

RV University

**Diabetes Prediction System**

**Software Requirements Specification**

**Version 1.0**

**11-08-2004**

**RV Restricted**

|  |
| --- |
| RV University. This document is not to be copied or distributed without the express written consent of RV University. No part of this document may be used for purposes other than those intended by RV University.    **Based on template Version: PRHB\_SRS\_T\_V\_1.0** |

Document Identification

|  |  |
| --- | --- |
| Project Name | Diabetes Prediction System |
| Document Home | <> |
| Version No | 1.0 |
| Status | Revised |
| Audience | RV internal |

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Description | Document owner / Modified by |
| 1.0 | 07-06-2023 | Document created | Akshay Kumar B P |

**Software Requirement Specification**

TABLE OF CONTENTS

13. Table Of Figures....................................................................................................................... 4

1. OBJECTIVES AND SCOPE................................................................................................. 5

2. INTENDED AUDIENCE...................................................................................................... 5

3. DEFINITIONS, ACRONYMS AND ABBREVIATIONS................................................... 5

4. REFERENCES....................................................................................................................... 6

5. REQUIREMENTS OVERVIEW........................................................................................... 6

5.1 Project Perspective 6

5.2 Project functions 7

5.3 Operating Environment 9

5.4 Customer enforced technology choices 9

5.5 User Documentation 9

5.6 Assumptions ,dependencies and external risks 9

6. EXTERNAL INTERFACE REQUIREMENTS................................................................ 10

6.1 User Interfaces 10

6.2 Hardware interfaces 10

6.3 Software interfaces 11

6.4 Communication interfaces 11

7. FUNCTIONAL REQUIREMENTS.................................................................................... 11

7.1 Graphical User Interface 11

7.2 Encoder/Decoder……………………………………………………………….14

7.3 Communication module 19

8. PERFORMANCE REQUIREMENTS............................................................................... 23

9. ACCEPTANCE REQUIREMENTS................................................................................... 23

10. MAINTAINABILITY REQUIREMENTS......................................................................... 23

11. OTHER REQUIREMENTS................................................................................................. 23

11.1 Portability Requirements 23

11.2 Reliability requirements 24

11.3 Scalability requirements 24

12. OPEN ISSUES...................................................................................................................... 24

13. Table Of Figures

Figure 1 Flow Diagram.. 6

Figure 2 PDU Format 7

Figure 3 PDU Fields. 8

Figure 4 Network Elements. 10

Figure 5 Manager-Agent 11

Figure 6 GetRequest Window. 12

Figure 7 SetReuest Window. 13

Figure 8 Response Window. 14

Figure 9 Error Window. 14

Figure 10 GetRequest PDU. 17

Figure 11 TLV Format 17

Figure 12 Socket call for UDP. 23

**1.**OBJECTIVES AND SCOPE

Using ML algorithms and data sets related to glucose levels and blood pressure, we have created a system to predict the presence of diabetes in a patient

**2.**INTENDED AUDIENCE

This document is intended to be reviewed by customer, test plan developers and the code developers.

**3.**DEFINITIONS, ACRONYMS AND ABBREVIATIONS

|  |  |
| --- | --- |
| **ML** | Machine Learning |
| **AI** | Artificial Intelligence |
| **DL** | Deep learning |
| **NB** | Naïve Bayes |

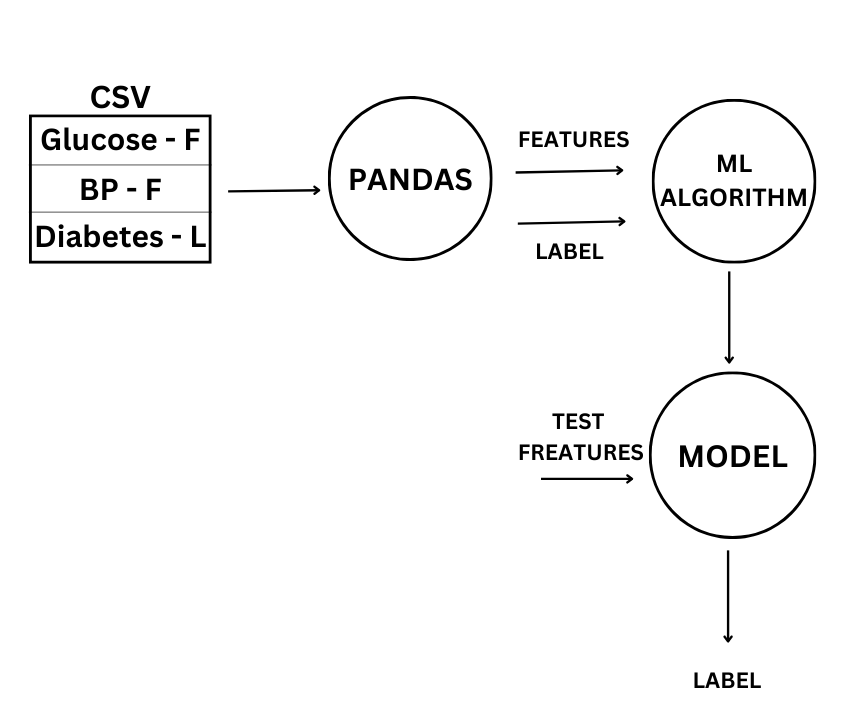
**4.**REFERENCES

Code provided

**5.**REQUIREMENTS OVERVIEW

## **Project Perspective**

This project is to be used to accurately predict whether a person is likely to have diabetes or not based on their glucose and blood pressure level. This is built to be a platform independent program that can be used easily and efficiently by providing the data set that is to be analyzed.



**Project functions**

PDU being the basic unit of transfer for the SNMP protocol implementation, The main functionality of the project involves constructing the PDU. To construct a PDU , Basic Encoding Rules have to be followed, which constitutes as a part, using the ASN.1 notation.

Figure 2 PDU Format

There are five types of PDU that can be transferred from the manager to the agent and vice versa .  
 SNMP is a simple request/response protocol. NMSs can send multiple requests without receiving a response. Five basic SNMP operations are defined which are described below.

Message Types:

Get

Allows the NMS to retrieve an object instance from the agent.

GetNext

Allows the NMS to retrieve the next object instance from a table or list within an agent. In SNMP, when an NMS wants to retrieve all elements of a table from an agent, it initiates a Get operation, followed by a series of GetNext operations.

GetResponse

This give a corresponding response for the get and the getNext.

Set

Allows the NMS to set values for object instances within an agent.

Trap

Used by the agent to asynchronously Inform the NMS of some event. The SNMP trap message is designed to replace the SNMP trap message.

PDU Packet Format

Figure 3 PDU Fields

Brief Description of the PDU Fields

Version

Community

SNMP PDU

PDU Type

Get-request[0]

Get-next-request[1]

Set-request[2]

Get-response[3]

Trap[4]

Request ID

To track a message with the expected response

Error Status

ErrorStatus::=

INTEGER

{

noError(0)

tooBig(1)

noSuchName(2)

badValue(3)

readOnly(4)

genErr(5)

}

Error Index

It is used to provide additional information on the error status.

Name and Value pair

## **Operating Environment**

· Hardware platform Pentium III, 166 MHz Pentium II or higher.

· Operating system Linux or HP-UX

· Software components C, GTK, Minimum of 64 MB of RAM

The software to be written will use the underlying UDP/IP protocol stack and the sockets for its complete implementation.

UDP/IP connectivity to at least one SNMP-capable device (agent)

## **Customer enforced technology choices**

· Operating system to be used should be Linux or Windows.

· Python should be used as a langauge for the implementation of the described product

## **User Documentation**

## Users will be provided with a user manual that will include instructions on how to use the product

## **Assumptions ,dependencies and external risks**

· The data set has been provided and is accurate

· The pattern developed can be used to predict the presence of diabetes

**6.**EXTERNAL INTERFACE REQUIREMENTS

## **User Interfaces**

· A window to accept **user data** from the user

· The user should enter accurate glucose and blood pressure levels

· A warning window will appear if the data seems to out of the normal range

## **Hardware interfaces**

CPU above i5  
 RAM of at least 8GB to 16GB  
 Storage to store the data set

## **Software interfaces**

Data set collection

Warning page

## **Communication interfaces**

Data collection interface

Data processing interface

Prediction interface

**7.**FUNCTIONAL REQUIREMENTS

## **Graphical User Interface**

### **7.1.1** **RFP Id-F1**

### **7.1.2** **Description**

F1.1 window for Prediction results

This user window will accept the

· Glucose

· Blood pressure

**8.**Code

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv("Naive-Bayes-Classifier-Data.csv")

df.head()

x=df.drop('diabetes',axis=1)

y=df['diabetes']

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.25,random\_state=42)

model=GaussianNB()

model.fit(x\_train,y\_train)

y\_pred = model.predict(x\_test)

y\_pred

**9.**MAINTAINABILITY REQUIREMENTS

Maintaining the data model