MEDICAL REPORT ANALYSIS

Medical Report Analysis
Software Requirements Specification

Version 1.0

Revision History

Date	Version	Description	Author
28/Dec/2023	1.0	Software Requirements Specification Document Initial Release.	Gokul.S Mohan Vamsi V P Sumaant K.B

Table of Contents

Software Requirements Specification

1. Introduction:

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete **MEDICAL REPORT ANALYSIS** by defining the problem statement in detail. The detailed requirements of the **MEDICAL REPORT ANALYSIS** are provided in this document.

1.1 Purpose:

The purpose of the document is to collect and analyze all that have come up to define the test report, its requirements with respect to patients. Also, we shall predict and sort out the problems of patients that the project will be explained about, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

This document outlines the Software Requirements Specification (SRS) for a Medical Report Analysis system specifically designed for processing and analyzing test results in a medical test center environment. The system aims to simplify and streamline result interpretation for lab technicians, enhance patient understanding, and provide valuable insights for healthcare professionals. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

1.2 Scope

This system will:

- Accept test data input from lab technicians through a user-friendly interface.
- Validate and clean the input data for accuracy and consistency.
- Analyse test values based on predefined reference ranges and established medical guidelines.
- Categorize results as normal, mildly increased, mildly decreased, significantly increased and significantly decreased based on defined thresholds.
- Generate clear and informative graphical visualizations of results for each patient.
- Provide concise interpretation and insights associated with categorized results.
- Ensure data security and user access control for patient confidentiality.
- Integrate seamlessly with existing data management systems if required.

1.3 Definitions, Acronyms and Abbreviations:

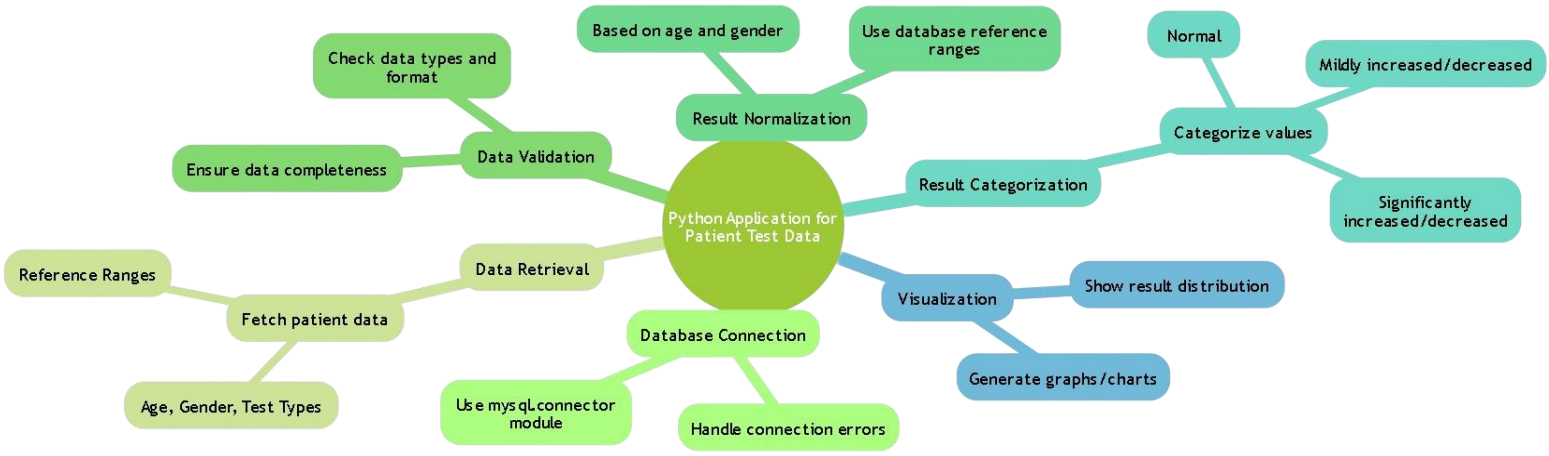
- **Medical Test**: A laboratory analysis of various test to assess various health parameters.
- **Reference Range**: The expected range of values for a specific test based on age, gender, and other factors.
- **Categorization**: Grouping of test results into predefined categories based on their deviation from reference ranges.
- **Visualization**: Graphical representation of results using Bar chart.
- **Insights**: Additional information or observations derived from the analysis of results.
- **Lab Technician**: Qualified personnel responsible for collecting, processing, and analyzing test samples.
- **Medical Test Center**: A healthcare facility offering various testing services to patients.

1.4 References

- Clinical Laboratory Standards Institute (CLSI) guidelines for specific medical tests.
- Relevant medical textbooks and research publications.
- Industry standards for software development and security.

1.5 Overview

This system will serve as a valuable tool for medical test centers, enabling efficient medical test analysis, accurate result interpretation, and improved patient care.



Overview of the proposed system

2. Overall Description:

2.1 Data Input and Management:

- The system shall provide a user-friendly interface for lab technicians to input patient demographics, test types, and measured values.
- Input fields shall be clearly labeled and validated for data type and format.
- The system shall automatically check for duplicate entries and missing data.
- Data cleaning procedures shall be implemented to handle outliers and inconsistencies.
- An audit trail shall record all data input and modification activities.

2.2 Result Analysis and Categorization:

- The system shall utilize predefined reference ranges and established medical guidelines for various blood tests.
- Test values shall be analyzed and categorized into normal, abnormal (mild, moderate, severe), or critical based on defined thresholds.
- The system shall be able to handle and categorize values falling outside reference ranges.
- Configurable parameters shall allow adjustments to reference ranges and thresholds if needed.

2.3 Visualization and Interpretation:

- The system shall generate clear and informative graphs for each test, depicting the distribution of categorized results.
- Graph types (e.g., bar charts, line graphs) shall be customizable based on user preferences.
- Concise interpretations shall be provided for each categorized result, explaining its clinical significance.
- The system may offer additional insights derived from the analysis of results, such as trends or patterns.

3. Specific Requirements:

3.1 Functional requirements:

- **Database Connection**: The Python application will connect to the MySQL database using the mysql.connector module and handle potential connection errors.
- **Data Retrieval**: The application will fetch patient test data, age, gender, test types, and corresponding reference ranges from the database using SQL queries.
- **Data Validation**: The system will validate retrieved data for proper data types, format, and completeness.
- **Result Normalization**: Test values will be normalized based on age and gender using reference ranges stored in the database.
- **Result Categorization**: Normalized values will be categorized into normal, mildly increased/decreased, significantly increased/decreased based on predefined thresholds.
- **Visualization**: The system will generate Bar graph showing the distribution of patient results across categories for each test.

3.2 Non - Functional Requirements:

- **Performance**: The system should perform database queries and data analysis within acceptable response times.
- **Security**: Ensure data encryption, access controls, and audit trails for both the database and Python application.
- **Usability**: The Python application interface should be user-friendly for lab technicians with varying technical expertise.
- **Maintainability**: Use clear and well-organized code in Python and design the database structure for maintainability.

3.3 System Constraints:

- The available data sources and reference ranges are specific to the lab's instruments and procedures.
- The system may encounter limitations specific to Python and MySQL.

4. Acceptance Criteria:

- The system successfully connects to the MySQL database and retrieves patient data.
- Data validation ensures retrieved data is complete and consistent.
- Test values are normalized using appropriate reference ranges.
- Results are accurately categorized into predefined categories.
- The system generates clear and informative visualizations.
- Performance meets defined response time requirements.
- The system is user-friendly for lab technicians.
- Error handling function effectively been used.
- The system is maintainable and can be easily updated.

5. Future Enhancements:

- Integrate directly with lab instruments for automatic data import.
- Develop a web application interface for wider access to results and insights.
- Expand functionality to analyze additional types of medical data.