**Vehicle Service Management System**

Software Requirement Elicitation, Modelling and Design Document.

Vendor: Caleb Davidson 12110634

Ellis Floriani 12139801

Nathan Downes 12046570

Due Date: Week12 Friday, 10th June 2022 11.45pm

**TABLE OF CONTENTS**

[1 SYSTEM REQUIREMENTS 2](#_Toc105769854)

[1.1 Functional requirements 2](#_Toc105769855)

[1.2 Non-Functional requirements 2](#_Toc105769856)

[2 SYSTEM ARCHITECTURE 3](#_Toc105769857)

[2.1 Architectural Design 3](#_Toc105769858)

[2.2 Class Diagram 4](#_Toc105769859)

[3 DATA DESIGN 5](#_Toc105769860)

[4 INTERFACE DESIGN 8](#_Toc105769861)

[5 TEST PLAN 9](#_Toc105769862)

# SYSTEM REQUIREMENTS

## Functional requirements

The functional requirements for the Vehicle Service Management System (VSMS) are to operate a detailed database. This database can add and store service bookings, vehicle and customer information as well as updating the database information. This is all accessible through java GUI (graphical user interface) which features saving, deleting, searching and changing information via the GUI. This information/data include many variables across the three sets of information Vehicle, Customer and services which all house unique data that are essential for the program. This program will have also installed on the GUI the ability to display any searched information in a display text box showing any information related to whatever data is searched on the GUI.

## Non-Functional requirements

The Vehicle Service Management System (VSMS) non-functional requirements are accessibility, security, and reliability. The accessibility will focus on where the program can be used and managed from within the client’s business. This will be done to give the client’s workforce access to the database across the business so that they can see what servicing customers are booked for, and to know which vehicle being serviced belongs to which customer. The security for this program will focus on prevent leaks of information and protection of data. This will be done through isolating the database to only be accessed via the program and making inaccessible outside the client’s devices. The reliability will focus on the program consistency and reduction of possible misinformation. This will be done through making accidents on the program near impossible and having the program built to work consistently via have reduced complexity.

# SYSTEM ARCHITECTURE

## Architectural Design

Graphical user interface, application

Description automatically generated

## Class Diagram

Diagram

Description automatically generated

# DATA DESIGN

**3.1 Data tables**

|  |
| --- |
| **Customer** |
| CustomerID INT (9) NOT NULL Auto Incremental Primary Key |
| FirstName VARCHAR(30) NOT NULL |
| LastName VARCHAR(30) NOT NULL |
| Address VARCHAR(50) NOT NULL |
| Phone VARCHAR(30) NOT NULL |

# 

|  |
| --- |
| **ServiceBooking** |
| BookingNumber INT (9) NOT NULL Auto Incremental Primary Key |
| Rego VARCHAR(30) NOT NULL |
| Price INT (9) NOT NULL |
| ServiceDescription VARCHAR(50) NOT NULL |
| ServiceDate DATE NOT NULL |

|  |
| --- |
| **Vehicle** |
| CustomerID INT (9) NOT NULL Auto Incremental Primary Key |
| Rego VARCHAR(30) NOT NULL |
| Make VARCHAR(30) NOT NULL |
| Model VARCHAR(50) NOT NULL |
| ManufactureYear DATE NOT NULL |
| Odometer INT (9) NOT NULL |

**3.2 SQL script for database**

Create database CarServiceDB;

Use CarServiceDB;

-------------------------------------------------------------

CREATE TABLE Customers

(

CUSTOMERID INT NOT NULL AUTO\_INCREMENT,

FIRSTNAME VARCHAR (20) NOT NULL,

LASTNAME VARCHAR (20) NOT NULL,

ADDRESS VARCHAR (20) NOT NULL,

PHONE VARCHAR (20) NOT NULL,

PRIMARY KEY (CUSTOMERID)

);

INSERT INTO CUSTOMERS (FIRSTNAME,LASTNAME, ADDRESS,PHONE) VALUES

('Tony','Blair','110 Cresent Road','0402839123'),

('Michael','Li','42 Arnest Street','0401777222'),

('John','Smith','3 BlaBlaRoad','0445155666');

CREATE TABLE Vehicles

(

VEHICLEID INT NOT NULL AUTO\_INCREMENT,

REGISTRATION VARCHAR(20) NOT NULL,

MAKE VARCHAR(20) NOT NULL,

MODEL VARCHAR(20) NOT NULL,

YEAR YEAR NOT NULL,

KILOMETERS INT NOT NULL,

CUSTOMERID INT NOT NULL,

CONSTRAINT VEHICLEID PRIMARY KEY (VEHICLEID,REGISTRATION) ,

FOREIGN KEY (CUSTOMERID) REFERENCES Customers(CUSTOMERID)

);

INSERT INTO Vehicles(REGISTRATION,MAKE,MODEL,YEAR,KILOMETERS,CUSTOMERID) VALUES

('SUN007','Toyota','Camry',2015,60000,1),

('680TER','Subaru','WRX',2003,174000,2),

('160AP4','Nissan','Skyline R33 GTS-T',1995,140000,3),

('138AZT','Holden','Commodore',2005,215000,2),

('990LKM','Volks Wagon','Golf R',2019,42000,3);

CREATE TABLE Services

(

SERVICENUMBER INT NOT NULL AUTO\_INCREMENT,

DESCRIPTION VARCHAR(1000) NOT NULL,

DATE DATE NOT NULL,

PRICE INT NOT NULL,

REGISTRATION VARCHAR(20) NOT NULL,

VEHICLEID INT NOT NULL,

PRIMARY KEY (SERVICENUMBER ),

FOREIGN KEY (VEHICLEID) REFERENCES Vehicles(VEHICLEID)

);

INSERT INTO Services (DESCRIPTION, DATE, PRICE, REGISTRATION,VEHICLEID) VALUES

('Engine rebuild', '2022/6/23', 6000, '160AP4',"3"),

('Oil Change', '2022/9/6', 20, 'SUN007',"1"),

('Head Gasket replacement', '2022/6/11', 400, '680TER',"2"),

('Re-tune', '2022/6/17', 900, '990LKM',"5"),

('Roll Cage Install', '2022/7/22', 200, '138AZT',"4");

# INTERFACE DESIGN

Graphical user interface

Description automatically generated

4:1 Screenshot of GUI

In 4:1 Screenshot of GUI you can see all of the details of the interface including all the buttons View Selected Cust Vehicles, Search Customer, Add New Customer, Search Vehicle and Add Vehicle. Search Customer & Vehicle are both used for searching through the databases information about both Customers & vehicles that have been already stored within the database. Add New Customer & Vehicle are used to add new data via filling the TextFields seen in the 4:1 Screenshot and adding them to the database. View Selected Cust Vehicles displays all the data within the database in a listed format in the tables that you can see are empty in the 4:1 Screenshot.

# 5 TEST PLAN

**Use case:** Adding Customer Information

**Actor:** Staff

**Inputs:** Customer Information

**Outputs:** Customer information added to database

Description: A Customer enters and asks if they could comeback and a later day to service their vehicle and asks for a time and date to specify. The staff then ask for details of the Customer so that that it can be added to database for them to contact the Customer.

Graphical user interface

Description automatically generated

5:1 Screenshot of Error for blank TextField

The Staff then see they accidently forgot to include the customer’s last name. They then correct this and add it.

Graphical user interface

Description automatically generated

5:2 Screenshot successful information being added to database

The Staff then see in the table the added customer details on the right of the of screen. Showing the that they are in in the database

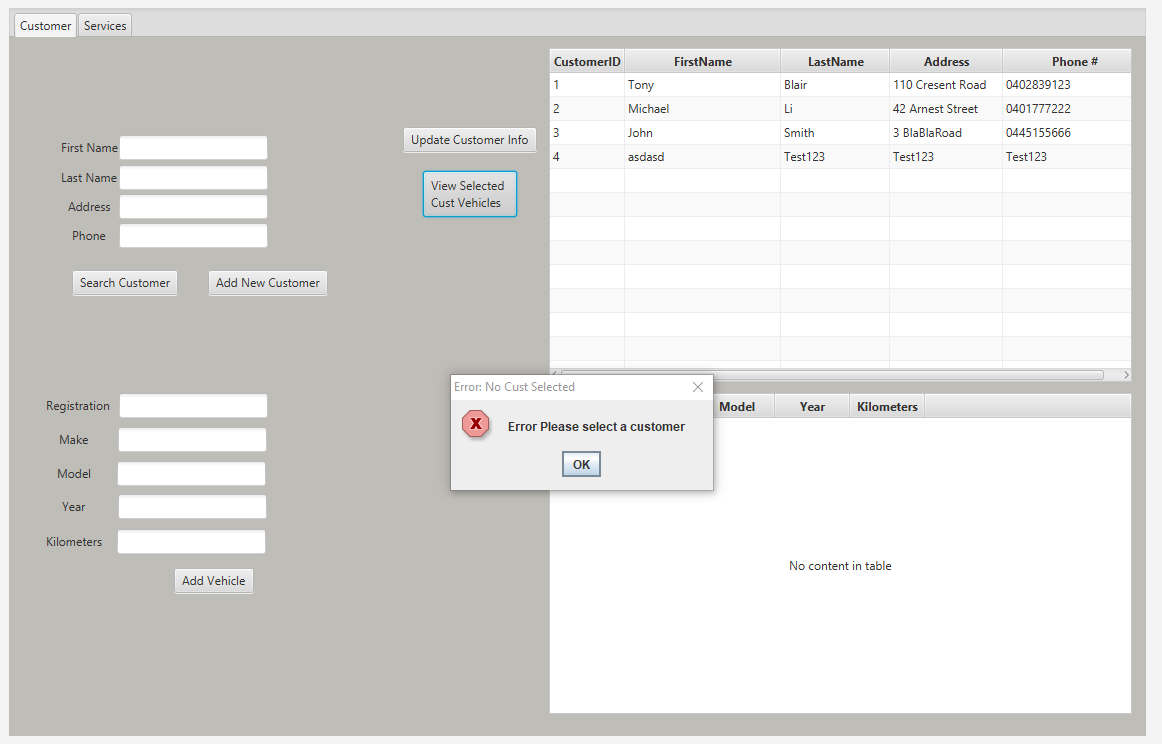
Graphical user interface, application

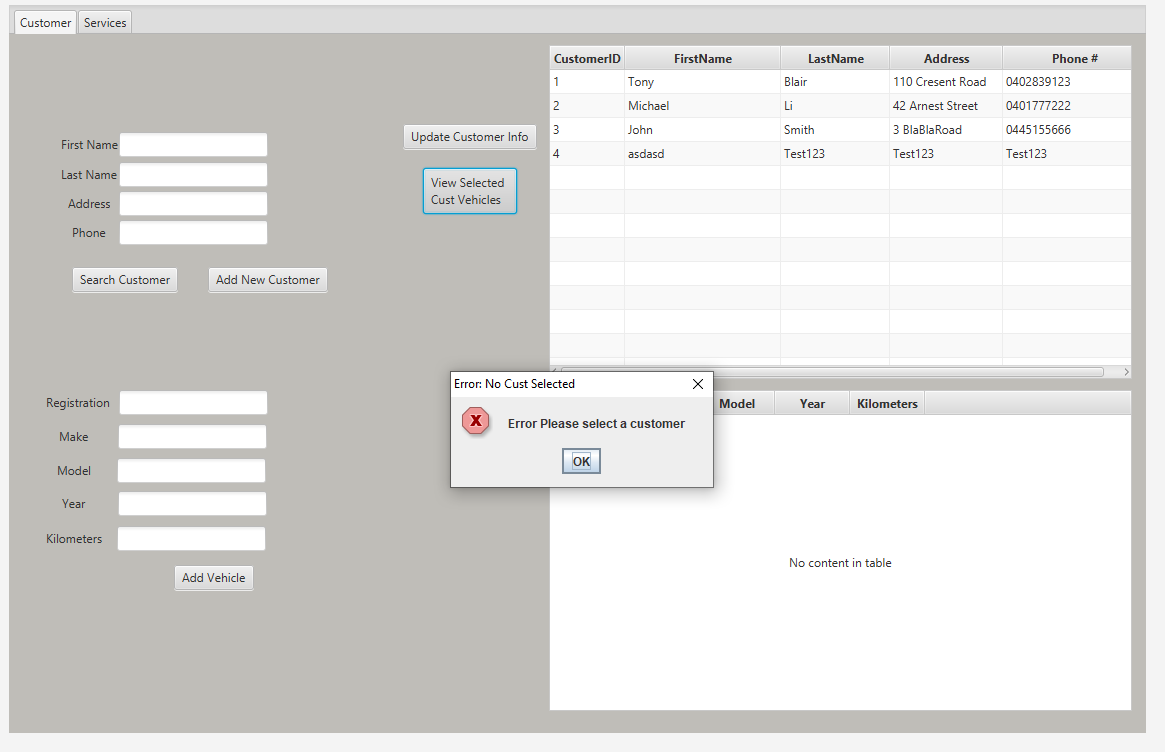
Description automatically generated

5:3 Screenshot of customer details being shown in the table

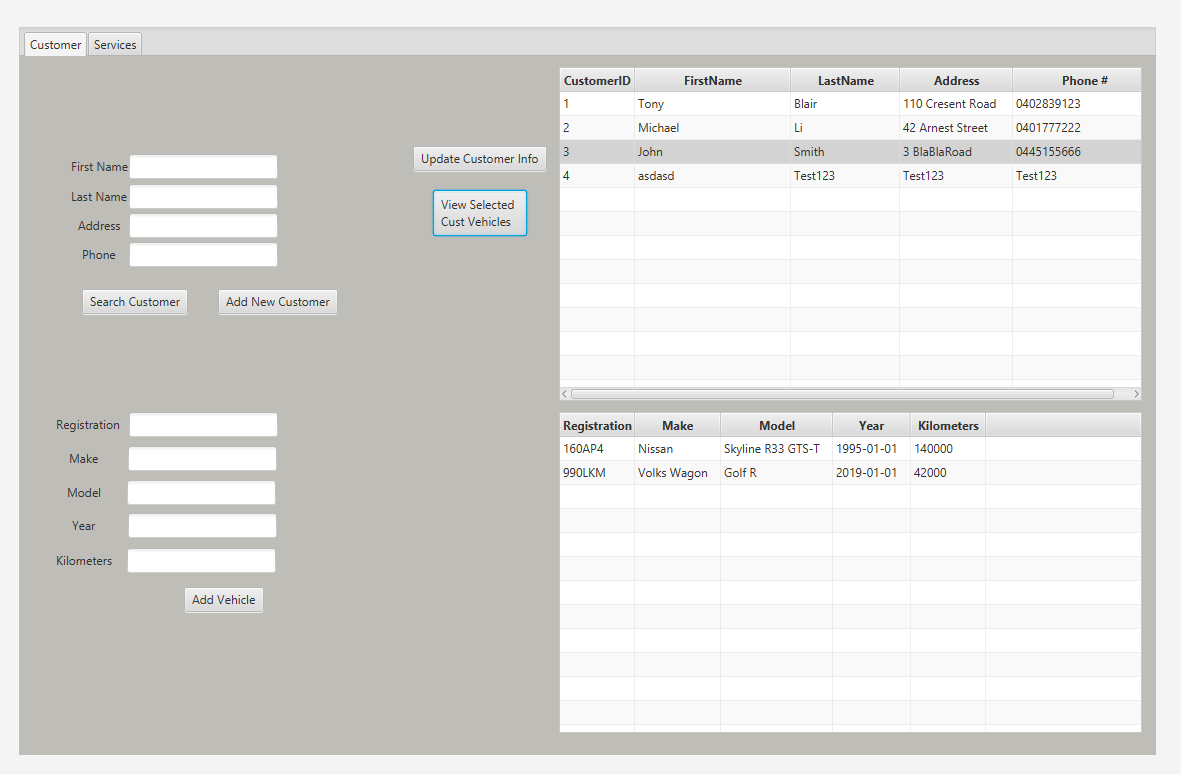
**Exception:**

Error prompt stating. Error please enter all fields.

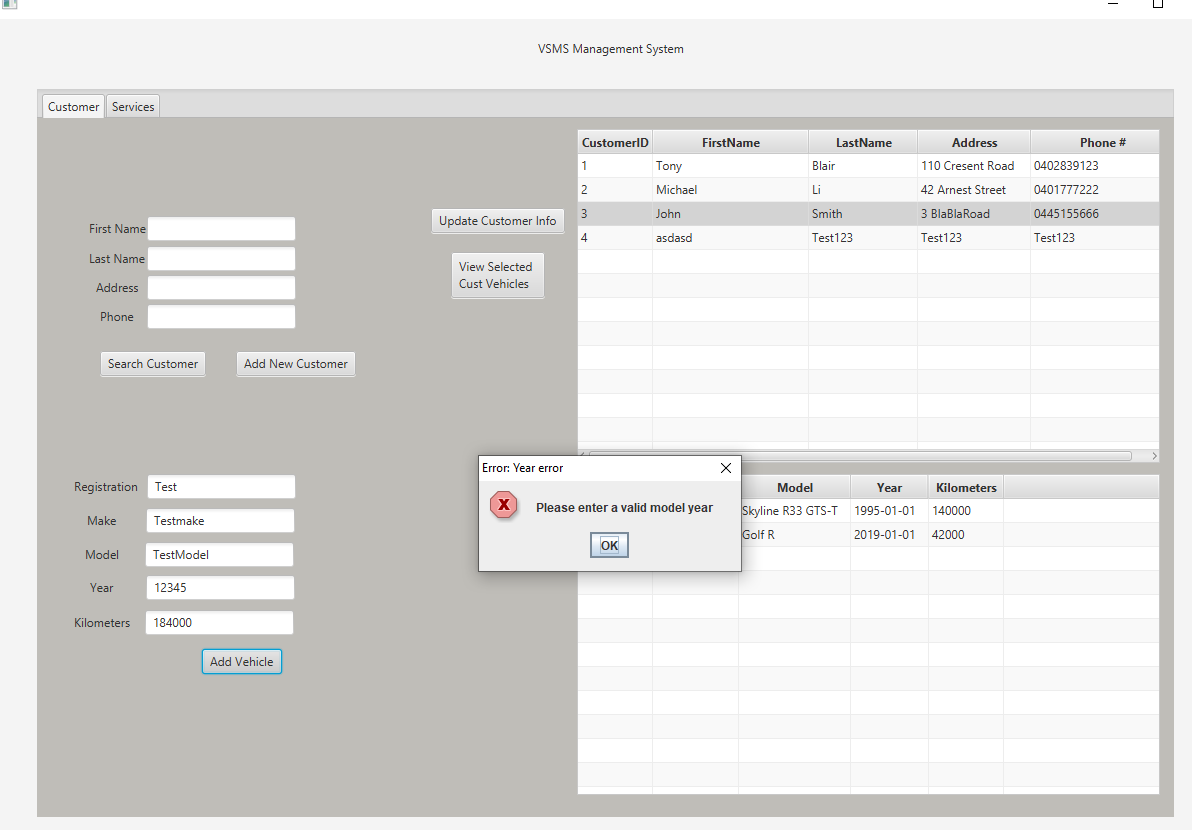




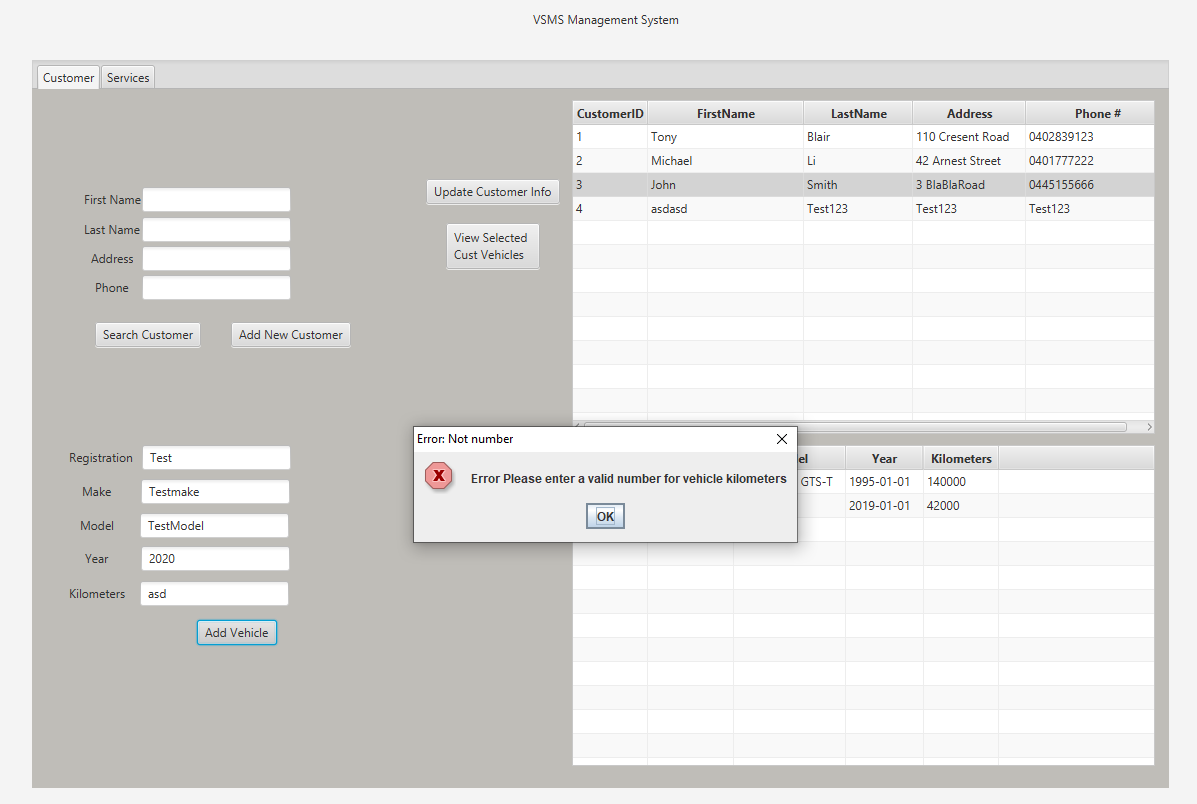
5:4 Screen shot of error when you try to view a customers vehicles without a customer being selected in the table



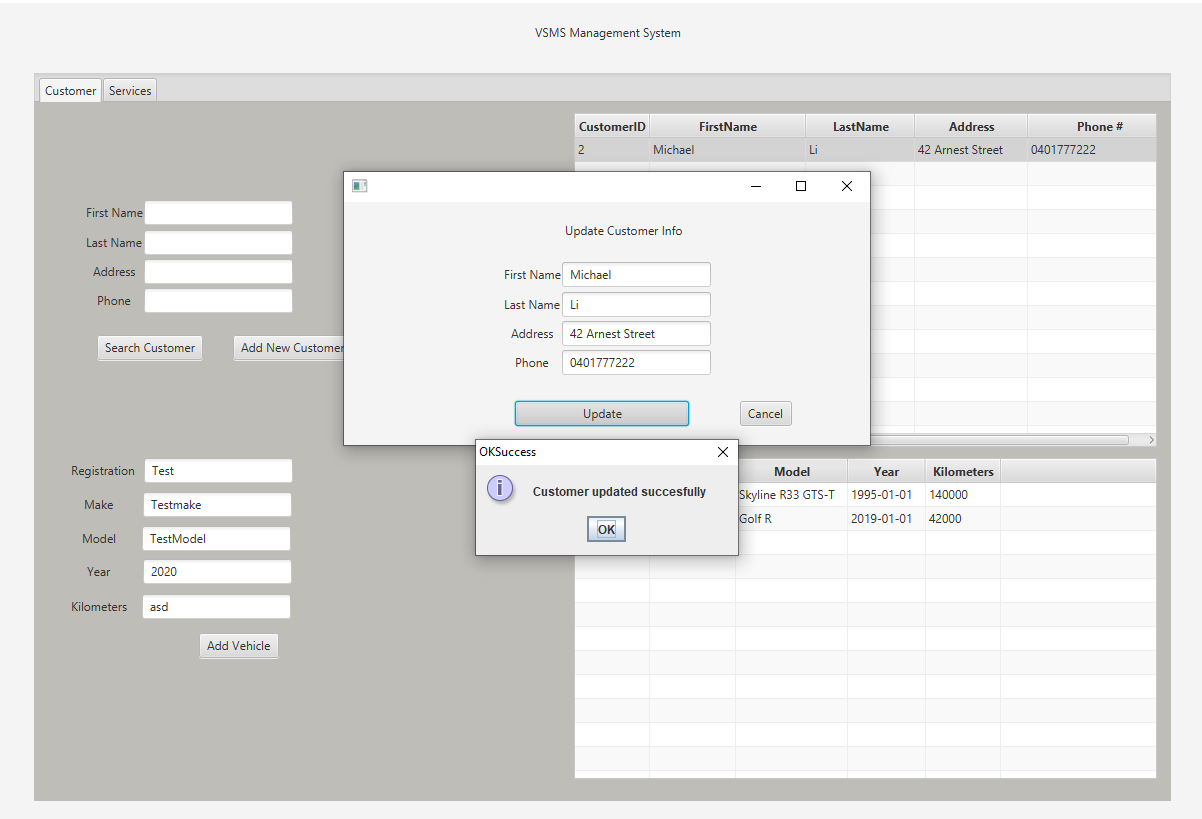
5:5 Showing vehicles attached to a customer



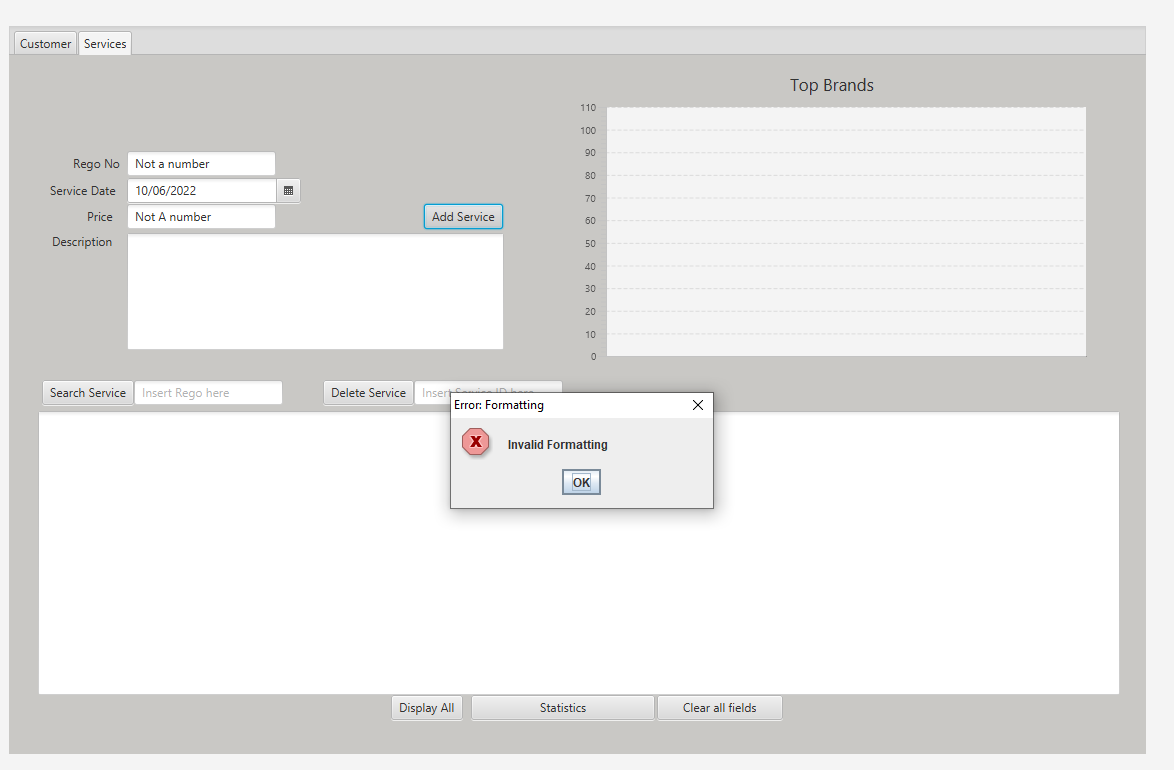
5:6 error if invalid date entered



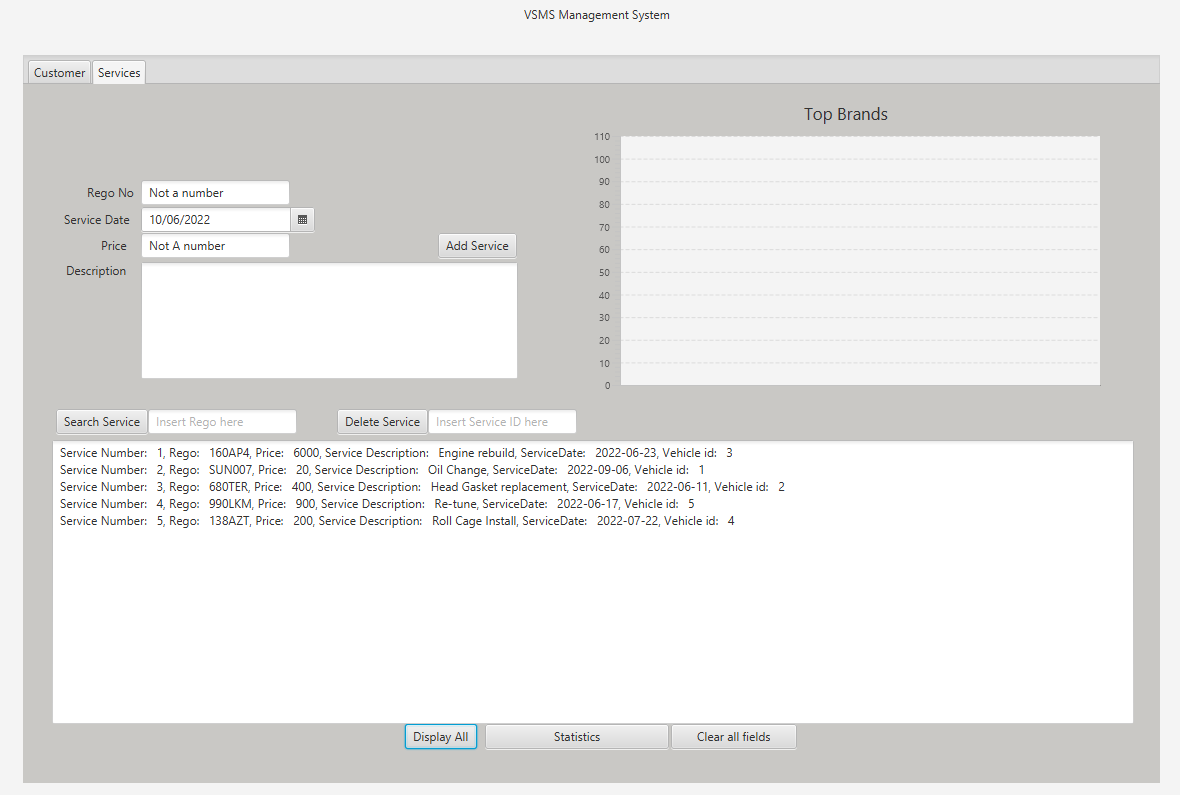
5:7 Error if invalid kilometers entered



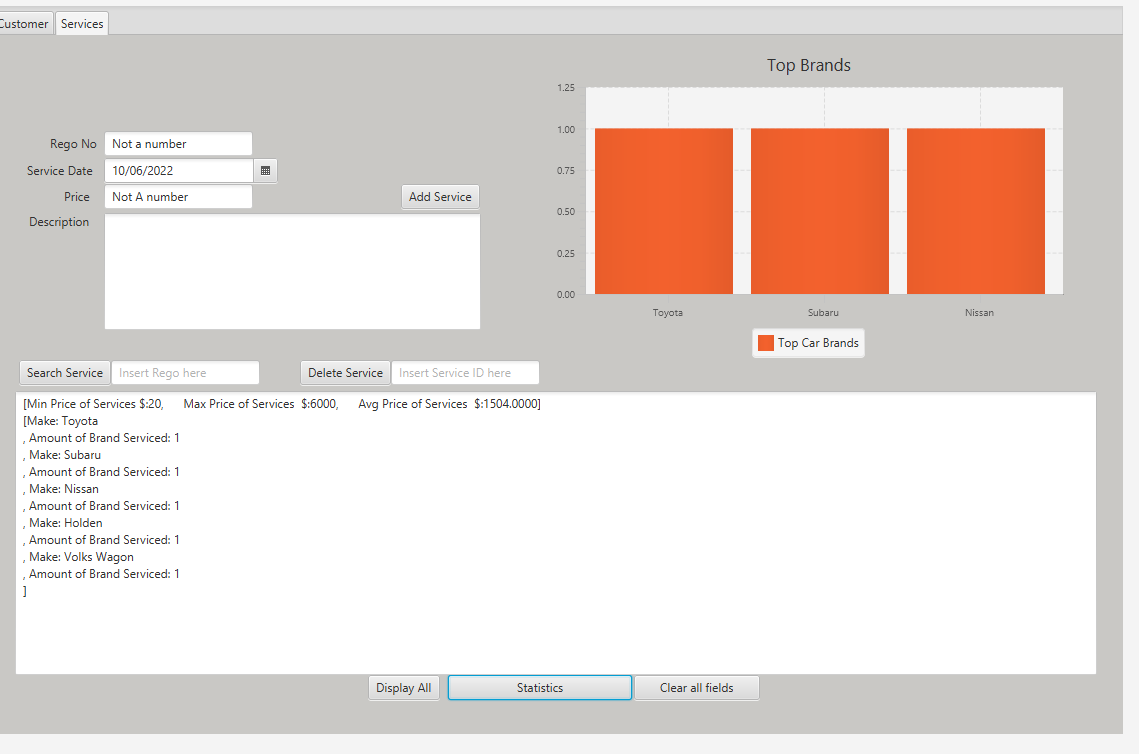
5:8 successful update of customer info



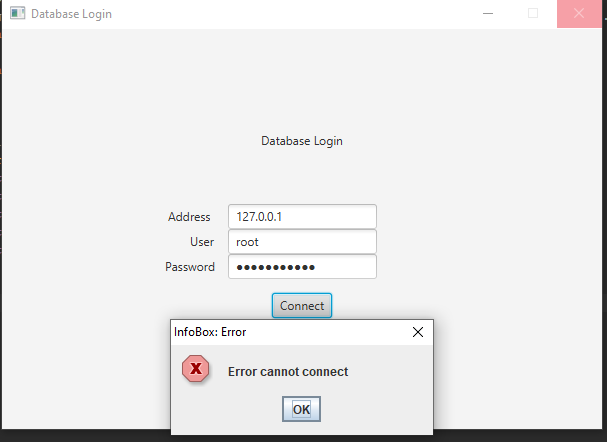
5:9 wrong information entered



5:10 Show all services



5:11 show statistics



5:12 error if cannot connect to database