FIT5032 Design Report (HD Code Task)

Warranty Repairs for War Heroes

Jeffrey Middendorf:27338096

Table of Contents

1. **Overview** 3
2. **Functional diagram** 5
3. **Core program functionality (description of core program functionality and how it works.)** 8
4. **Usability Design Review (based on the Flow Bohl usability categories)** 18
5. **Checklist of site functionality** 22
6. **User stories** 24
7. **Data dictionary** 25

1. Overview

The class of the FIT5032 has been asked to complete a high distinction task application as a part of the workload. One of the important steps of the application development life cycle is understanding its primary objectives. Assuming a standard waterfall cycle, the requirements form the basis of design and implementation. This program had a number of goals that development was trying to achieve. The primary goals of the high distinction task are development of SQL Server database with at least two related tables, use of the Model-View-Controller (MVC) architecture framework, jQuery demonstrations, and advanced functionalities of our choosing.

The primary goal of the application of the application was to demonstrate applied knowledge of the MVC architecture. This includes the utilization of three different files that will all work together to form your application. The first type of file that is included is the view files. The view files are files that are at the boundary of the application that provide the user interaction functionality. These views are managed by controller classes. The controller classes are the classes that return all of the views to the calling methods. The controller class is also responsible for managing the model. The model is the class responsible for interacting with the database. The development of the database is the second goal of this application.

The next goal of the application is developing a SQL Server database that has at least two table that are related. To accomplish this functionality, the primary key of one table must appear as the foreign key in another table. This data will then be used to as the basis for the Create, Read, Update, Delete (CRUD) functionality of the application. This will be created before any objects can be created, after the initial project creation. This is because the development is using a database first development model. So, the database will be developed first and all of the other MVC files and coding will be developed around supporting that database. The other model of code first, the database is developed, to match the code, is not in the scope of this class. The code can be further enhance by JavaScript.

JavaScript is a client-side scripting language that is used to enhance the functionality of an application. This could be used to add functionality that is not possible with server-side scripting. On example of a possible function that cannot not be completed on the server is an input date picker. These are too dynamic to require to requery the server every time you set focus on a form object that has the date picker. There are also numerous libraries that can be leverage so we would not have to write the date picker from scratch. This application will make use of jQuery, jQuery-UI, and jQuery-DataTables. These libraries can also be used to accomplish all of the jQuery demonstrations for this assignment.

MVC functionality, database development, and jQuery implementation are the major objectives of this project. A database-first MVC arichecture will provide the basis

for the coding. The coding will be used to service a SQL Server database that houses related information. The functionality will be further extended by the utilization of jQuery client-side scripting. These elements will work together to form a single project to serve as my high distinction task.

2. Functional diagram

Functional diagrams are a cornerstone of illustrating the functions and processes of the internet application. This application has three major functions that will be described using diagrams. Visio and Umlet will be the tools used to creat the functional diagrams for the website. Program structure is the most important non-functional requirement that will require a diagram. The requirements will also have to fully understood in regards to the application. A use case diagram will be used to illustrate the new requirements that were implemented. An Entity Relationship diagram for the database will also be provided to fully understand the application.

The layout of the program is composed of a layer that will serve as the boundary that allows the user interactions. The views are separated into the four categories for the different database table and a home view. Each category has their own set of views. The home category contains an index, documentation, and about page. The other views contain CRUD views to interact with the three tables. The views are managed by the controllers of the system. There is one controller for each of the categories of views.

The controller is also responsible for managing the model. The model serves as the connection to the database. The database will be the storage for all of the data in the application.

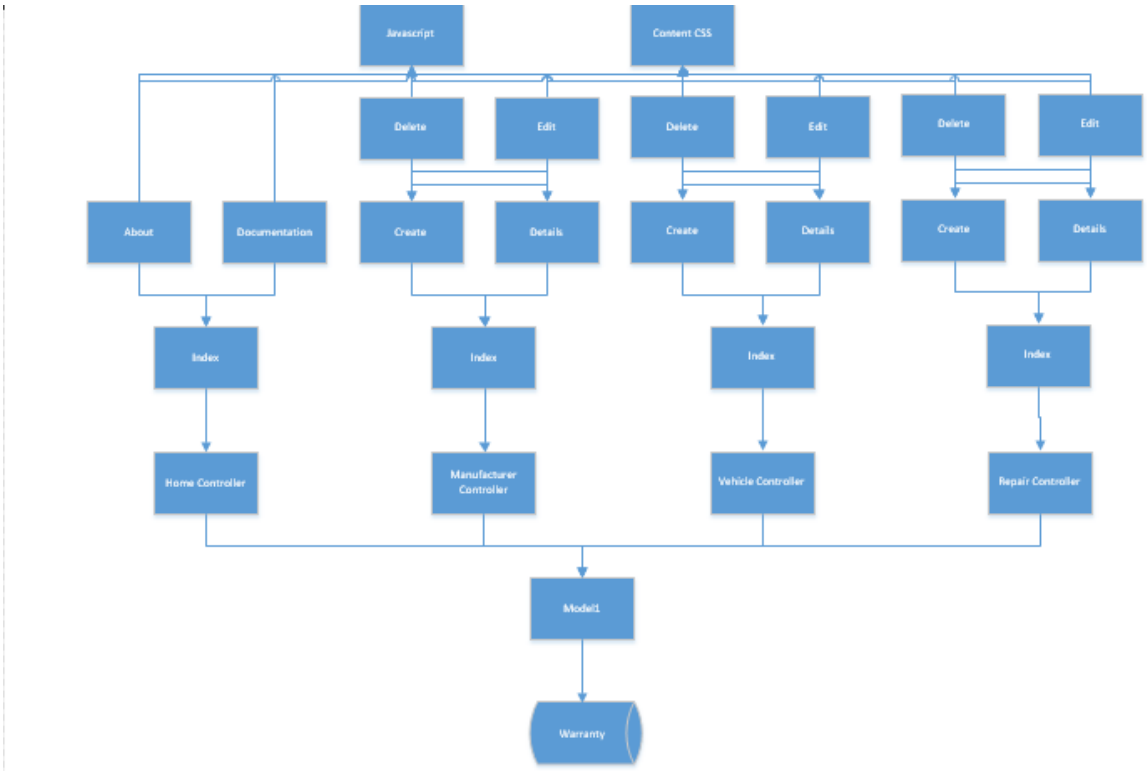


Figure 1 – Hierarchy of Application

Requirements analysis is another important consideration in the functional analysis of an application. One of the methods for understanding the requirements is completing a use case diagram. A use case diagram is one of the techniques that is utilised to understand the actors' interaction with the system. For the purposes of this assignment, the four use cases that will be modelled are the four CRUD functionalities.

These functionalities are used to manipulate the information in the database. The database will also have its own diagram.

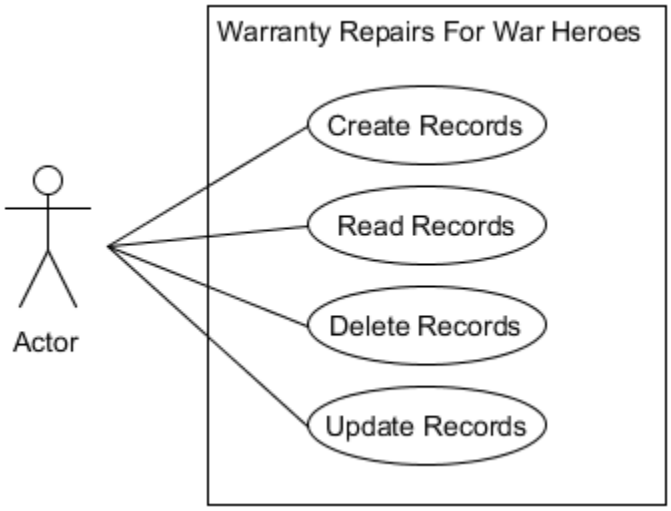


Figure 2 – Use case diagram

The entity relationship diagram is the diagram that will be used to show the model of the database. The ER diagram will use standard crows foot notation to model the degree and cardinality of the database. It will be a graphical diagram of all of the relationships that are in the database. It will also indicate the types of relationships like one to many. This diagram will be the last diagram for this assignment.

The database diagram, requirements analysis, and functional diagrams are important to the understanding the application. The functional diagrams are used to understand the architecture of the program. The architecture will implement the use cases from the use case diagram. The use cases are the CRUD function for manipulating

the data represented in the use case diagram. These will be used to implement the functionality required for the assignment.

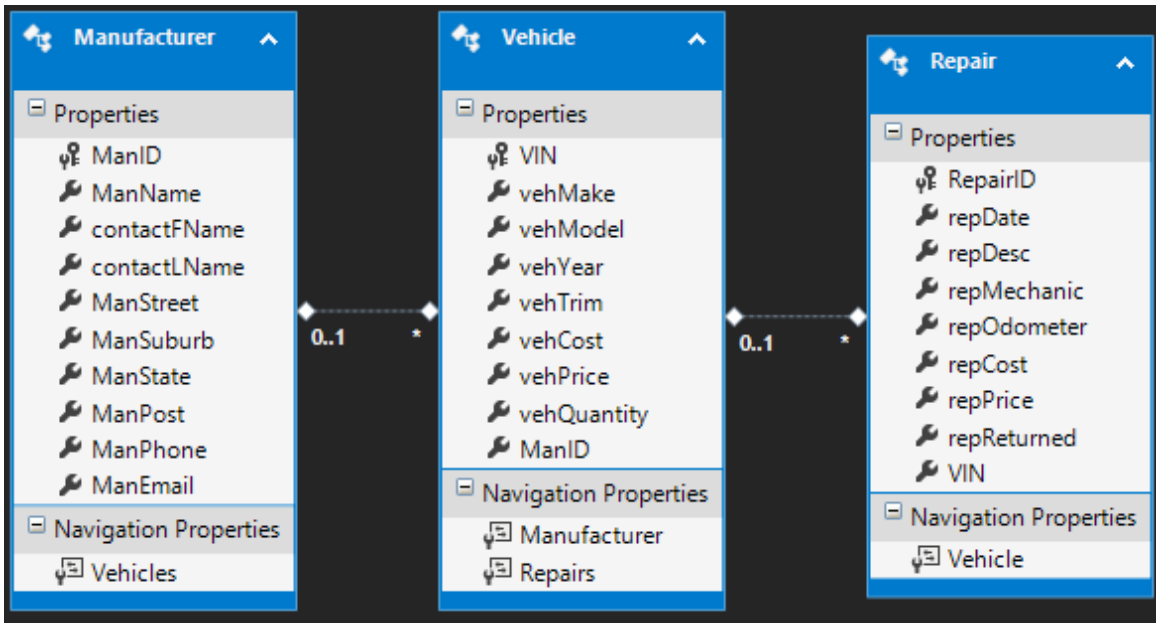
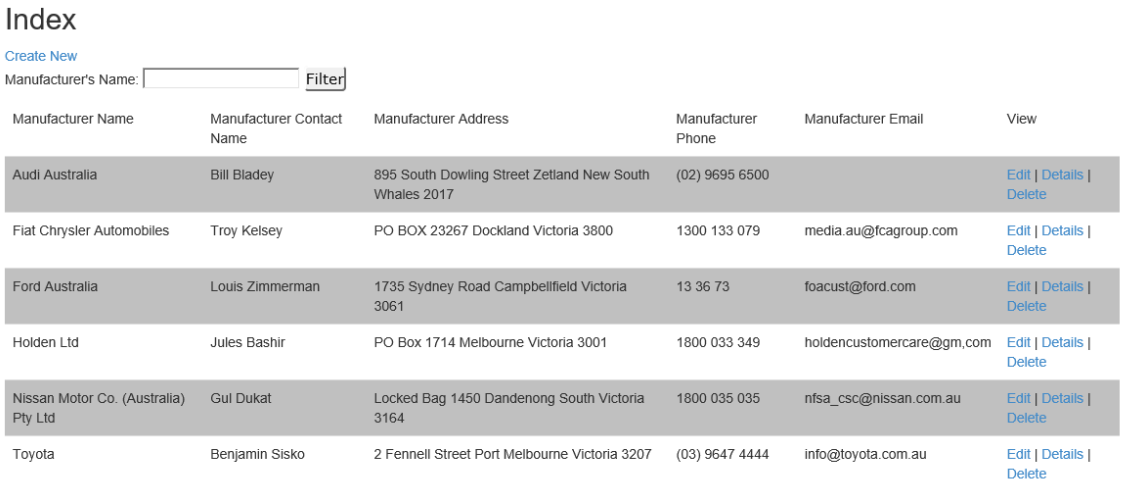


Figure 3 - ERD

3. Core program functionality (description of core program functionality and how it works.)

Functionality descriptions are a critical component of the software development process. The descriptions can be utilized to analyse the program from both a quality assurance and user acceptance testing purposes. This program has a number of requirements that are important considerations when reviewing the programs. The export to csv and Excel, jQuery, and CRUD are the important features that were implemented into this application.

The first functionality that was implemented was CRUD manipulation of the database. The first functionality that displays for the database is the read portion of the functionality.



Index

Create New

Manufacturer's Name: Filter

Manufacturer Name	Manufacturer Contact Name	Manufacturer Address	Manufacturer Phone	Manufacturer Email	View
Audi Australia	Bill Bladey	895 South Dowling Street Zetland New South Whales 2017	(02) 9695 6500		Edit | Details | Delete
Fiat Chrysler Automobiles	Troy Kelsey	PO BOX 23267 Dockland Victoria 3800	1300 133 079	media.au@fcagroup.com	Edit | Details | Delete
Ford Australia	Louis Zimmerman	1735 Sydney Road Campbellfield Victoria 3061	13 36 73	foacust@ford.com	Edit | Details | Delete
Holden Ltd	Jules Bashir	PO Box 1714 Melbourne Victoria 3001	1800 033 349	holdencustomercare@gm,com	Edit | Details | Delete
Nissan Motor Co. (Australia) Pty Ltd	Gul Dukat	Locked Bag 1450 Dandenong South Victoria 3164	1800 035 035	nfsa_csc@nissan.com.au	Edit | Details | Delete
Toyota	Benjamin Sisko	2 Fennell Street Port Melbourne Victoria 3207	(03) 9647 4444	info@toyota.com.au	Edit | Details | Delete

Figure 4 – Manufacturer Index View

This is displayed through the index view using html display for boxes. The index view of the manufacturer table also contains the single field search and alternating row colour.

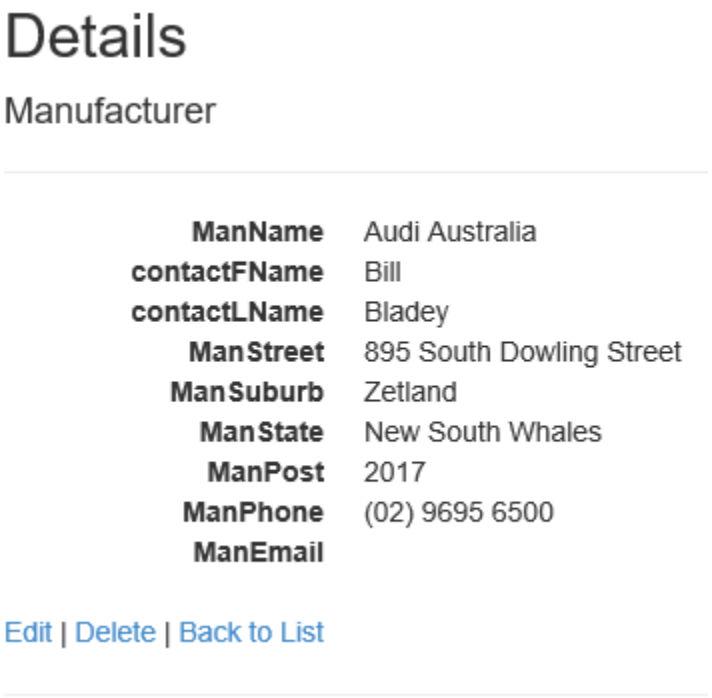


Figure 5 – Individual Record Read Screen

The hyperlink at the top of the page leads to the create view which allows records to be created in the database. The index view also contains a link to the detail view which is the view that shows the values for a single record.

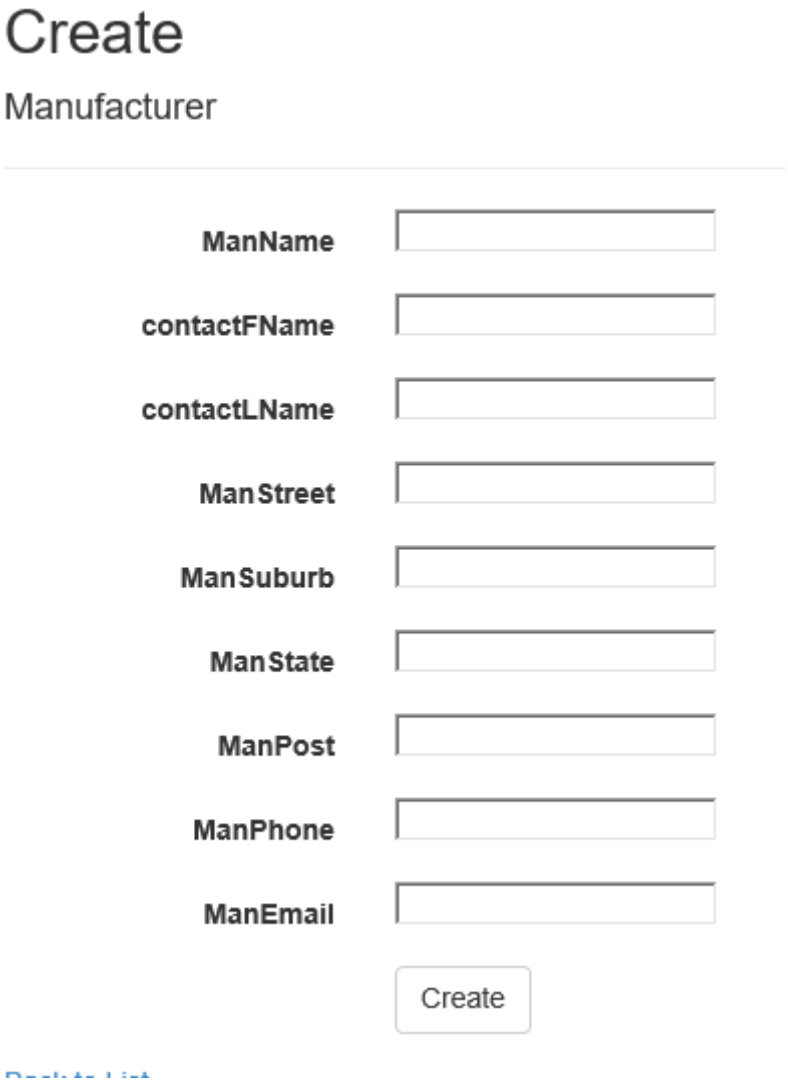


Figure 6 – Manufacturer Create View

This view was update to allow access to the delete view that provides the delete functionality. The edit view provides the final update functionality from the CRUD implementation.

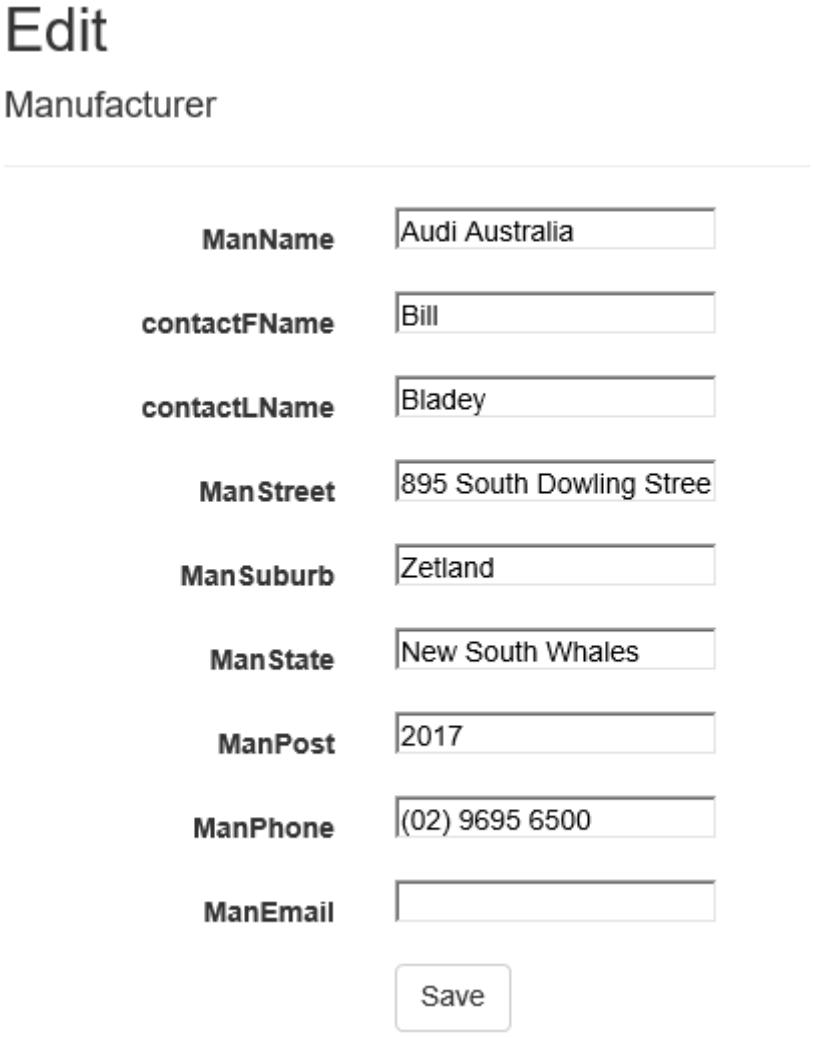


Figure 7 – Edit Manufacturer View

The CRUD functionality is then extended by the jQuery libraries to add client-side scripting to the program. The first jQuery functions that were implemented into the table were the database libraries. Row highlighting, search features, and results per page are implemented on the vehicle index view.

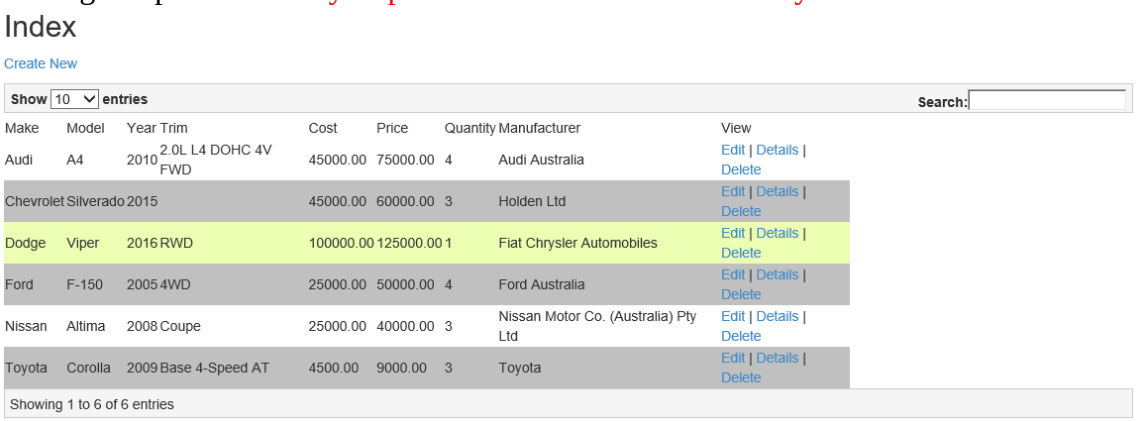


Figure 8 – Data Table Functionality Including Row Highlighting

The data picker feature was also added to the create repair view for the repair date and the returned date. This allows the date to be in the correct data format when it is entered.

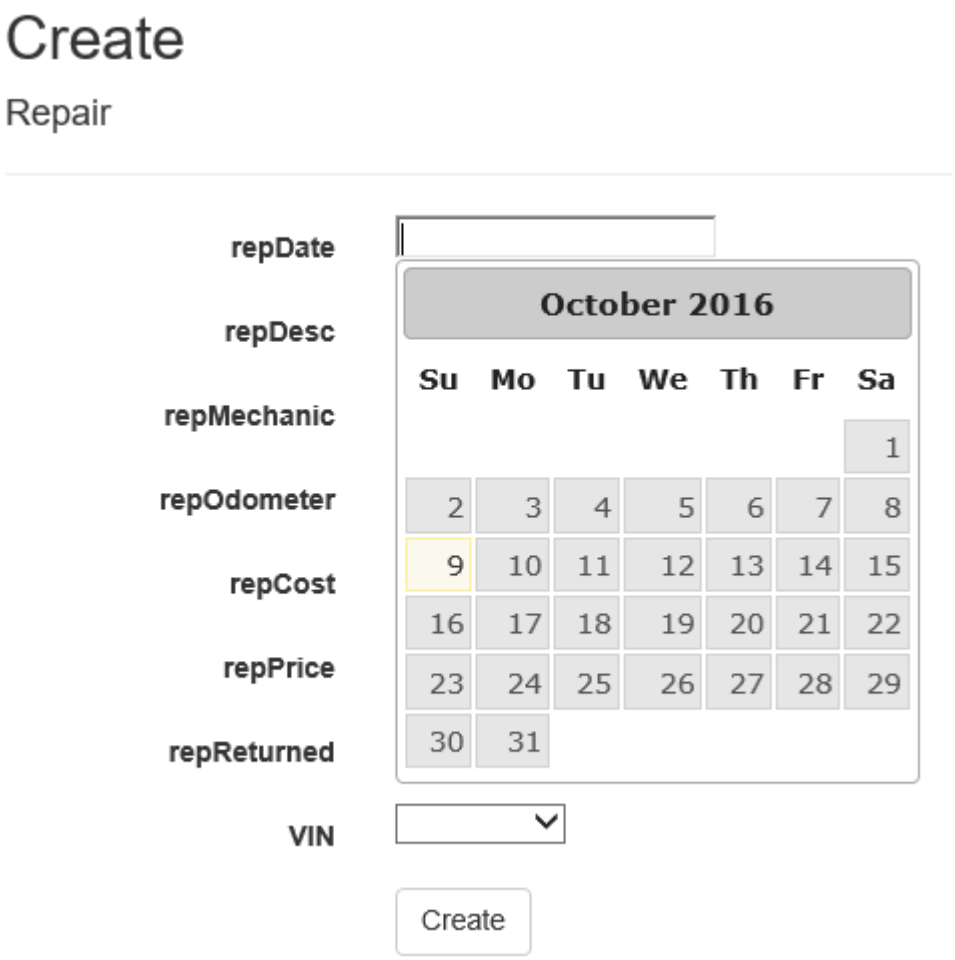


Figure 9 – Date Picker in Action

The About page has a jQuery Tab functionality to differentiate between the history and the frequently asked questions. The documentation page has a jQuery accordion to allow for collapsing and organizing data.

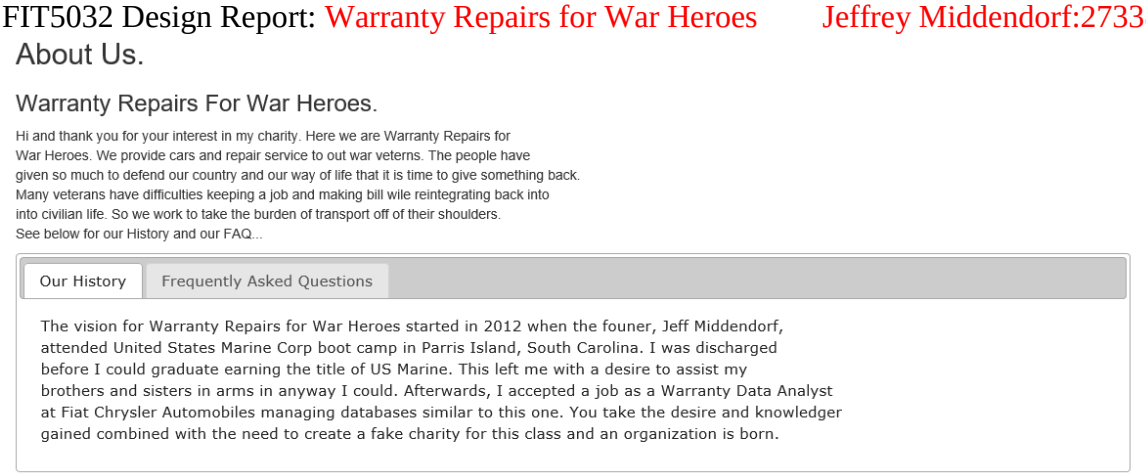


Figure 10 – Tab Implementation in About Us

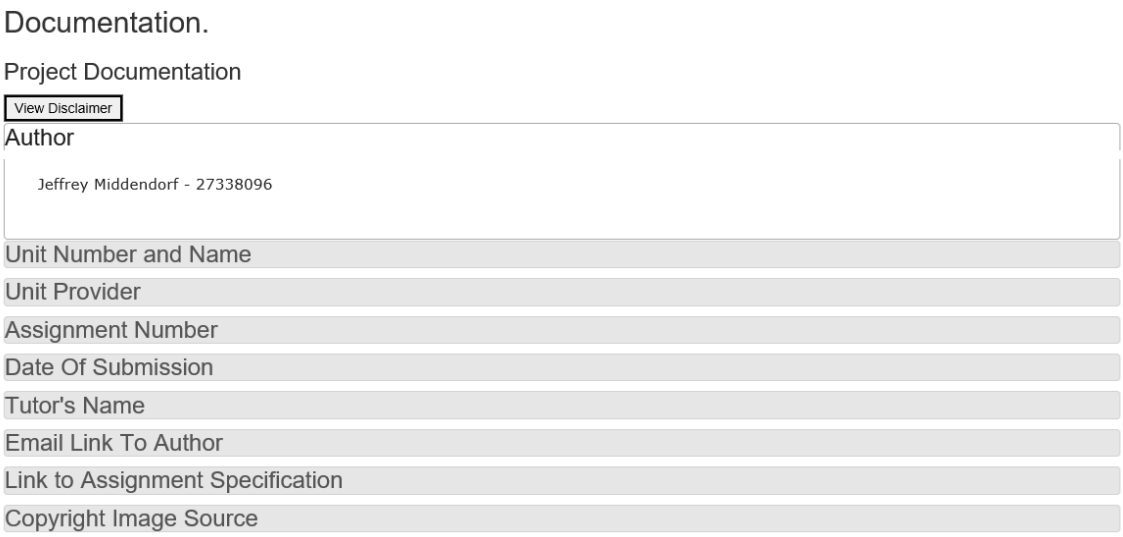


Figure 11 – Accordion Implementation in Documentation

There is also a tooltip that adds to the feedback for the application on the manufacturer index view. This view also has a jQuery button to execute the search functionality.

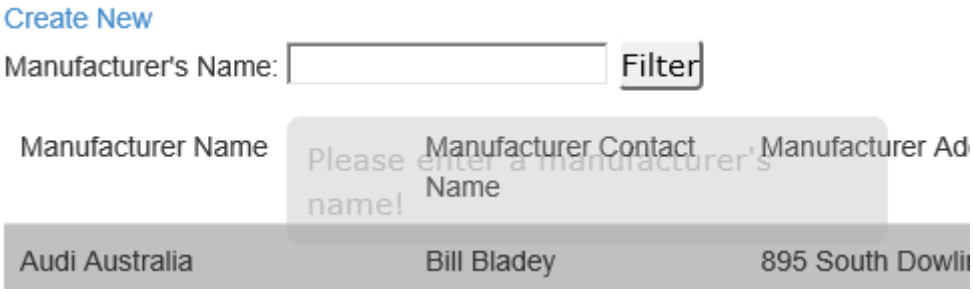


Figure 12 – Tooltip with jQuery Button

The final jQuery functionality is a dialog box to show the liability release for the university. JQuery is not the only functionality that was implemented in this application that will need to demonstrated.

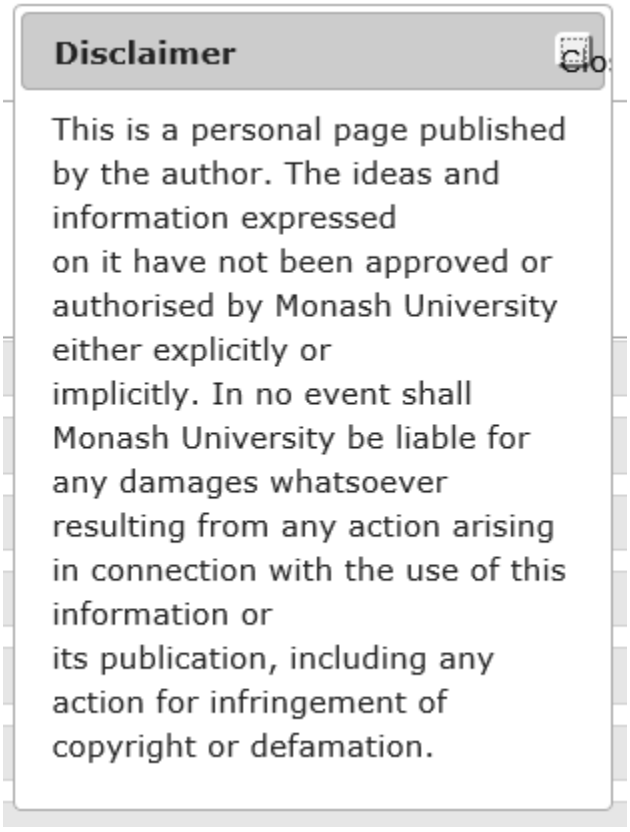


Figure 13 – Liability Release Through a jQuery Dialog Box

The specification for the assignment also indicated that there had to be advanced functionality implemented into the program. The functions that were added to the program were the ability to export to a comma separated value file and an Excel file.

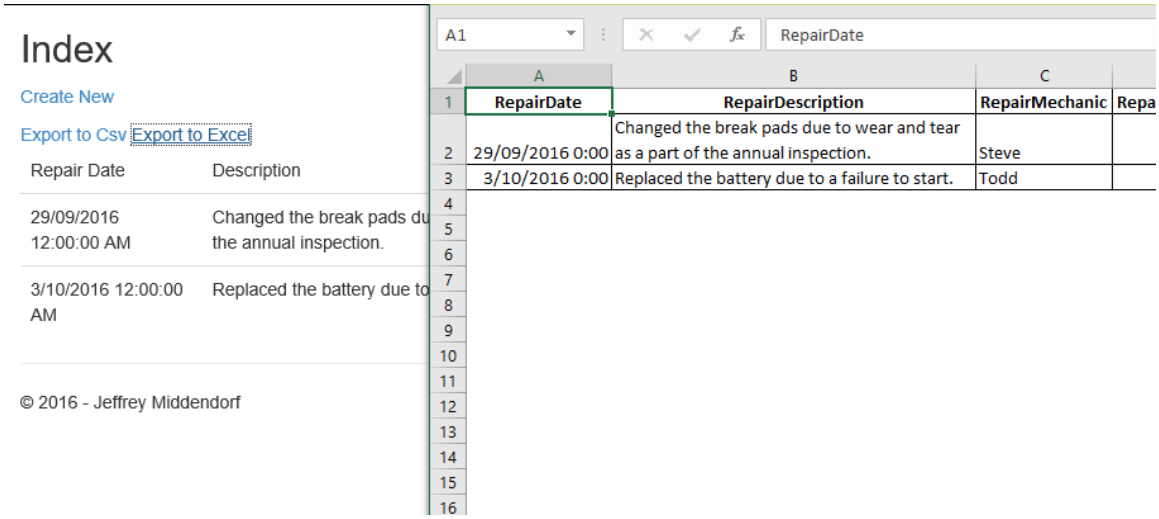


Figure 14 – Export to Excel Functionality

The code was adapted from code provided by Adam Belecki on his blog. A hyperlink is provided at the top of the repair view that produces the respective file. This is accomplished by methods that were added to the repair controller. With these methods added, this should complete the requirements for the assignment specification.

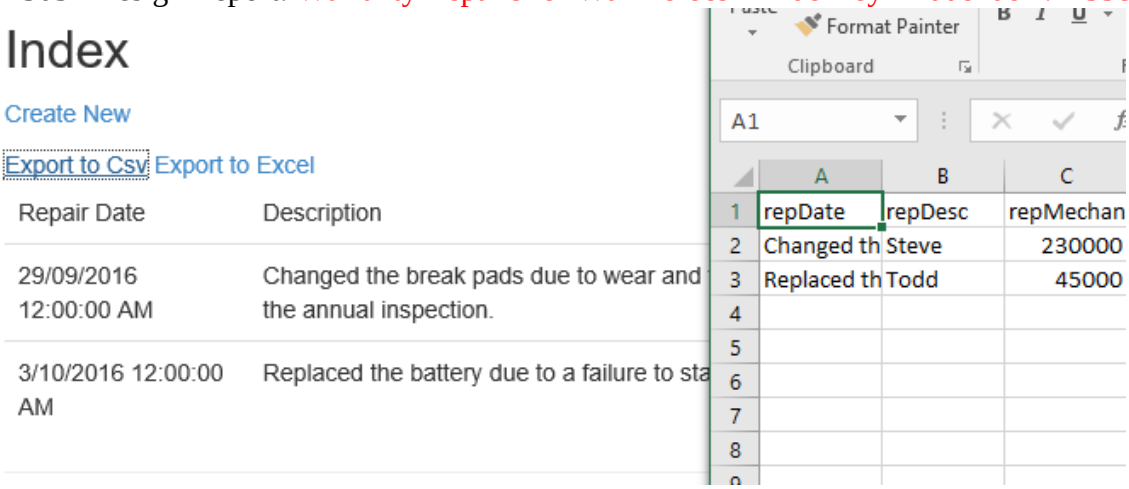


Figure 15 – Export to CSV

The specification required that an MVC application with jQuery extensions and advanced ability. There are three different required jQuery additions with an additional six different methods that were implemented. There was also advanced functionality that was added to complete all of the requirements in the assignment specification. With the functionality complete, the usability analysis portion of the application review can commence.

4. Usability Design Review (based on the Flow Bohl usability categories)

Navigation is the first aspect of the Flow Bohl diagram that will be reviewed for this report. Navigation is how an end-user navigates through the program. The program has decent navigation throughout the program. For example, all index views have links to all of the other views for a particular table that it represents. There are also enough navigation controls to alleviate any navigation issues. The title navigation bar at the top

of the master page is displayed throughout the program. The website should be intuitively familiar to a new user.

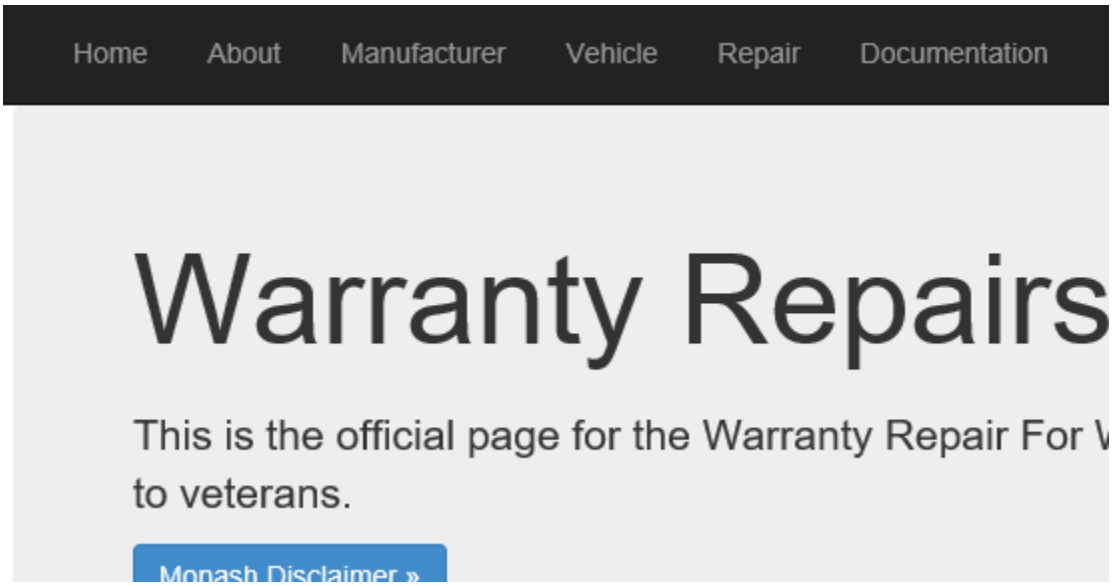


Figure 16 – Navigation Bar

A second consideration of the usability review is familiarity. Familiarity is the websites look and feel when compared to other external websites. If the program significantly different to other websites, new user will be uncomfortable having to learn a new website layout and design. This program seems to have a consistent theme with other websites. Title bar navigation and a central display should be familiar to most users. This is similar to the Flow Bohl aspect known as consistency.

The third aspect of the Flow Bohl analysis is consistency. Consistency is defined as the website having the same look and feel throughout the entire program, internally. If a website has a look and functionality difference, it can create a confusing experience for the end-user. This program consistent throughout the entire program with the

predominately greyscale colour theme. The different views are also similar enough to follow consistency protocols. Consistency is not the last design concept that will need to be considered.

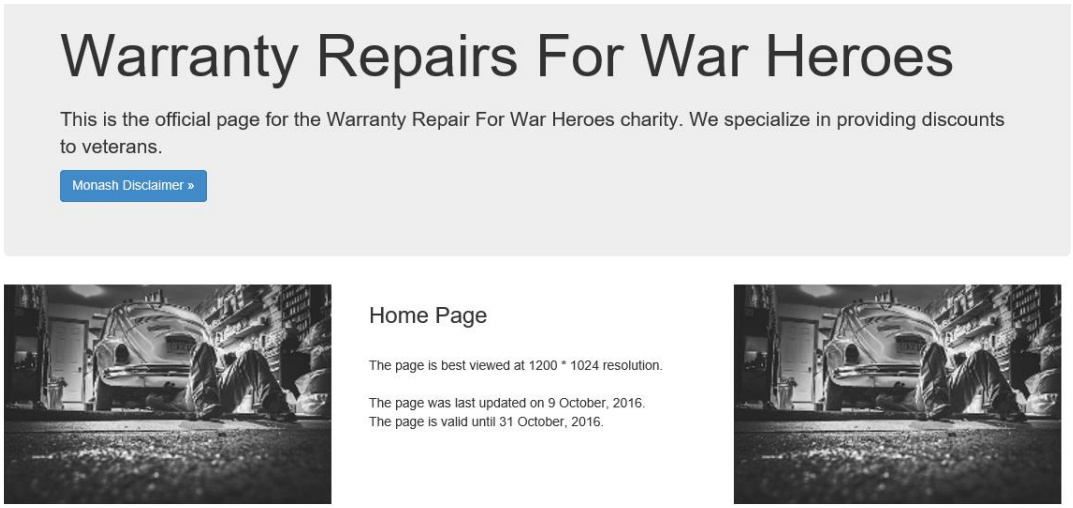


Figure 17 – Consistency Illustration

Error prevention is the fourth aspect of the design usability review. Error prevention refers checking user inputs for errors. This is an important concept for two major reasons. The first reason it is important is because it can create data anomalies if you allow invalid inputs into the database. The second reason that it is important is because you could cause an unhandled exception. This program has some data validation controls that are associated with the views. There are also features like the datepicker that forces a date to be entered. Once the inputs have been validated feedback will have to be provided.

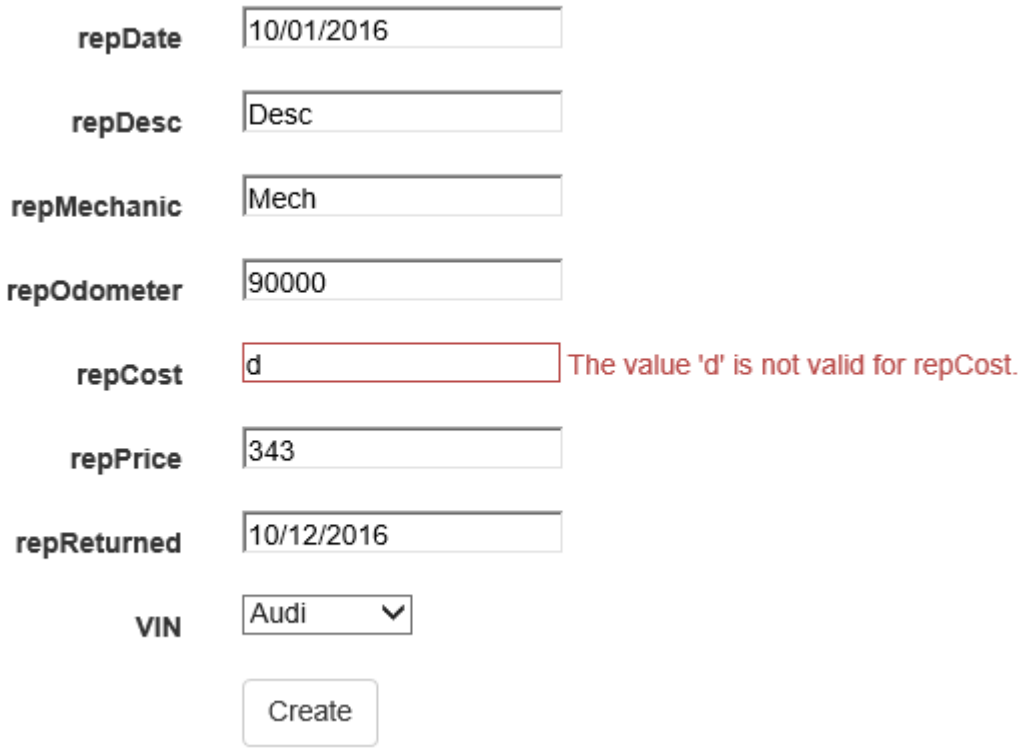


Figure 18 – Error Message

Another aspect of the usability design review is feedback provided by the system. Feedback refers to a form displaying a message when code has been processed when you initiate an action for the form. There are detailed error messages when you input invalid choices. However, there are no messages to indicate that changes have been made to the database. The user can see the changes when they view the index again. It is important to remember that the message will have to be displayed in a clear manner.

Visual clarity is another consideration during the design review process. Visual clarity is the aesthetic presentation of the program. This program uses a greyscale colour scheme that blend well together. It also uses a light red hue for the error messages to that

does not seem to clash with the darker colouring. There is one more aspect of the review that will have to be analysed during this review.

The last aspect that will have to be reviewed during usability is flexibility and efficiency. Flexibility and efficiency refers to the how flexible and efficient the website is to use and update. The code is efficient and flexible for the purposes of a high distinction task. The checklist that follows illustrates the functionalities that were implemented to increase both flexibility and efficiency.

5. Checklist of site functionality

	TICK if complete
1. Master Page	
Design	X
2. Home Page	
Design	X
3. Content Pages	
Design	X
Alphabetical list	X
Add Records	X
View Details	X
Edit Records	X
Delete Records	X
Add/Edit Validation	X
Search Function (Custom Template)	X
jQuery Functionality	X
Additional Functionality	
Export to CSV	X
Export to Excel	X

Documentation	
Author/assignment details	X
Audit	
No breaking of copyright	X

6. User stories

Nick Palmetto – 27 Aug., 16

As the warranty data analyst, I want an application that all data to be easily extract so I can perform complex analyses on it. I want to be able to extract to an Excel format for manager and csv for myself.

Steve Bashir – 27 Aug., 16

As an end-user of the Warranty Repairs for War Heros application, I want a way to search the database looking for specific information. I spend a lot of time considering what vehicles we have and how many repair they have. A search box would make it very convenient to use.

Bill O'Brien – 27 Aug., 16

As the inventory manager, I want a system that I can search the inventory that we have on hand to find a specific vehicle. I also want to be able to search in any field because I don't always know what I am looking for.

7. Data dictionary

Database Dictionary

Manufacturer table

Field	DataType	Constraints	Description
ManID	INT	Identity	This is the primary key of the table. This is an auto incrementing field that is automatically generated when a new number is inserted.
ManName	NVARCHAR	50	This is the manufacturer's name.
contactFName	NVARCHAR	50	This our contact's first name at the manufacturer.
contactLName	NVARCHAR	50	This our contact's last name at the manufacturer.
ManStreet	NVARCHAR	50	This is the manufacturer's street address.
ManSuburb	NVARCHAR	50	This is the manufacturer's suburb.
ManState	NVARCHAR	50	This is the manufacturer's state.
ManPost	NVARCHAR	50	This is the manufacturer's post code.
ManPhone	NVARCHAR	50	This is the manufacturer's phone number.
ManEmail	NVARCHAR	50	This is the manufacturer's email.

Vehicle Table

Field	DataType	Constraints	Description

VIN	NVARCHAR	20	This is the primary key of the table. This is a vehicle identification that is assigned by the OEM.
vehMake	NVARCHAR	25	This is the make of the particular vehicle.
vehModel	NVARCHAR	25	This is the model of a particular vehicle.
vehYear	CHAR	4	This is the year of the particular vehicle.
vehTrim	NVARCHAR	50	This is the trim package that is offered in a particular vehicle.
vehCost	MONEY	Float	This is the cost basis for the vehicle.
vehPrice	MONEY	Float	This is the sale price for the vehicle.
vehQuantity	INT	Integer	This is the quantity of a particular vehicle that we have.
ManID	INT	Integer	This is a foreign key that relates back to the manufacturer OEM that produced the vehicle.

RepairID table

Field	DataType	Constraints	Description
RepairID	INT	Identity	This is the primary key of the table. This is an auto incrementing field that is automatically generated when a new number is inserted.
repDate	DATE	Valid Date	This is the date that the vehicle came into the shop.
repDesc	NVARCHAR	150	This is a description of the issue that was fixed.
repMechanic	NVARCHAR	50	This is the mechanic that is assigned to fix a particular car.
repOdometer	INT	Int	This is the odometer of the vehicle at the time of the repair.
repCost	MONEY	Float	This is the cost that was incurred performing a repair.
repPrice	MONEY	Float	This the price charge for a particular repair.
repReturned	DATE	Valid date	This is the date that a vehicle was returned to the person who brought

			it in.
VIN	NVARCHAR	20	This is the VIN of the vehicle that was repaired. It is a primary key that links to the vehicle table.