

# EXPERT SYSTEM FOR SWEAR ANALYSIS

## SOFTWARE REQUIREMENTS SPECIFICATION

Team members:

MATHI JESSICA (2843649)

SAHITHI VARMA RAGHAVARAJU(2837094)

NITHIN ARYAN AGRISHETTY(2837033)

HALAA PRANAVI VANGARA(2837402)

### 1.0 Introduction

Expert systems are computer programs that emulate the decision-making ability of a human expert. They are designed to solve complex problems by reasoning through a set of rules defined by an expert in the field.

The three main components of an expert system are the knowledge base, the inference engine, and the user interface. The knowledge base stores all the relevant information and rules that the expert system needs to solve a problem. The inference engine is responsible for reasoning through the rules in the knowledge base to find a solution. The user interface is how the user interacts with input information and receives results from the expert system.

Expert systems have been used successfully in various domains, such as medical diagnosis, fault diagnosis, interpreting measurement data, and configuring systems. They are beneficial because they can improve the productivity of human experts, apply expertise uniformly, and save time of human experts.

When designing an expert system, there are some factors to consider. The first is the type of reasoning that will be used. There are two main types of reason: rule-based reasoning and case-based reasoning. Rule-based reasoning relies on a set of rules that are defined by the expert. Case-based reasoning relies on previous cases that have been solved by the expert.

The next factor to consider is the knowledge representation. There are a variety of ways to represent knowledge, such as frames, rules, or ontologies. The expert must choose the representation that best suits the problem at hand.

Finally, the expert must decide on the user interface. The user interface must be designed in a way that is easy to use and understand. It should be able to input information and receive results from the expert system clearly and concisely.

### **1.1 Goals and objectives**

1. To develop a system to analyze students' performance in an examination.
2. To provide a graphical interface for the user to interact with the system.
3. To provide a way for the user to input the data into the system.
4. To provide a way for the system to output the analysis results.

### **1.2 Statement of scope**

The Expert system for swear analysis is a software application designed to help users analyze their performance in either a chatbot or Tkinter. This system will take in input from the user in the form of text, then analyze the text to see if there are any swear words present. If swear words are present, the system will then provide feedback to the user via either a chatbot or Tkinter message. This feedback will help the user to understand their performance and to improve their language usage.

The major inputs to the Expert system for swear analysis are the pieces of text that the user wishes to analyze. These texts can be inputted into the system via a chatbot or Tkinter interface. Once the text has been inputted, the system will then analyze the text and provide feedback to the user.

The Expert system for swear analysis is designed to help users analyze their performance in either a chatbot or Tkinter. This system will take in input from the user in the form of text and then analyze the text to see if any swear words are present. If swear terms are present, the system will provide the user with feedback via either a chatbot or Tkinter message. This feedback will help the user to understand their performance and to improve their language usage.

The outputs of the Expert system for swear analysis are the chatbot or Tkinter messages generated based on the inputted text. These messages will provide feedback to the user in the form of either a chatbot or Tkinter message. This feedback will help the user to understand their performance and to improve their language usage.

### **1.3 Software context**

The project is about developing an expert system for swear analysis. The system will either be in the form of a chatbot or a graphical user interface (GUI) built with Tkinter. The project is intended to help users understand their performance by providing questions and grades.

The project has three main modules: student information form, verbal, and quantitative. In the student information form, users will input their details. The oral and quantitative modules will test the user's understanding of the material. The project is still in development, and more modules may be added in the future.

The project is intended to help users improve their performance. The project is still in development, and more features may be added in the future.

## **1.4 Major constraints**

First, the software must be compatible with the existing chatbot or Tkinter application. Second, the software will need to be able to handle a large number of users simultaneously. Third, the software will need to identify swear words and provide appropriate responses accurately. Finally, the software must be tested to ensure it functions correctly.

## **2.0 Usage scenario**

The first scenario is as follows:

A student wants to use the software to find out how well they are doing in school. They will input their name, and the software will output their current grade.

The second scenario is as follows:

A student wants to use the software to determine their grades if they got a certain score on a test. They will input their current score, and the software will output their predicted angle.

The third scenario is as follows:

A student wants to use the software to determine how many points they need to get a certain grade in a class. They will input the quality they want, and the software will output the number of points they need.

## **2.1 User profiles**

There are three user categories in this project, which are:

1. Student Information Form
2. Verbal
3. Quantitative

The Student Information Form is for students who want to know their performance in the verbal and quantitative sections. The Verbal section is for the students who wish to improve their performance in the verbal area. The Quantitative section is for students who want to improve their performance in the quantitative section.

## **2.2 Use-cases**

The first use case is for the student information form. In this form, the student will enter personal information like name, age, gender, etc. They will also enter their academic information like the school's name, year of study, etc. This information will be used to generate the student's report card.

The second use case is for the verbal section. In this section, the student will be asked a series of questions. Based on their answers, the software will generate a score. This score will be used to determine the student's grade in the verbal section.

The third use case is for the quantitative section. In this section, the student will be asked a series of questions. Based on their answers, the software will generate a score. This score will be used to determine the student's grade in the quantitative section.

### **2.3 Special usage considerations**

Since this is a student information system, some special requirements may be associated with using the software. For example, the software may need to be able to interface with a student information database. Additionally, the software may need to be able to generate reports or export data to other software applications.

Another special requirement that may be necessary is the ability to track changes made to the data. This is important if errors are made or changes need to be reverted. Additionally, it may be necessary to have a user login system in place so that only authorized users can access the data.

Overall, the requirements for this software will depend on the specific needs of the users. However, some general requirements that may be necessary include the ability to interface with other software applications, the ability to generate reports, and the ability to track changes made to the data.

## **3.0 Data Model and Description**

1. Student Information Form
2. Verbal
3. Quantitative

### **3.1 Data Description**

The Student Information Form module is used to store the details of the students. The details include the student's name, roll number, contact details, etc.. This module also allows the user.

The verbal module is used to test the vocal ability of the students. In this module, the students are provided with a set of questions to be answered. Based on the answers provided by the students, they are given a grade.

The Quantitative module is used to test the quantitative ability of the students. In this module, the students are provided with a set of questions to be answered. Based on the answers provided by the students, they are given a grade.

### **3.1.1 Data objects**

- Student – name, roll number, contact details, etc.
- Question – the question to be answered, the options to choose from, the correct answer, etc.
- Grade – the grade given to the student based on their answers.

### **3.1.2 Relationships**

Relationships among data objects are described using an Entity-Relationship Diagram (ERD) form. No attempt is made to provide detail at this stage.

### **3.1.3 Complete data model**

An ERD for the software is developed.

### **3.1.4 Data dictionary**

A reference to the data dictionary is provided. The dictionary is maintained in electronic form.

## **4.0 Functional Model and Description**

The project's main function is to create a GUI application using Tkinter. The project has three main modules: Student Information Form, Verbal, and Quantitative. The Student Information Form module is used to enter the details of the students. The Verbal module is used to provide some questions to the students and to provide grades based on their answers to the questions. The Quantitative module gives some questions to the students and provides rates based on their responses to the questions

### **4.1. Description of Major Functions**

1. Student Information Form: This module is responsible for storing and retrieving student information.
2. Verbal: This module is responsible for administering verbal questions to the students and calculating their scores.
3. Quantitative: This module is responsible for administering quantitative questions to the students and calculating their scores.

#### **4.1.1**

Student Information Form: This module requires the ability to store and retrieve student information.

#### **4.1.2**

Verbal: This module requires the ability to administer verbal questions to the students and to calculate their scores.

### **4.1.3**

Quantitative: This module requires the ability to administer quantitative questions to the students and to calculate their scores.

### **4.1.4**

Reporting: This module requires the ability to generate reports based on student information and scores.

## **4.2 Software Interface Description**

The software interface of the above project can be divided into two parts. The first part is the graphical user interface (GUI) , which is used by the user to input the students' details and view the students' performance. The second part is the command line interface (CLI), which the developers use to view the code and to make changes to the code.

The project's graphical user interface (GUI) is developed using the Tkinter module. The Tkinter module is a standard Python module that is used to create GUI applications. The Tkinter module provides a powerful object-oriented interface to the Tk GUI toolkit. The Tkinter module exposes the following geometry manager classes: pack, grid, and place. The group () method organizes widgets in blocks before placing them in the parent widget. The grid() method contains devices in a table-like structure in the parent widget. The place () form organizes widgets by placing them in a specific position in the parent widget.

### **4.2.1 External machine interfaces**

The above project describes interfaces to other machines in a chatbot or using Tkinter. The chatbot is used to interact with the user, and the Tkinter is used to create a GUI application. The chatbot is used to ask the user questions, and the Tkinter provides a graphical interface for the user to input their answers. The chatbot then analyzes the responses and provides feedback to the user. The Tkinter interface is used to display the feedback to the user.

### **4.2.2 External system interfaces**

- The Tkinter module is used to create GUI applications.
- The Student Information Form is used to input student information.
- The Verbal module provides questions and grades based on answers to those questions.
- The Quantitative module provides questions and grades based on answers to those questions.

### **4.2.3 Human Interface**

The above project involves the development of a chatbot or an Expert system for swear analysis. The Expert system will be used to analyze students' performance by providing some questions and grades based on their answers to the particular questions.

The chatbot or the Expert system will have a human interface which the students will use to interact with the system. The human interface will be designed in such a way that it will be easy for to use and understand by students. The human interface will also be designed in such a way that it will be able to provide the required information to the students effectively and efficiently.

### **5.0 Restrictions, Limitations, and Constraints**

- 1) The first issue is related to the Student Information Form. The form requires the student's name, date of birth, contact information, etc. But not all the students need to have all this information. So, there should be some provision to skip some fields if the student does not have that information.
- 2) The second issue is related to the verbal module. In this module, the questions asked should be such that they test the student's understanding of the concepts and not just their memory. Otherwise, the results obtained will not be accurate.
- 3)The third issue is related to the quantitative module. In this module, the questions asked should be such that they test the student's understanding of the concepts and not just their computational skills. Otherwise, the results obtained will not be accurate.
- 4) The fourth issue is related to the implementation of the software. The software should be such that the students can easily use it. Otherwise, they will not be able to use it properly, and the results will not be accurate.