Software Requirements Specification (SRS)

Quality Analysis Report Automation (QARA)

Authors: Jacob Weber, Mario DiMartino, Doug Kantor, Ryan Mathews, Jesse Wolven

Customer: GM Lansing Delta, Lansing Grand River, and Lake Orion Assembly Plants

Instructor: James Daly

1 Introduction

This SRS document will detail everything involved with the QARA system. In the coming sections you will learn everything you need to know for operation of this system. Section 1.1 will explain the purpose of the QARA system, as well as the necessity for its creation. Section 1.2 will dive into the scope of the project and explain what the software does. Section 1.3 will define all potentially confusing terms used throughout the SRS document.

Section 2 will dive into the product itself and explain the system in more detail. Section 2.1 is about the product perspective and will describe the context of the system and how it will fit into the bigger system. Section 2.2 will summarize the major functions that the software will perform. Section 2.3 provides the expectations for the user and gives a background on how the operator will use the QARA system. Section 2.4 provides the constraints of the QARA system. Section 2.5 provides all of the assumptions and dependencies made by the QARA development team. This includes hardware, software, the work environment, and user interactions. Section 2.6 provides requirements that are beyond the scope of this system but may be explored further in later iterations of QARA.

Section 3 provides specific requirements for the QARA system. All of the requirements are provided numerically

Section 4 will provide models of the QARA system. This will include a use case diagram, high level class diagram, sequence diagram, and lastly a state diagram. These diagrams will depict all of the different uses of the software and provides all of the different operations, member variables and relationships.

Section 5 will describe the different modes within the prototype and give an idea of what the final version will look like for the operators. This details both INPUT mode and REPORT GENERATION mode. Instructions on how to operate the prototype are provided in section 5.1. Section 5.2 will give a sample scenario of how the prototype runs.

1.1 Purpose

The purpose of this document is to show how the QARA system is used. This document will inform operators examining vehicles on the assembly line for flaws in the finish of all the features of the QARA system along with a detailed description on the use of these features. It will also inform developers of design constraints for implementing this system, as well as inform stakeholders of the product they will be investing in.

1.2 Scope

The scope of this system involves using the QARA software for operators to record data on the flaws in the finish of vehicles on the assembly line. The QARA system is used for the automatic generation of proper reports from data entered by the operator. Reports are generated by the system automatically or at the request of the operator, covering a specified amount of time. The operator uses these reports to determine issues with the production line regarding the finish of the vehicles. The benefits of using this system is that it will reduce paper usage and increase efficiency within the plants. The data being recorded in the system includes the location, severity, and the type of flaw. This data is then used to find the defects per unit (DPU) of the different car models. The QARA will run as a desktop application on the Windows Operating System. The system will not automatically detect defects on the automobiles, these will have to be manually entered by the operators. The system will also not share data between the different plant locations.

1.3 Definitions, acronyms, and abbreviations

Defect- Flaws or imperfections in the paint and finish
DPU - Defects Per Unit
Horizontal - The roof and hood of the vehicle
HW- Hardware
QARA - Quality Analysis Report Automation
SEV - Severity
SRS - Software Requirements specification
SW - Software
Vertical - The sides of the vehicle
Deck - Trunk of the vehicle

1.4 Organization

This Software Requirements Specification document will go through many different sections and will outline all the different aspects of the software. Section 2 will give an overall description of the project and dive into the Product Perspective, Product Functions, User Characteristics, Constraints, Assumptions & Dependencies of the software, and Approportioning of Requirements. Section 3 will go into the Specific Requirements of the project. Section 4 will give the Modeling Requirements of the software. Section 5 will show prototype information and provide instructions with how to run the prototype, along with a sample scenario for the operator to test with. Section 6 will show all references for this document.

2 Overall Description

Section 2 will describe all of the different aspects of the software. Section 2.1 will show the product perspective. This will show the different sequence of events that will occur with the software and show some of the constraints that are within the QARA system. Section 2.2 will dive into the different functions and features of the software. This includes the major functions of the software along with a description of what they do. Section 2.3 will dive into the background required for the user and what is expected for the user to know for operation of this software. Section 2.4 will dive into the constraints of the software. Section 2.5 will discuss some of the assumptions and dependencies that the software uses. This includes things like the hardware, environment, user interactions, and other software being used along with QARA. Section 2.6 will dive into the Approportioning Requirements. This involves information from the negotiations with the customers, and shows the requirements that are determined to be beyond the scope of the QARA system.

2.1 Product Perspective

QARA is the automation of the process currently in use for analyzing paint defects in vehicles manufactured at a given plant. At checkpoints along the assembly line, the operator examines vehicles for flaws in the paint and finish and marks down the location, type, and severity of the flaw. After a sufficient sampling of vehicles, the collected data is collated and a report is generated on the nature of the flaws, which is then analyzed for any potential problems with the manufacture of the vehicles.

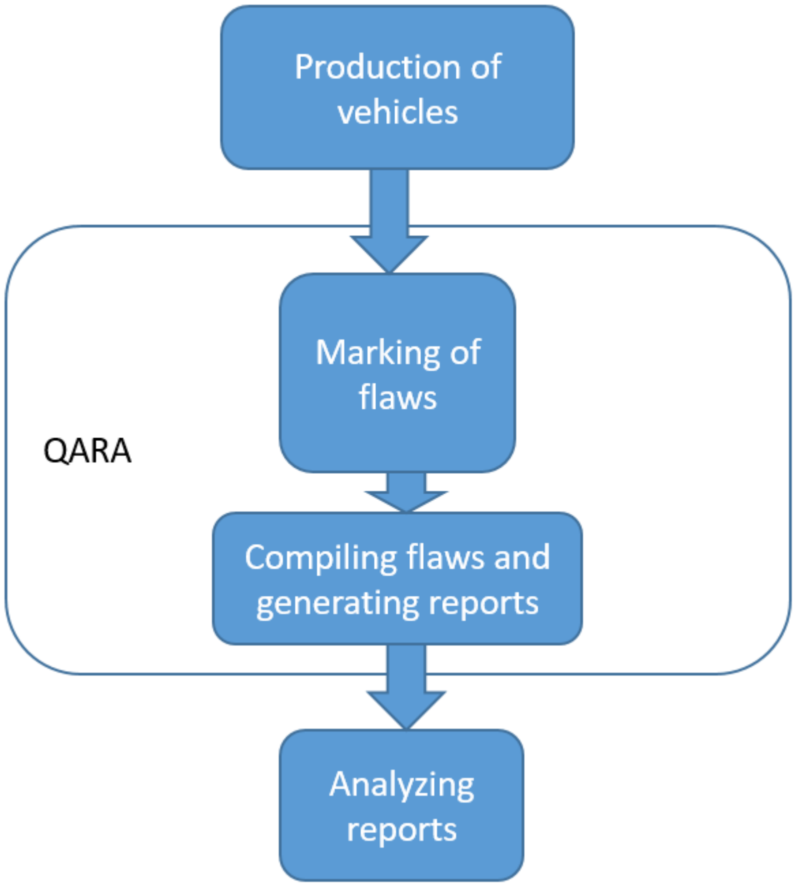


Figure 1: Data flow diagram outlining how the QARA system fits into the current system

The QARA system will take as input the location, type, and severity of flaw as well as the model of vehicle examined. It will then automatically collate the data and generate a report for a time specified by the operator, such as a daily or weekly report.

There are some constraints that could happen with the software. The hardware must be able to run the software and should have the capability to type in data and click through menus. The hardware must have enough memory for installation of the QARA application. The hardware must also provide some form of portability to allow for the operator to examine a vehicle on the assembly line.

2.2 Product Functions

During the examination of the vehicle, the major function of QARA will be to take input from the operator about the vehicle model, flaw locations, flaw types, and severities of the flaws and save this record to a database. The second major function of QARA will be to compile all of the records and generate reports based on the data collected over a specified amount of time. This second function occurs automatically to produce daily and weekly reports, but may produce a report by any operator-specified time period.

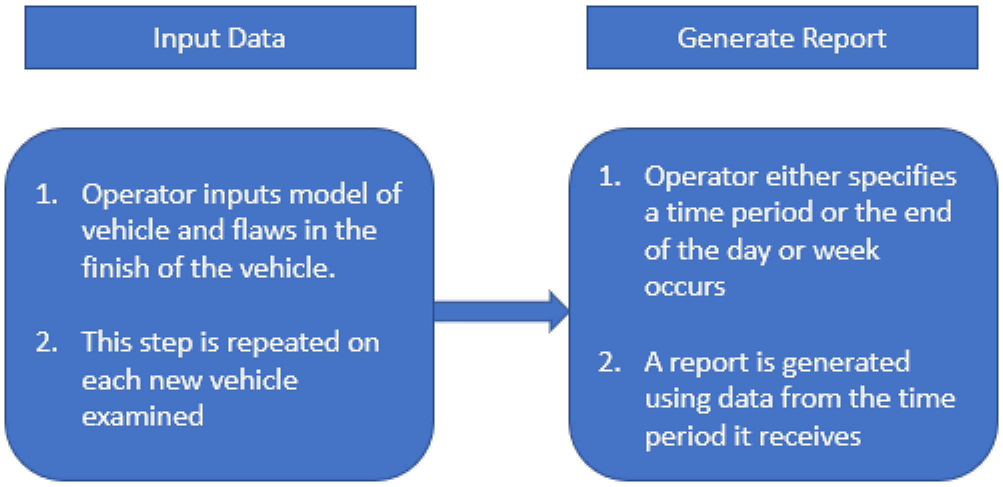

Input Data
Generate Report
1. Operator inputs model of vehicle and flaws in the finish of the vehicle.
2. This step is repeated on each new vehicle examined
1. Operator either specifies a time period or the end of the day or week occurs
2. A report is generated using data from the time period it receives

Figure 2: High-level goal diagram. Outlines the processes the operator will perform to achieve their goal of automatic report generation.

2.3 User Characteristics

The user is expected to have expertise in identifying and describing the flaws found in the finish of a vehicle. The user must be able to differentiate between all the different types of flaws and correctly identify any and all flaws. The user must be able to recognize the severity of these flaws and correctly give a severity level to any and all flaws. The user must also be able to navigate the GUI interface of the software.

2.4 Constraints

There aren't many constraints with this system, but the system should accurately depict the physical vehicle it is representing. This means that the user must accurately place the defect in the software and accurately describe its type and severity. As far as this system goes, there are limited worries about safety-critical properties relating to human safety. But, for safety of the data in the system it will be critical that all markings go onto the model of the vehicle and that the data be recorded as such to limit corruption. The system will not perform properly and will not produce accurate results if the analyst uses the wrong model for the vehicle or enters other incorrect data.

2.5 Assumptions and Dependencies

It is assumed that the hardware to be used is a tablet, with a Windows operating system. The environment QARA will be used in will be along the vehicle assembly line. The user will interact with QARA during the input of the location, type, and severity of flaws, as well as the model of the vehicle examined. User interaction will also occur when the operator requests a report to be generated over a specific period of time.

2.6 Approportioning of Requirements

Providing the name of the operator examining the vehicles as well as the name of the operator that requested a report to be generated on the report itself was deemed uncritical at this point of time. While this may be addressed later, it was unimportant for demonstrating the general performance of QARA. Being able to upload new wireframes to QARA to match new vehicle models being produced as well as only allowing validated operators from accessing the system were also determined to be uncritical for the first prototype.

Sharing a database between all three factories and generating reports containing information from each factory was determined to be beyond the scope of QARA at this point of time. This may be addressed later, but it is not required for the basic operation of QARA.

3 Specific Requirements

1. The operator can input location, type, and severity of flaw, as well as vehicle model.
    a. The location is indicated by color-coded markings on a wireframe of each vehicle.
    b. The QARA system will store this data as an entry in a database.
2. The operator can edit previous entries in the database.
3. At the request of the operator, the QARA system will generate a report based on all input data over a period of time specified by the operator.
4. The QARA system will generate a daily and weekly report automatically.
5. The data in the database will be held for an indeterminate amount of time.

4 Modeling Requirements

Use Case

The operator uses the QARA system to add a new record, edit an existing record, or generate a report. An SQL database also interacts with the QARA system during adding and editing records and report generation. The database saves the records and provides existing records.

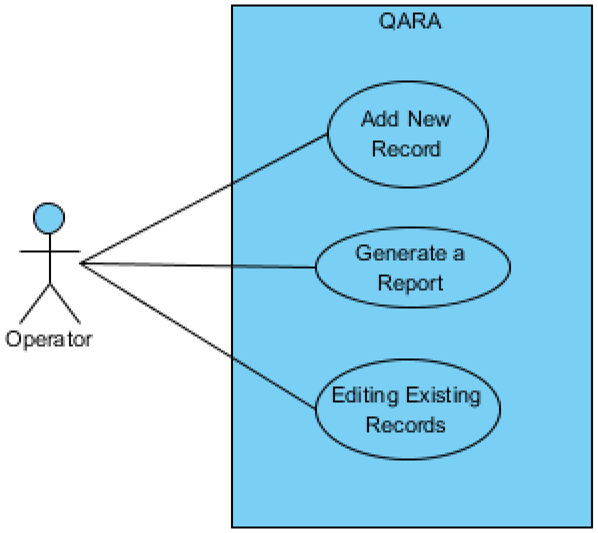


Figure 3: Use case diagram of the QARA system

Class Diagram

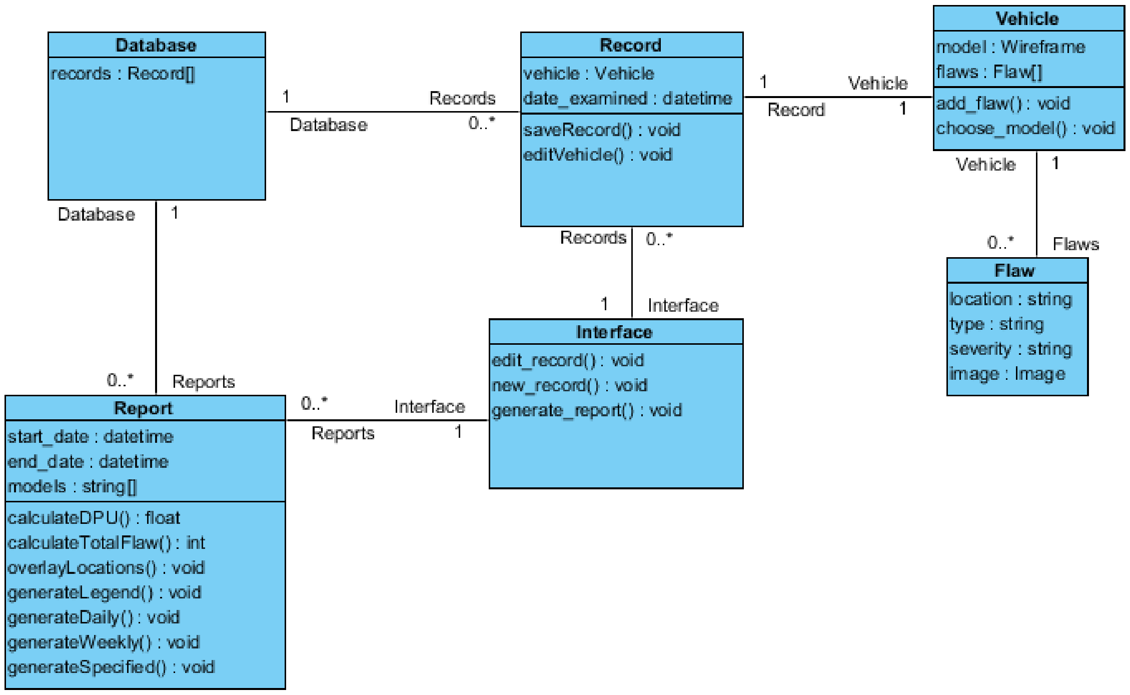


Figure 4: Class diagram of the QARA system

Through the Interface class, the operator can edit or add records, as well as generate reports. The Record class records a Vehicle and its list of Flaws, the date and time that it was examined, and whether or not it has been edited. The Vehicle class contains the model of the vehicle, which is expressed by the wireframe, as well as a list of the Flaws for that vehicle. The Flaw class represents the flaws found in the finish of the vehicle, and indicate the location, type, and severity of the flaws. The Report class takes a start date and an end date as well as the list of vehicle models to include to generate a report analyzing the data from that time period.

Data Dictionary:

Database		**SQL database where Records are stored**
Attributes		
	records: Record[]	Representation of the data the SQL database will be storing
Relationships	References Record	

Flaw		**Details the location, type, and severity of a flaw**
Attributes		
	location : string	General area flaw is located at
	type : string	Type of flaw
	severity : string	Severity of the flaw
	mark : Ellipse	Marking on wireframe detailing where flaw is located
Relationships	Is referenced in Vehicle	

Interface		**Class allowing for the operator to interact with the system**
Operations		
	editRecord(): void	Pulls Record from the Database to allow editing
	newRecord(): void	Creates new Record for data entry
	generateReport(): void	Calls the Report class to generate a report
Relationships	Calls the Record and Report classes	

Record		**Details the vehicle examined and the time at which it was examined**
Attributes		
	vehicle: Vehicle	Vehicle being examined

	date: datetime	Date and time vehicle was examined
Operations		
	editVehicle():void	Allows the Interface class to edit the Vehicle class
	saveRecord():void	Saves the Record to the Database
Relationships	References Vehicle, Is saved to Database	

Report		**Used for generating the reports over a specific time frame requested by the operator**
Attributes		
	start_date:datetime	Start date for the time frame
	end_date:datetime	End date for the time frame
	models: string[]	List of models to include in the report
Operations		
	calculateDPU(): float	Calculates DPU of all models
	calculateTotalFlaw(): int	Calculates total amount of flaws
	overlayLocations():void	Overlays all flaw markings over a wireframe
	generateLegend():void	Generates legend for the flaw markings
	generateDaily():void	Generates a daily report
	generateWeekly():void	Generates a weekly report
	generateSpecified():void	Generates a report over a specified time period
Relationships	Pulls data provided by Record from the Database	

Vehicle		**Class containing the wireframe of the vehicle model and a list of flaws**
Attributes		
	model : Wireframe	Wireframe of vehicle model
	flaws : Flaw[]	list of flaws
Operations		
	add_flaw (): void	Adds a new flaw to list of flaws
	choose_model(): void	Choose the model of vehicle examined
Relationships	Contains a list of Flaws, Is referenced in Record	

Sequence Diagram

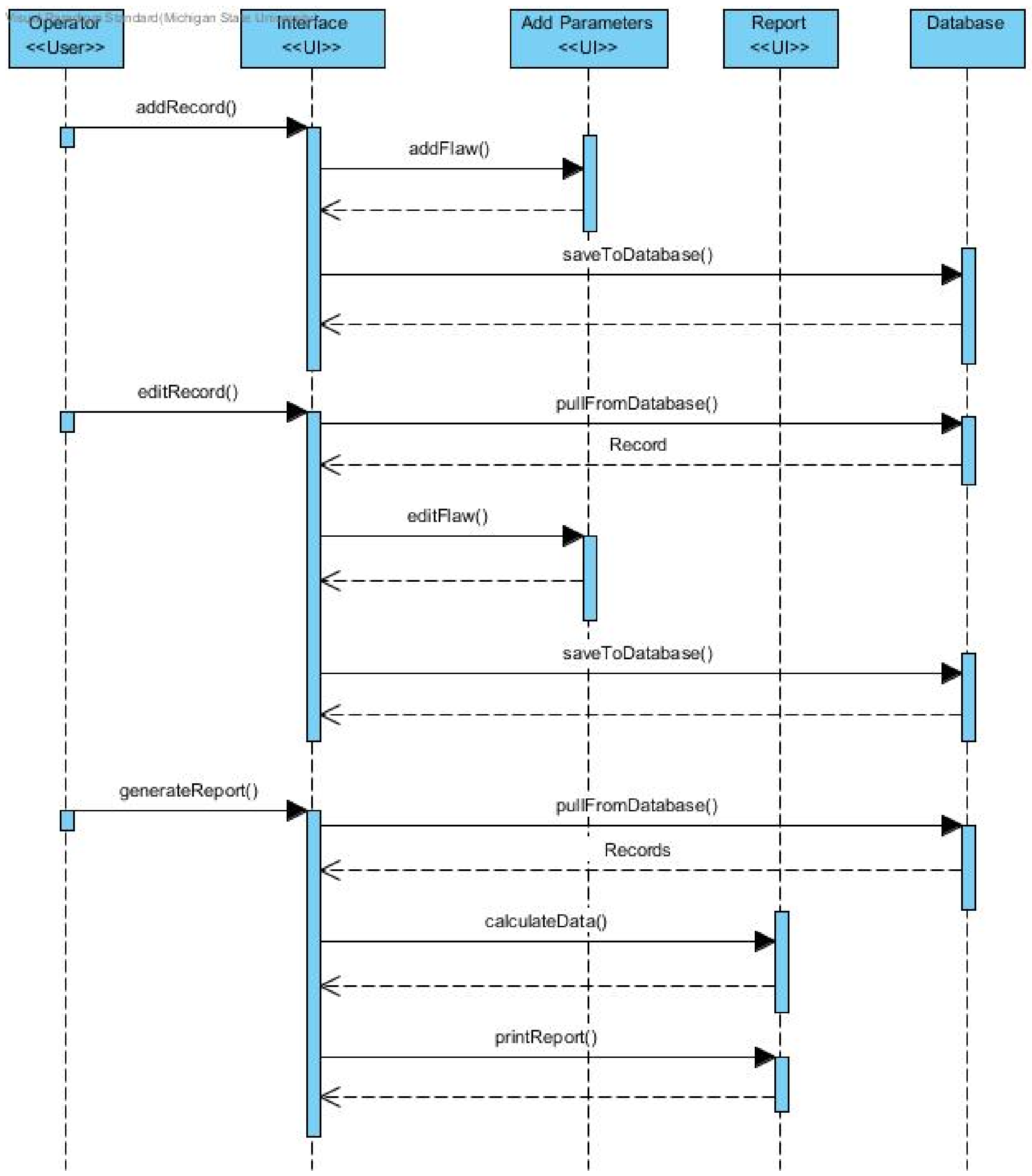


Figure 5: Sequence diagram for QARA system

The scenario for the sequence diagram in Figure 5 is this. The operator wants to add a new record to the database for a vehicle they are examining. They only noticed a single flaw, so once they have chosen the model of the vehicle, they input the location, type, and severity of the flaw. Thinking they are done, they save the record to the database. However, they notice almost immediately that they missed a flaw. They select the option to edit a record, which pulls the record they just saved from the database. They edit in the new flaw, then save it again to the database, replacing the old save. That vehicle was the last for the day, so the operator requests a daily report to be generated.

This pulls all of the records created during that day from the database and collates the data into the report.

State Diagram

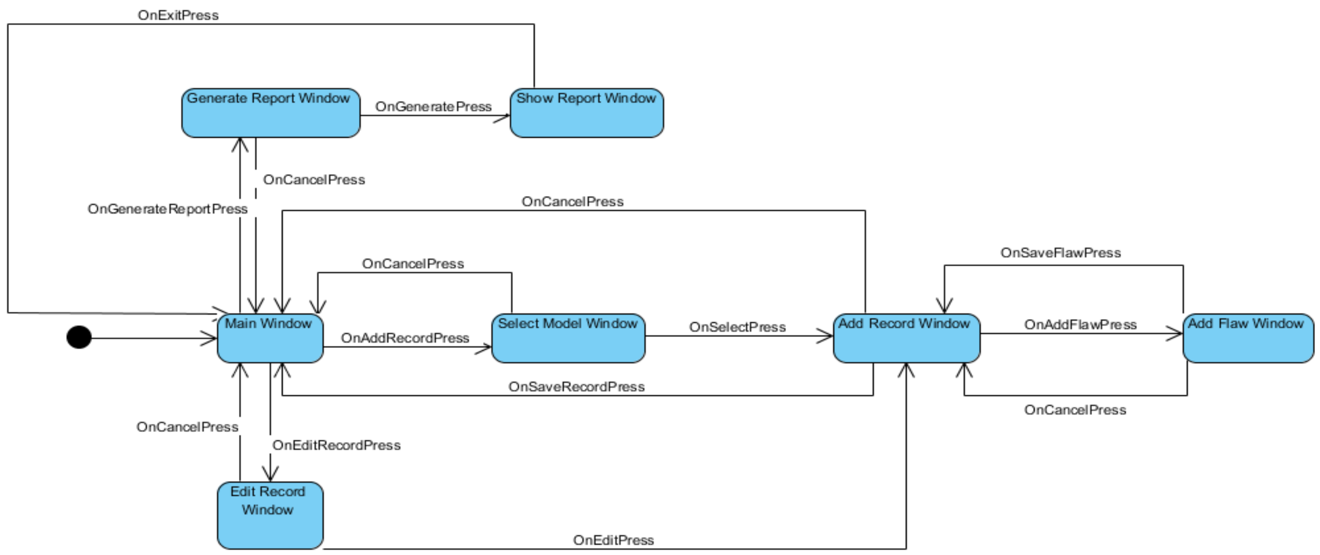


Figure 6: State diagram for QARA system

This state diagram starts off at the Main Window. From here the user can either choose to Add Record, Edit Record, or Generate Report. Once the user clicks "Add Record", the user must click one of the available models within the software. Next the user may add flaws to the record by clicking the "Add Flaw" button. Then the user chooses the severity of the flaw and provides its location. The user can save this flaw to the record by clicking on the "Save Flaw" button. This completes the actions for one flaw, from here, the user can decide to add another flaw and repeat the previous step. Once the user finishes adding flaws, the user can click the "Save Record" button to save the record and return to the Main Window. Users can also choose to edit previous records or generate a report of all records in a certain timeframe.

5 Prototype

The prototype will give a representation of how the software will run for the operators out on the assembly line. This prototype will start with a menu including different options.

INPUT

After clicking this option, a prompt will appear asking for the make and model of the vehicle being inspected. Once the appropriate information is entered, an image of the wireframe is produced. On one side of the window, a button will appear saying "Add…". This will create a new prompt asking for 2 different inputs. These are severity, and type of the flaw. The "Add…" menu will also allow for the user to mark the location of the flaw on the diagram. After the user marks the location of the flaw on the diagram they will select from a drop down menu which part of the car the flaw is located on. After finishing this, a list below the “Add…” will show all of the different flaws that have been input for this specific vehicle. Once all flaws are input, the vehicle can be named, and the “Finish” button can be clicked to add the entry to the database, along with a timestamp of its entry.

REPORT GENERATION

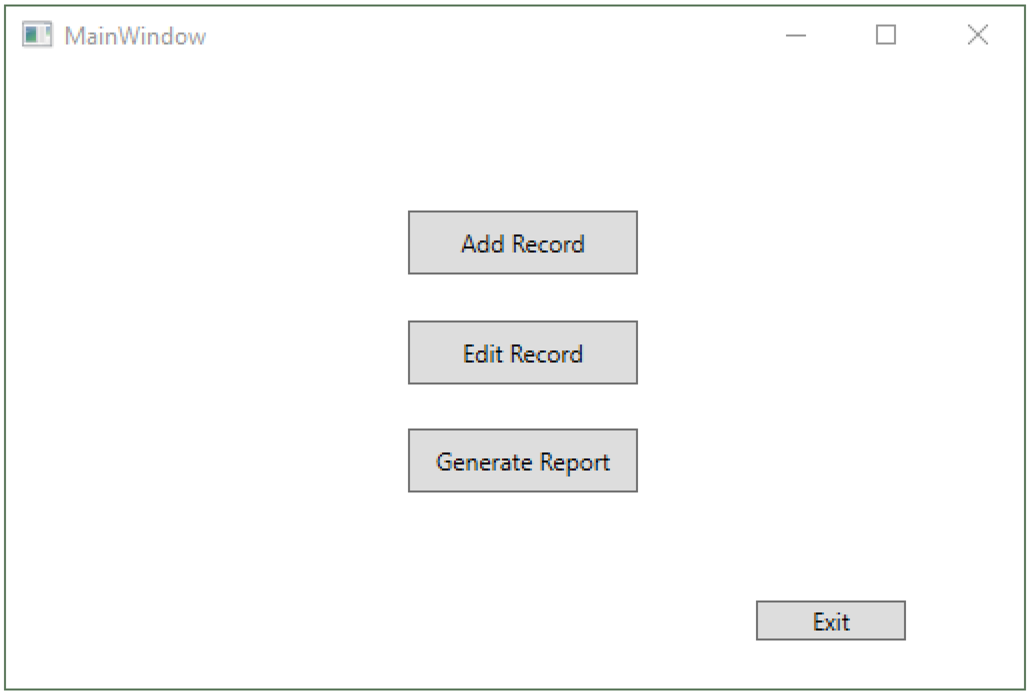
After clicking this option, a prompt will appear asking for the timeframe of the report desired. Once the beginning and end dates are entered, a report for that timeframe will be generated. The count of each severity and the DPU will be calculated and then displayed in a table. The markings indicating the location and type of flaw will be overlayed over each other onto a single wireframe, with a legend located on the left depicting type of flaw. The number of units, total defects, total DPU, and the timeframe will also be displayed at the top of the report.

5.1 How to Run Prototype

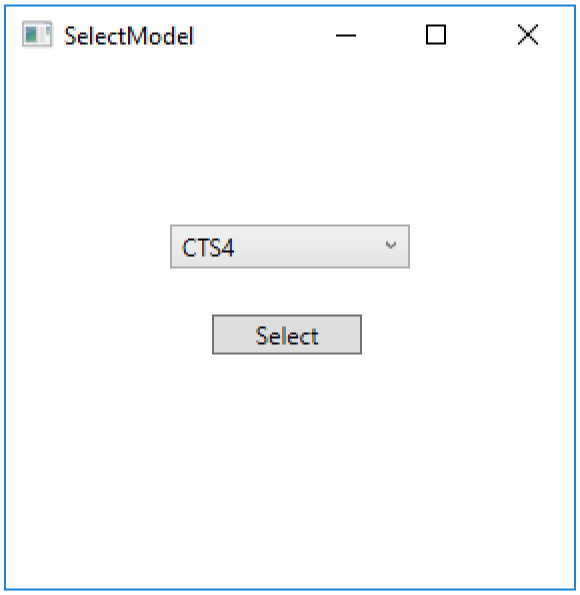
In order to run our prototype all that is needed is a machine running a Windows operating system that has access to the internet. The URL for the prototype is http://cse.msu.edu/~weberja4/cse435.

5.2 Sample Scenarios

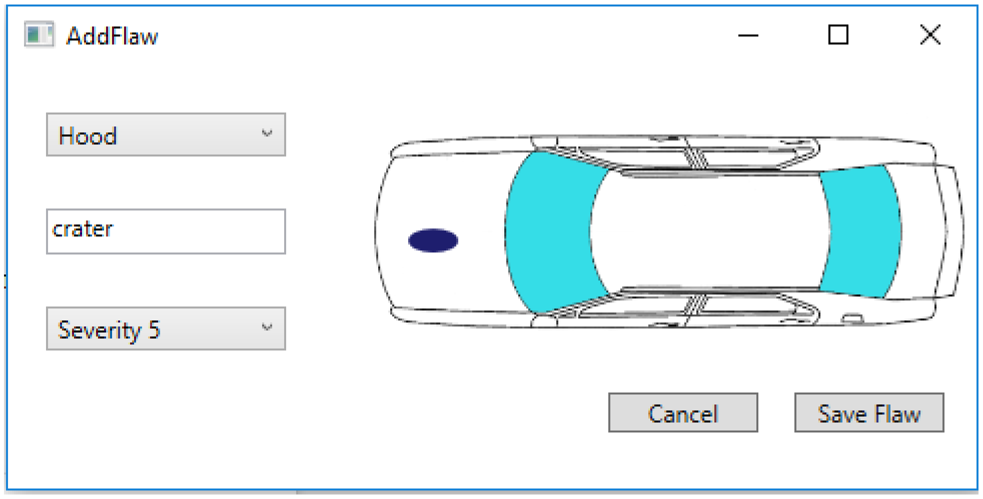
A sample scenario involves a Cadillac CTS coming down the assembly line and stopping at the operator station. Upon inspection of the vehicle, the operator notices a crater on the hood of the vehicle. The operator will open the QARA system and be presented with a menu with three different choices.



Once the operator clicks on “Add Record”, the operator will select CTS4 from the “Select Model” dropdown menu and click “Select”.



Next the operator will click the “Add Flaw” button on the AddRecord page. From here the operator will choose Hood from the “Flaw Location” dropdown menu. They will then type in Crater within the “Flaw Type” textbox. The operator will then select Severity 5 from the “Flaw Severity” dropdown menu. The operator will then draw on the image where the flaw is located and then click the “Save Flaw” button.



The operator will then click on the “Save Record” button. The Cadillac CTS then moves on down the assembly line and the next vehicle comes to the operator station.

6 References

Team 11 Website
http://cse.msu.edu/~weberja4/cse435

Automotive Paint Defect Analysis
Author: James Daly
Date: October 10, 2017
http://www.cse.msu.edu/~cse435/Projects/F2017/Project-Description.pdf

Project Lecture Slides
Author: James Daly
http://www.cse.msu.edu/~cse435/Projects/F2017/Project-Description-Assignment.pdf

Quality Analysis Report
Author: Sylvia Kulak
Date: February 9, 2011
Publishing Organization: Daimler Chrysler & Team Industrial Services
http://www.cse.msu.edu/~cse435/Projects/F2017/Resources/QualityAnalysisReport.pdf