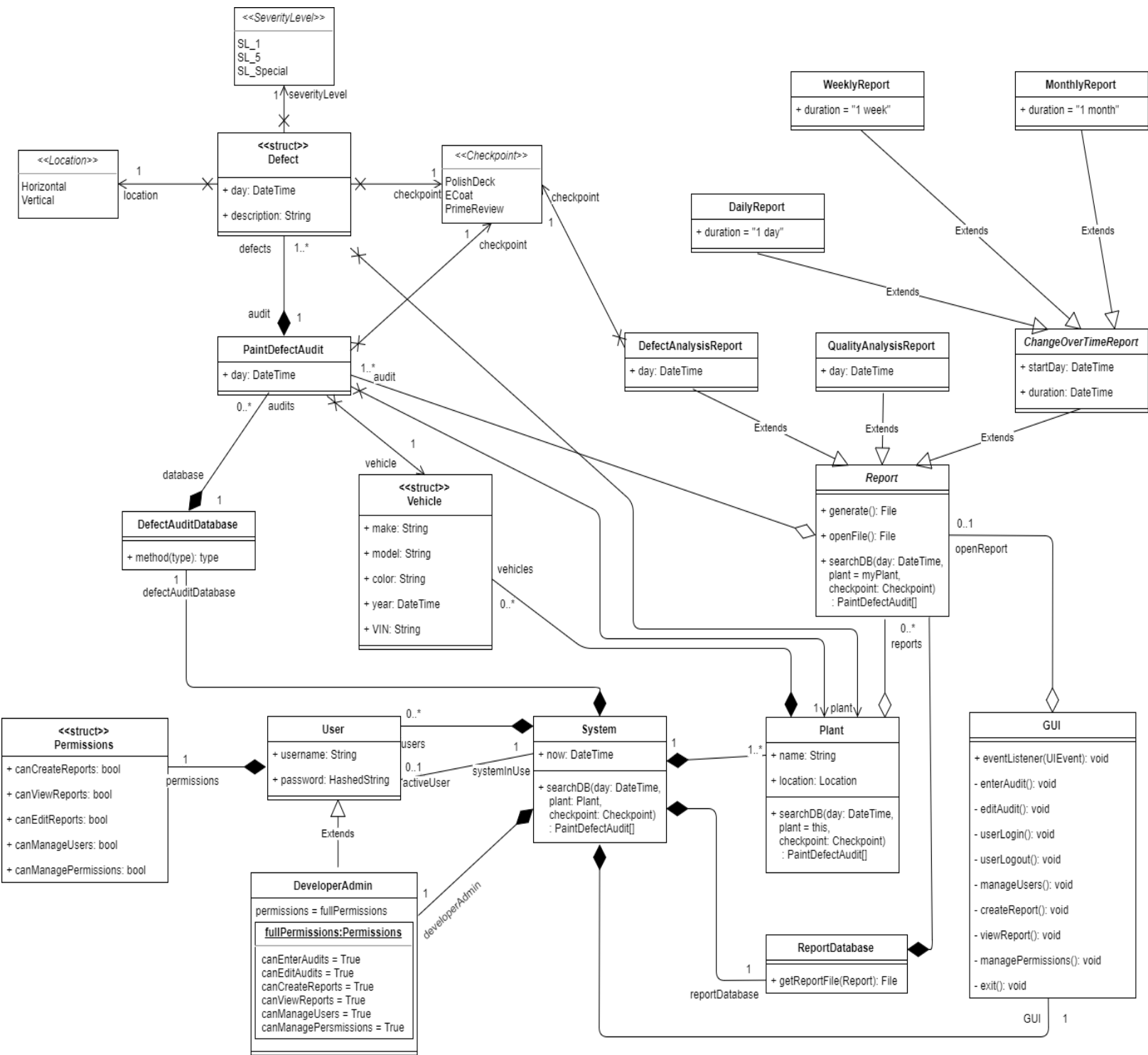


Figure 3: Class Diagram

Object Name	Description	Datatype/ return type
System (class)	Central controller for entire system	
Now (attr.)	Tracks current time	datetime
searchDB (method)	Method to search audit database for audits	paintdefectaudit[]
User (class)	Represents each user	
Username (attr.)	User's unique name	string
Password (attr.)	User's password	Hashedstring
Permissions (struct)	Keeps track of what user is allowed to do	
canCreateReports (attr.)	Tracks if user is allowed to create reports	bool
canViewReports (attr.)	Tracks if user is allowed to view reports	bool
canEditReports(attr.)	Tracks if user is allowed to edit reports	bool
canManagerUsers(attr.)	Tracks if user can manage other users	bool
canManagePermissions (attr.)	Tracks if user can manage other user permissions	bool
DeveloperAdmin (class)	Special user who can manage others but cannot be managed by a others, extends user	
GUI (class)	Class that manages the view users see	
eventListener (method)	Deals with U.I. elements, what the user clicks	
enterAudit(method)	Method for user to create a new audit	
editAudit(method)	Method for editing an existing audit	
userLogin(method)	Method for dealing with users logging in	
userLogout(method)	Method for signing users out	
manageUsers(method)	Method allowing users with permissions to edit other users	
createReport(method)	Method for user to create new reports	
viewReport(method)	Method to display existing reports	

managePermissions (method)	Method for allowing users with correct permissions to change other user permissions, except for DeveloperAdmin user	
exit(method)	Method for closing down the program safely	
ReportDatabase (class)	Class that manages connection to database holding reports	
getReportFile (method)	Method for retrieving reports from DB	File
Report (class)	Class representing a report, meant to be extended	
generate(method)	Generates report file based on audits	File
openFile(method)	Opens the file containing the report data	File
searchDB(method)	Used by generate to get audits	paintdefectaudit[]
DefectAnalysisReport (class)	Extended Report class that represents a defect analysis report	
QualityAnalysisReport (class)	Extended Report class that represents a quality analysis report	
ChangeOverTimeReport (class)	Extended Report class that is meant to be extended further	
DailyReport(class)	Extended ChangeOverTimeReport representing a daily report	
WeeklyReport(class)	Extended ChangeOverTimeReport representing a weekly report	
MonthlyReport(class)	Extended ChangeOverTimeReport representing a monthly report	
DefectAuditDatabase (class)	Class that manages connection to database holding audits	
searchDB (method)	Method that finds audits on the database	paintdefectaudit[]
PaintDefectAudit(class)	Class representing an audit	
Day (attr.)	Attribute describing the day an audit was made	datetime
Defect(class)	Class representing a single defect	
Day (attr.)	Attribute describing when defect was found	datetime
Description (attr.)	Short description of the defect	string
Severity (attr.)	Attribute describing how severe defect is	severityLevel

Location (attr.)	Attribute describing where defect is on car	Location
Checkpoint (attr.)	Attribute describing which checkpoint the defect was found on	checkpoint
Plant (class)	Class representing one of the plants/factory where system runs	
Name (attr.)	Attribute describing name of the plant	string
Location (attr.)	Attribute describing location of the plant	location
searchDB (method)	Method that finds audits on the database	paintdefectaudit[]

Figure 4: Data Dictionary

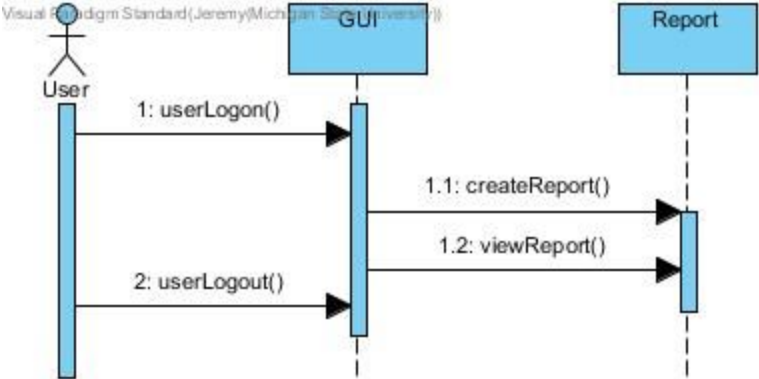


Figure 5: Sequence Diagram - Scenario 1

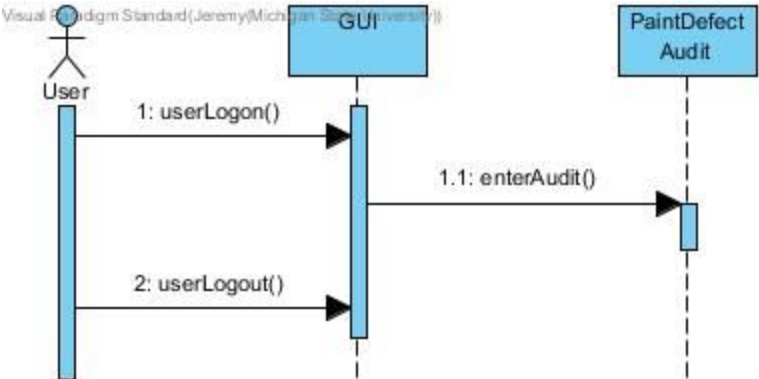


Figure 6: Sequence Diagram - Scenario 2

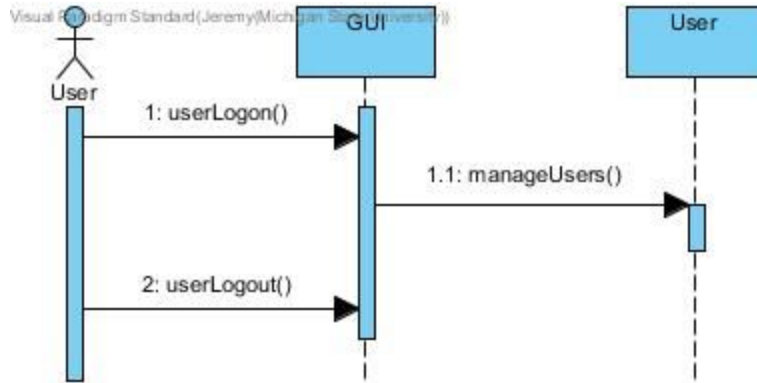


Figure 7: Sequence Diagram - Scenario 3

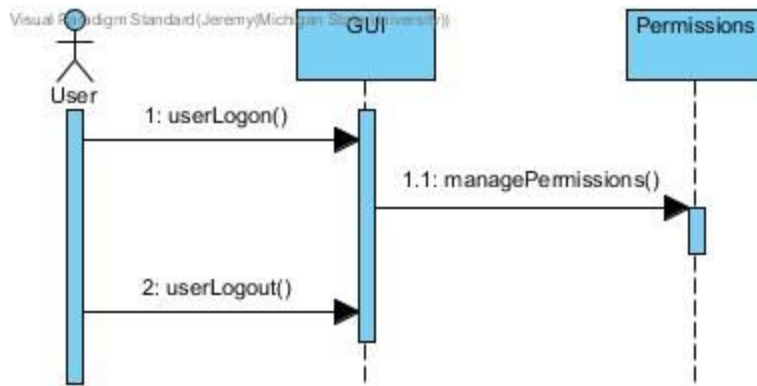


Figure 8: Sequence Diagram - Scenario 4

5 Prototype

The prototype will show a complete use of the system. The user will be able to enter the information about the defects found on the vehicle, and the system will then compile the information into low-level paint defect reports.

5.1 How to Run Prototype

The prototype is a GUI utilizing C++, and QT Creator. The reports are generated using R statistical software. The first prototype is only the interface of the system, and is not executable.

Figure 9: Main Window of the Paint Defect System.

- Audit Button: Creates a new dialog screen (Figure 2) that the user inputs information into.
- Drop Down Box 1: Contains the options for which GM Plant to create the report of.
- Drop Down Box 2: Contains the options for which time option to create report of.
- Drop Down Box 3: Contains the options for which report to create.
- Generate Report Button: Generates the report with the selected options

The screenshot shows an 'Information Input Window' with the following structure:

- Vehicle:** A dropdown menu.
- Plant:** A dropdown menu.
- Right Vertical:** A section containing three input boxes labeled 'Sev 1', 'Sev 5', and 'Special'.
- Left Vertical:** A section containing three input boxes labeled 'Sev 1', 'Sev 5', and 'Special'.
- Hood:** A section containing three input boxes labeled 'Sev 1', 'Sev 5', and 'Special'.
- Roof:** A section containing three input boxes labeled 'Sev 1', 'Sev 5', and 'Special'.
- Deck:** A section containing three input boxes labeled 'Sev 1', 'Sev 5', and 'Special'.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom center.

Figure 10: Information Input Window.

- Vehicle: Contains the different vehicles to input information for.
- Plant: Contains the plant that the information is being inputted with respect to.
- Right Vertical: The right side of the vehicle.
- Left Vertical: The left side of the vehicle.
- Hood: The hood of the vehicle.
- Roof: The roof of the vehicle.
- Deck: The back deck of the vehicle.
- Sev 1: The number of Sev 1 defects.
- Sev 5: The number of Sev 5 defects.
- Special: The number of Special defects.

5.2 Sample Scenarios

The following scenario is of a user generating a Defect Analysis Summary.

The user will select All Automotive plants, Weekly summary, and Defect Analysis Summary.

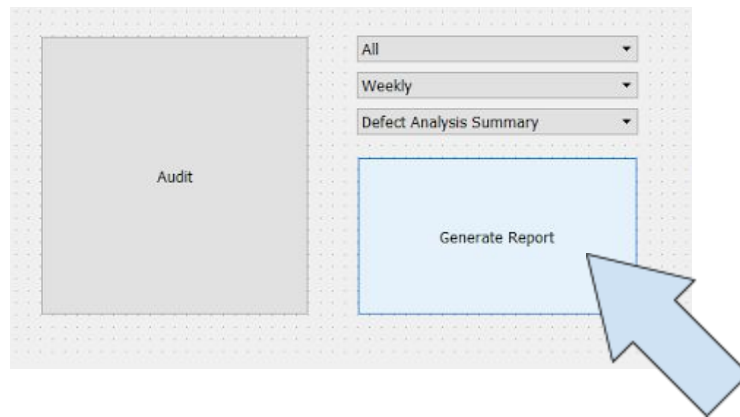


Figure 11: Sample Scenario

The user will then click Generate Report and the outputted report will contain the information in figure 2. This report contains the Analyst's name that generated the report, the plant(s) that were included, and the time frame of the report. Also included is a statistical summary of the number of defects and units, and the defects per unit. The final piece of the report is the Location statistics. The location statistics include each location, the total number of defects, and the average defects per unit.

Defect Analysis Summary

Analyst: Jim Whaddaname
 Plant: All
 Time: Weekly

Summary:
 # of Defects: 221
 # of Units: 70
 Defects per Unit: 3.15

Defect Location	Defect Total	Defects per Unit
Right Vertical	45	.64
Left Vertical	20	.28
Roof	67	.95
Hood	39	.55
Deck	50	.72

Figure 12: Defect Analysis Summary

6 References

D. Thakore and S. Biswas, "Routing with Persistent Link Modeling in Intermittently Connected Wireless Networks," Proceedings of IEEE Military Communication, Atlantic City, October 2005.

Burt, Brandon. "Automotive Paint Defect Analysis". November, 2017.
<https://webdev.cse.msu.edu/~burtbra3/cse435/>

7 Point of Contact

For further information regarding this document and project, please contact Dr. J. Daly at Michigan State University (dalyjame at msu dot edu).