Software Requirements Specification

for

Automotive Expert System

Prepared by:

Swapnaneel Nandy (IIT2014111)

Amit Vijay (IIT2014110)

Utkarsh Srivastava (IIt2014507)

Mohammad Abdullah ()

Shivam Beri ()

Date: - 18/11/2016

Table of Contents

Table of Contents **1**
1. Introduction **2**
1.1 Purpose 2
1.2 Document Conventions 2
1.4 Product Scope 2
1.5 References 2
2. Overall Description **2**
2.1 Product Perspective 2
2.2 Product Functions 3
2.3 User Classes and Characteristics 3
2.4 Operating Environment 3
2.5 Design and Implementation Constraints 3
3. External Interface Requirements **3**
3.1 User Interfaces 3
3.2 Hardware Interfaces 3
3.3 Software Interfaces 3
4. Functional Requirements **4**
4.1 Taking responses of the user 4
4.2 Based on updated knowledge base infer the solution 4
4.3 Inferred Solution is printed on the screen 4

5. Other Nonfunctional Requirements **5**
5.1 Performance Requirements 5
5.2 Portalability Requirements 5
5.3 Maintainbility Requirements 5
5.4 Security Requirements ..... **Error! Bookmark not defined.**
6. Other Requirements ..... Error! Bookmark not defined.
Appendix A: Glossary **5**
Appendix B: Analysis Models **6**

1. Introduction

1.1 Purpose

This SRS gives a description of the Automotive Expert System. It is an expert system that can be used to repair automobiles or cars. It gives suggestions regarding fixes and remedies for various common issues and problems of a car or automobile in general.

1.2 Document Conventions

In this document we have used bold words to highlight important points. Glossary of important terms has been supplied at the end. Relevant diagrams and charts have also been included in the appendix.

1.3 Project Scope

The Automotive Expert System is a rule-based expert system that gives expert advice to fix general issues and problems related to a vehicle. An expert system is a computer program that uses artificial intelligence (AI) technologies to simulate the judgment and behavior of a human or an organization that has expert knowledge and experience in a particular field. Here we have implemented a rule-based expert system which has a knowledge-base of facts and rules. The solution of a problem is inferred from these facts and rules based on the input of the user.

1.4 References

1. Artificial Intelligence: A Modern Approach Textbook by Peter Norvig and Stuart J. Russell.

2. www.clipsrules.net.

3. www.aa1car.com/

4. www.carcare.org/

5. www.autobytel.com/

6. autorepair.about.com

2. Overall Description

2.1 Project Perspective

Our project implements a simple rule based expert system. An expert system is an example of knowledge-based system which has expert knowledge of a specific domain. It consists of two modules namely the knowledge base and the inference engine. The knowledge base consists of facts pertaining to the concerned domain. The inference engine is an automated reasoning system

that evaluates the current state of the knowledge-base, applies relevant rules, and then asserts new knowledge into the knowledge base. Using this updated knowledge base the system can infer relevant solutions for the problems of the user and give valuable advice for concerned issues. A rule based expert system uses rules of the form IF-THEN-ELSE to construct the knowledge base. The rules and their related Decision Trees have been appended in the appendix.

2.2 Product Functions

The Automotive Expert System is a virtual mechanic which suggests solutions for various issues related to cars and automobile in general. It asks a series of question to the user in the interview phase to give a solution at the end. It accounts for various general and some domain-specific quirks to devise the solution from its knowledge base.

2.3 User Classes and Characteristics

Our project can be used by car owners, drivers, mechanics and automobile engineers. It does not require any specific expertise of the user rather than the use of a general personal computer.

2.4 Operating Environment

The project can work on any general computer with Windows, Mac or Linux operating systems. The only requirement is that the CLIPS inference engine should be installed in the system.

2.5 Design and Implementation Constraints

The only implementation constraint is that the system used by the user or developer must be capable of running CLIPS inference engine which in turn needs a C compiler to run.

3. External Interface Requirements

3.1 User Interfaces

Presently the project has been implemented using CLIPS. Thus it can be run from the CLIPS console or a batch file. We intend on integrating our present system with JAVA Swing using the CLIPSJNI library to create an interactive software in the future.

3.2 Hardware Interfaces

No specific hardware is required to run our project.

3.3 Software Interfaces

The main software used to develop this project is the CLIPS tool for building expert systems. CLIPS stands for "*C Language Integrated Production System*". Thus it is a system used to develop expert system based on production rules. As the name suggests, it is developed using C programming language. Thus a C compiler is required to run CLIPS program.

4. Functional Requirements

The primary functional requirements of the project have been listed below:-

4.1 Taking responses of the user

4.1.1 Description and Priority

The responses of the user relating to a number of well devised questions are taken to comprehend the issue and come up with a solution.

4.1.2 Stimulus/Response Sequences

Here, the user gives various answers in the form of binary responses or multi-responses. These responses update the knowledgebase of the system.

4.2 Based on updated knowledge base infer the solution

4.2.1 Description and Priority

The recorded responses of the user updates the knowledge base. From this knowledge base CLIPS uses forward-chaining to infer the solution. Every time a rule is fired, it searches through the present knowledge base. If the conditional fact is present in the database it will assert a new fact in the database.

4.2.2 Stimulus/Response Sequences

There is no external stimulus at this step as the inference is an internal function of the CLIPS framework.

4.3 Inferred Solution is printed on the screen

4.2.1 Description and Priority

The solution inferred in the previous step is printed on the terminal of CLIPS.

4.2.2 Stimulus/Response Sequences

Here the response is the final result.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The software should have acceptable performance for an expert system. For this software the final result is not known in the beginning. Thus we cannot use backward chaining in this case which is generally faster than forward chaining. Thus we have to use a fast framework for forward chaining which in our case we have selected to be CLIPS.

5.2 Portability Requirements

It is of utmost importance that the software should be portable. Thus we have used CLIPS which is a C based forward chaining Expert System framework. As C is very common and has compiler for every OS, our software is very portable.

5.3 Maintainability Requirements

Maintainability is a major focus point for an expert system as new facts, rules and domain knowledge can be added any time. Thus our software is developed in a properly modular fashion while keeping maintainability in mind.

5.4 Security Requirements

The software is locally developed in CLIPS which is an open source tool without the use of any third-party software. Thus the software is secure and trustworthy for the user.

Appendix A: Glossary

Expert System: A piece of software which uses databases of expert knowledge to offer advice or make decisions in such areas as medical diagnosis.

Knowledge Base: A database consisting of knowledge pertaining to a specific domain in the form of rules and facts.

Rules: rules are conditional statements to represent knowledge written in the form of IF-THEN-ELSE statements

Facts: Facts are already known results in a knowledge base.

Inference Engine: It is a tool from artificial intelligence used to infer results from a knowledge base.

Forward Chaining: Forward chaining is one of the two main methods of reasoning when using an inference engine and can be described logically as repeated application of modus ponens.

Backward Chaining: Backward chaining (or backward reasoning) is an inference method that can be as working backward from the goal.

Appendix B: Analysis Models

The initial question asked in the expert system is the type of problem with answers Engine, Tyre, Headlight, Steering and Suspension. Based on the choice we have five decision trees as follows.

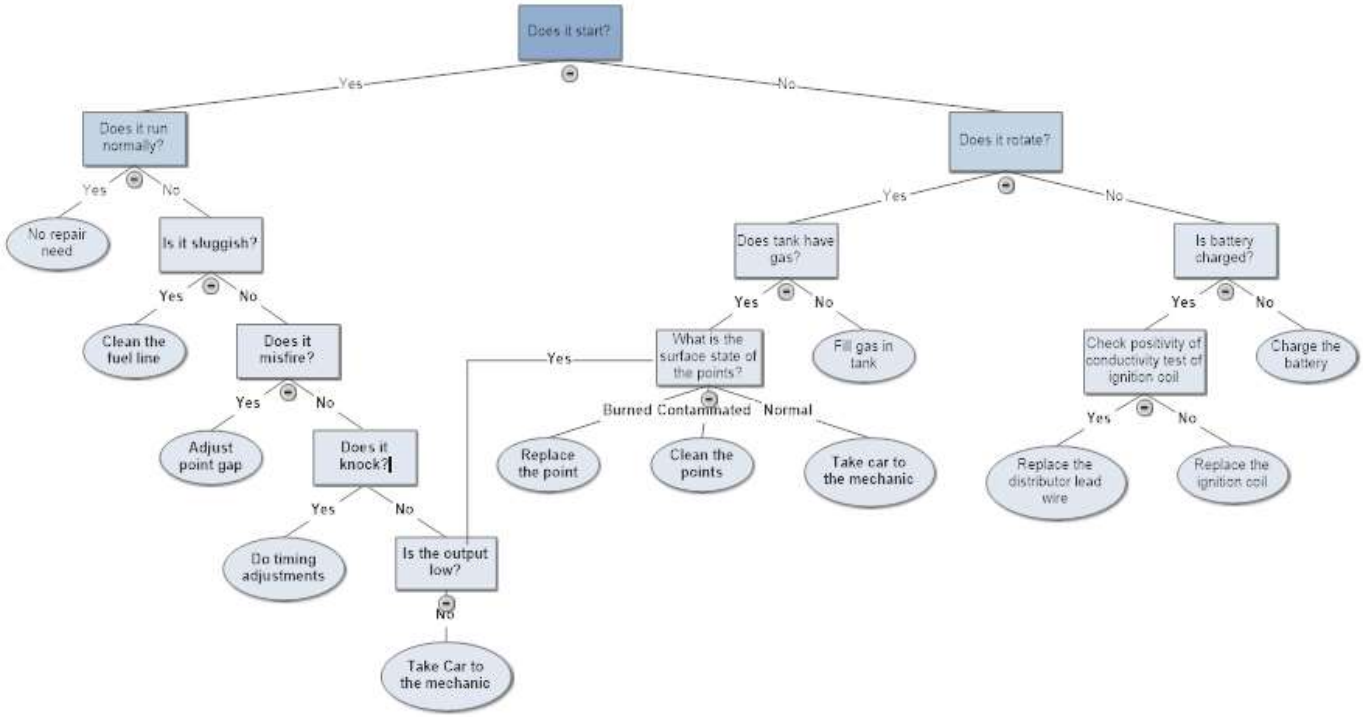

Figure 1: Engine Problems

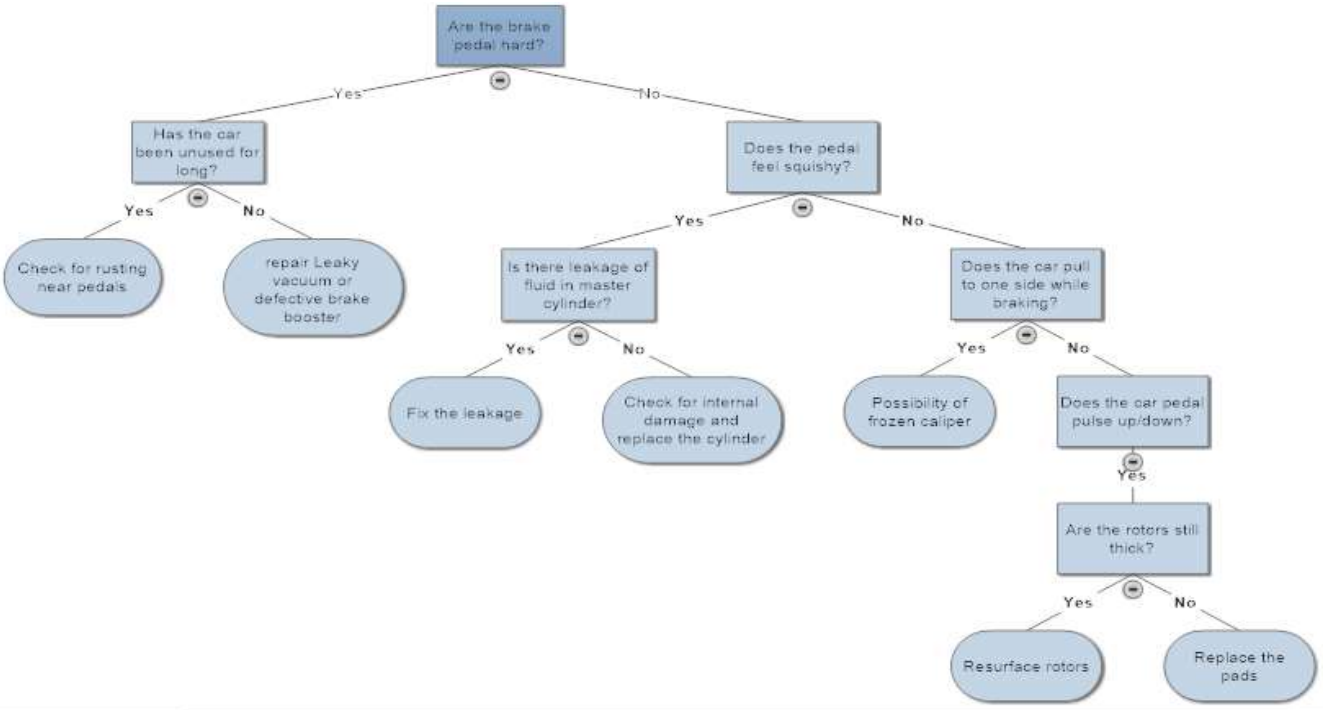

Figure 2: Brake Problems

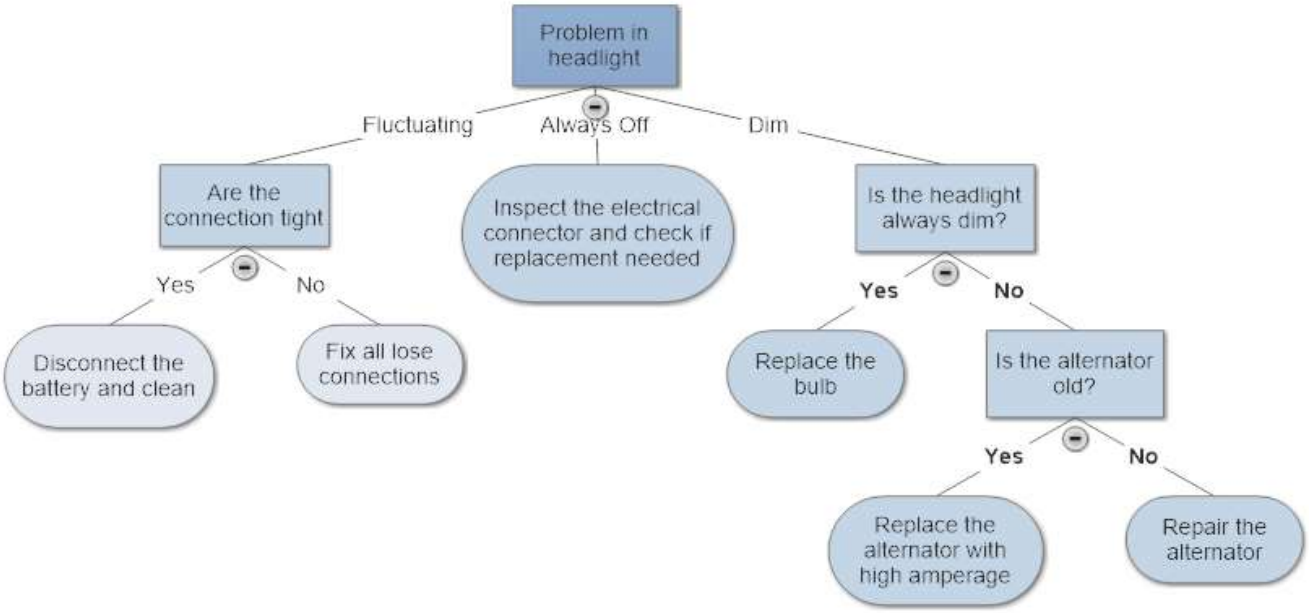

Figure 3: Headlight Problems

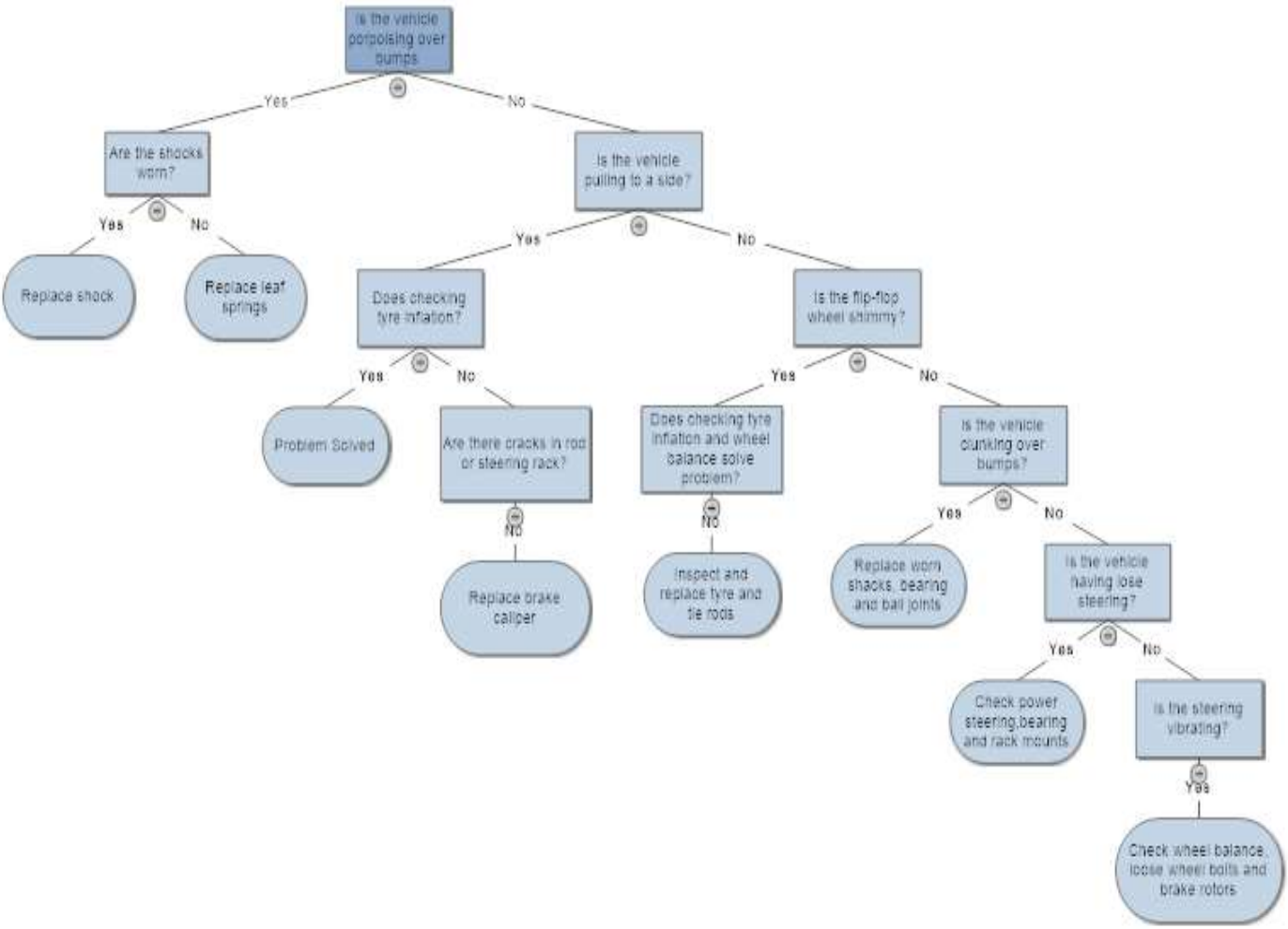

Figure 4: Suspension and Steering Problems

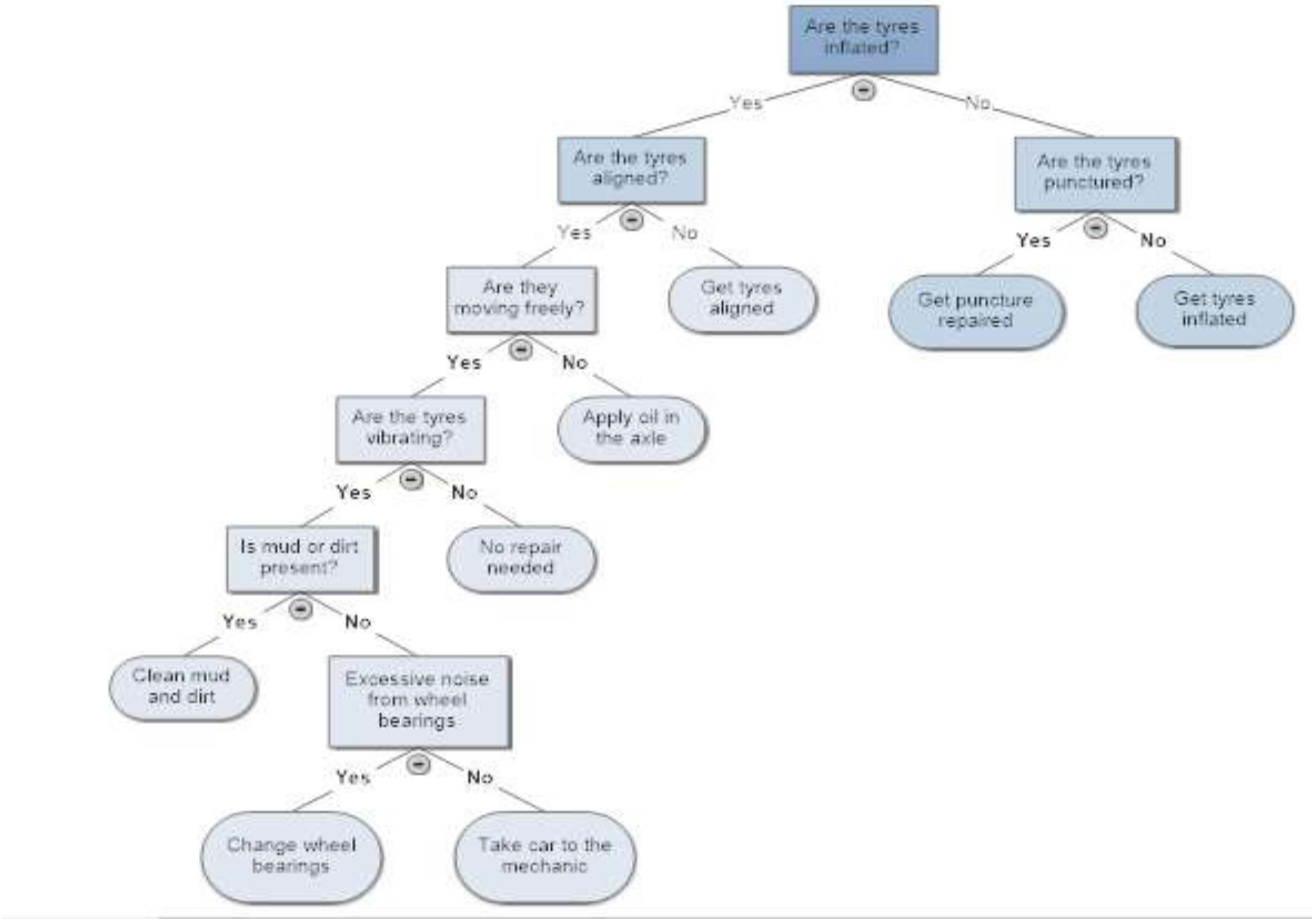

Figure 5: Tyre Problems