## Sleep Apnea Detection Using

## Datamining Techniques

#### A PROJECT REPORT

***Submitted by***

**Gokul Reddy.A [REGISTER NO:211417104005]**

**Hari Vikash.A [REGISTER NO:211417104019]**

**Jeevan Kalyan.Ch [REGISTER NO:211417104041]**

***in partial fulfillment for the award of the degree of***

### BACHELOR OF ENGINEERING

**IN**

#### COMPUTER SCIENCE AND ENGINEERING

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

**ANNA UNIVERSITY: CHENNAI 600 025**

**APRIL 2020**

**BONAFIDE CERTIFICATE**

Certified that this project report **“SLEEP APNEA DETECTION USING DATAMINING TECHNIQUES ”** is the bonafide work of **“ A.GOKUL REDDY(2017PECCS355),A.HARIVIKASH(2017PECCS363), JEEVAN KALYAN(2016PECCS379) ”** who carried

out the project work under my supervision.

#### SIGNATURE SIGNATURE

**Dr.S.MURUGAVALLI,M.E.,Ph.D., Mr. M.MOHAN,M.Tech.,(Ph.D.), HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR(GRADE-I),** DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE, NAZARATHPETTAI, NAZARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on 22.09.2020

#### INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors

**Tmt.C.VIJAYARAJESWARI**, **Thiru.C.SAKTHIKUMAR,M.E.,** and

**Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,**

for the support extended throughout the project.

We would like to thank my **Project Guide Mr.M.Mohan,M.Tech.,(Ph.D.)** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

M. ARAVINDH, PL. ARUN PANDI,

M. BHARATHKANNAN.

## ABSTRACT

Obstructive Sleep Apnea (OSA) is a common, but severely under-diagnosed sleep disorder that affects the natural breathing cycle during sleep with periods of reduced respiration or no airflow at all. It is our long-term goal to increase the percentage of diagnosed OSA cases and reduce the time to diagnosis with user friendly and cost-efficient tools for sleep analysis at home. We choose the patient dataset for find the Obstructive Sleep Apnea. We clustering the dataset cluster for apply for KNN algorithm. These ratio can be more or less generalized throughout the industry. The reason of bias towards classification models is that mode analytical problem involves making decision. In our project we get the affected patient list and non-patient list with KNN algorithm. We apply KNN algorithm in blood pressure.

## We get the output in bar chart. Who find affected patient list and non-patient list with KNN algorithm. We achieve an accuracy of 96.6% with a combination of respiration data from blood pressure. These good results are also achieved with the simple KNN technique. Preprocessing information it is nothing it can remove null value. And Fetch the data we take particular information for KNN calculation. In our project we get the exact output who are nearest disease and non-disease patient list with Bar chart output.

## 

### 

### LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **4.** | **SYSTEM DESIGN** |  |
|  | 4.1. ER-Diagram | 25 |
|  | 4.2. Data Flow Diagram of Doctor | 28 |
|  | 4.3. Data Flow Diagram of Patient | 28 |
|  | 4.4. Use Case Diagram | 33 |
|  | 4.5. Class Diagram | 34 |
|  | 4.6 Sequence Diagram | 35 |
|  | 4.7 Timeline Chart | 36 |
| 5. | SYSTEM ARCHITECTURE |  |
|  | 5.1. System Architecture | 38 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **3.** | **SYSTEM ANALYSIS** |  |
|  | 3.1. Minimum Hardware Requirements | 13 |
|  | 3.2. Minimum Software Requirements | 13 |
| **4.** | **SYSTEM DESIGN** |  |
|  | 4.1. Data Dictionary of User | 26 |
|  | 4.2. Data Dictionary of Patient | 26 |
|  | 4.3. Data Dictionary of Doctor | 27 |
| **6.** | **SYSTEM TESTING** |  |
|  | 6.1. Test Case for User Module | 46 |
|  | 6.2. Test Case for Knowledge System | 47 |
|  | 6.3. Test Case for Doctor Module | 47 |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | iv |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 Overview | 2 |
| **2.** | **LITERATURE SURVEY** | 4 |
| **3.** | **SYSTEM ANALYSIS** | 10 |
|  | 3.1 Existing System | 11 |
|  | 3.2 Proposed system | 11 |
|  | 3.3 Requirement Analysis and Specification | 11 |
|  | 3.3.1 Input Requirements | 12 |
|  | 3.3.2 Output Requirements | 12 |
|  | 3.3.3 Functional Requirements | 12 |
|  | 3.4 Feasibility Study | 12 |
|  | 3.4.1 Technical Feasibility | 12 |
|  | 3.4.2 Economic Feasibility | 12 |
|  | 3.5 Hardware Environment | 13 |
|  | 3.6 Software Environment | 13 |
|  | 3.7 Software Specification | 14 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **4.** | **SYSTEM DESIGN** | 24 |
|  | 4.1. ER diagram | 25 |
|  | 4.2 Data dictionary | 25 |
|  | 4.3 Data Flow Diagram | 27 |
|  | 4.4 UML Diagrams | 29 |
| **5.** | **SYSTEM ARCHITECTURE** | 37 |
|  | 5.1 Architecture Overview | 38 |
|  | 5.2 Module Design Specification | 38 |
|  | 5.2.1 Module Explanation | 39 |
|  | 5.3 Program Design Language | 40 |
| **6.** | **TESTING** | 43 |
|  | 6.1 System Testing | 44 |
|  | 6.2 Unit Testing | 45 |
|  | 6.3 Test Cases & Reports / Performance Analysis | 46 |
| **7.** | **CONCLUSION** | 48 |
|  | 7.1 Conclusion and Future Enhancements | 49 |
|  | **APPENDICES** |  |
|  | A.1 Sample Screens | 50 |
|  | A.1 Sample Code | 54 |
|  | **REFERENCES** | 93 |

**CHAPTER 1 INTRODUCTION**

**1.1 OVERVIEW**

Obstructive sleep apnea syndrome is becoming a prevalent disease for both adults and children. It is described as the cessation of breath for at least 10 seconds during sleep. Detecting sleep apnea is considered as a troublesome and time-consuming method, which requires the patients to stay one or more nights in dedicated sleep disorder rooms with sensors physically attached to their body. Undiagnosed thereby untreated sleep apnea patients are under high risk of hypertension, heart attack, traffic accident through fatigue and sleeplessness. In this project, nasal and oral respiratory information is obtained with utilizing thermocouple and oxygen saturation in the blood is obtained with utilizing pulse oximeter. An analog hardware circuit is designed to readout thermocouple and pulse oximeter signals. According to this respiratory and pulse oximetry signals, obstructive sleep apnea is detected in real time with using a software implemented into an ARM based processor. An Android mobile application is developed to record and display the oxygen saturation, heart rate and respiratory signal data during sleep. ARM based processor and mobile application communication is established via Bluetooth interface to reduce cabling on the patient. In summary, a portable, low cost and user friendly device to detect obstructive sleep apnea which is able to share the necessary information to the patients and doctors for the duration of the whole sleep cycle is developed.

The two key contributions of this work are related to systematic comparison of signal combinations and data mining techniques, and to data preprocessing. First, we train our classifiers with only minimal preprocessing of raw-data and still achieve rather good classification results; while the main body of existing works performs feature extraction before training. Second, we conduct an exhaustive, systematic comparison of all possible combinations of signals that are relevant for home monitoring with five data mining techniques. One surprising insight from our evaluation is that using a single signal can lead to equally good or even better classification results than using all signals. Furthermore, we obtain very good results even with the very simple KNN data mining technique. In fact, KNN in many cases outperforms more sophisticated techniques like the SVM and ANN, and achieves more than 90% accuracy, sensitivity and specificity for all signal combinations with high quality data. This particular result stands in contrast to

The current major focus in the research community on complex deep-learning and sophisticated feature extraction techniques. The reminder of this paper is structured as follows: Section II discusses related works. Section III describes the used method and Section IV presents our evaluation results. Section V summarizes the conclusions and future work. One important difference to our work is the emphasis, in related works, on preprocessing the data set, feature selection, and feature extraction. We only normalize the raw data and down-sample it to 1 Hz. The classification performance is rather high for the Kidney disease data set.

KNN: We use as the distance function the City block function d(p,q) = √∑(pi – qi)2

Where pi and qi represents the it attribute of objects p and q, respectively, and n the number of attributes in the data objects. We vary the number of neighbors used to determine the majority class. Ties among classes are resolved by choosing the class with the lowest index, which in our case is the one denoting epochs with normal breathing. Based on our preliminary experiments, we determine not to apply weighing of neighbors based on their distance to the test object All related works summarized in Table 2 conclude that the signal combinations they investigate lead to good classification results such that it is possible to gain reliable predictions of the severity of OSA with one or a few respiratory signals instead of a full blood pressure. We used dataset upload to sql so we have maintained easily and comfortable for users.

# CHAPTER 2 LITERATURE SURVEY

**Title: The association between sleep apnea and the risk of traffic accidents**

**Year:** 1999

**Author:** J. Teran-Santos, A. Jimenez-Gomez, and J. Cordero-Guevara

Obesity hypoventilation syndrome (OHS) is a major respiratory complication caused by severe obesity, being associated with significant morbidity, negative impacts on quality of life and reduced survival if not treated appropriately. Positive airway pressure therapy is the first-line treatment for OHS although the optimal modality remains unclear. The goal of this study is to identify the efficacy of home bi-level positive airway pressure therapy by comparison to continuous positive airway pressure therapy and determine the best strategy for patients with OHS.

his study will be conducted following the Preferred Reporting Items for Systematic Review and Meta-Analysis Protocols 2015 statement. We will search the following databases: PubMed, Web of Science, EMBASE, Cochrane Central Register of Controlled Trials and CINAHL. Ongoing studies will be identified through the ClinicalTrials.gov and WHO International Clinical Trials Registry Platform Search Portal. Grey literature will be recognised through Google Scholar and other search engines. Only randomised controlled trials meeting the eligibility criteria will be included. The risk of bias of the included studies will be evaluated through the Cochrane Collaboration’s tool. RevMan V.5.3.5 software will be used for data analysis. The Q statistic and I2 index will be used for investigating heterogeneity, and subgroup analysis or sensitivity analysis will be used to explore the source of heterogeneity. In addition, the Grading of Recommendations Assessment, Development and Evaluation system will be used to inspect the quality of evidence.

Strengths and limitations of this study

* The first systematic review and meta-analysis to compare the efficacy of home bi-level positive airway pressure and continuous positive airway pressure therapy in patients with obesity hypoventilation syndrome (OHS).
* This study will help identify the most appropriate home positive airway pressure therapy strategy for patients with OHS.
* Only randomized controlled trials, representing a high quality of evidence, will be included in this study.
* To determine the most appropriate patients to receive bi-level positive airway pressure therapy through subgroup analysis.

#### Data collection process

Two investigators (BP and YT) will independently extract data using a standardised data collection form. The extracted data will include: (1) study characteristics such as first author, journal, year and study design; (2) patients’ characteristics including the number of patients in each group, age, sex, BMI and severity of OHS; (3) intervention characteristics (type of PAP therapy, mode of bilevel therapy, inspiratory PAP, expiratory PAP and back-up rate), mean daily use of ventilation and treatment duration; and (4) primary and secondary outcomes as described above. For crossover trials, we will only extract the data from the first phase, primarily because of the carry-over effect. When the extraction process is complete, we will merge the two collections into one for further analysis. Should there be any disagreements at this stage, the two investigators will reach an agreement through discussion. If agreement cannot be reached, the third investigator (LQ) will participate in the discussion to reach a final conclusion.

**Title** Automatic detection of respiratory arrests in OSA patients using PPG and machine learning techniques.

**Year:** 2017

**Author:**  M. K. Uçar, M. R. Bozkurt, C. Bilgin, and K. Polat

Obstructive sleep apnea is a syndrome which is characterized by the decrease in air flow or respiratory arrest depending on upper respiratory tract obstructions recurring during sleep and often observed with the decrease in the oxygen saturation. The aim of this study was to determine the connection between the respiratory arrests and the photoplethysmography (PPG) signal in obstructive sleep apnea patients. Determination of this connection is important for the suggestion of using a new signal in diagnosis of the disease. Thirty-four time-domain features were extracted from the PPG signal in the study. The relation between these features and respiratory arrests was statistically investigated. The Mann–Whitney U test was applied to reveal whether this relation was incidental or statistically significant, and 32 out of 34 features were found statistically significant. After this stage, the features of the PPG signal were classified with k-nearest neighbors classification algorithm, radial basis function neural network, probabilistic neural network, multilayer feedforward neural network (MLFFNN) and ensemble classification method. The output of the classifiers was considered as apnea and control (normal). When the classifier results were compared, the best performance was obtained with MLFFNN. Test accuracy rate is 97.07 % and kappa value is 0.93 for MLFFNN. It has been concluded with the results obtained that respiratory arrests can be recognized through the PPG signal and the PPG signal can be used for the diagnosis

**Title** Sleep stage and obstructive apneaic epoch classification using single-lead ECG

**Year:** 2010

**Author:**  B. Yılmaz, M. H. Asyalı, E. Arıkan, S. Yetkin, and F. Özgen

Polysomnography (PSG) is used to define physiological sleep and different physiological sleep stages, to assess sleep quality and diagnose many types of sleep disorders such as obstructive sleep apnea. However, PSG requires not only the connection of various sensors and electrodes to the subject but also spending the night in a bed that is different from the subject's own bed. This study is designed to investigate the feasibility of automatic classification of sleep stages and obstructive apneic epochs using only the features derived from a single-lead electrocardiography (ECG) signal.

For this purpose, PSG recordings (ECG included) were obtained during the night's sleep (mean duration 7 hours) of 17 subjects (5 men) with ages between 26 and 67. Based on these recordings, sleep experts performed sleep scoring for each subject. This study consisted of the following steps: (1) Visual inspection of ECG data corresponding to each 30-second epoch, and selection of epochs with relatively clean signals, (2) beat-to-beat interval (RR interval) computation using an R-peak detection algorithm, (3) feature extraction from RR interval values, and (4) classification of sleep stages (or obstructive apneaic periods) using one-versus-rest approach. The features used in the study were the median value, the difference between the 75 and 25 percentile values, and mean absolute deviations of the RR intervals computed for each epoch. The k-nearest-neighbor (kNN), quadratic discriminant analysis (QDA), and support vector machines (SVM) methods were used as the classification tools. In the testing procedure 10-fold cross-validation was employed.

### Results

QDA and SVM performed similarly well and significantly better than kNN for both sleep stage and apneaic epoch classification studies. The classification accuracy rates were between 80 and 90% for the stages other than non-rapid-eye-movement stage 2. The accuracies were 60 or 70% for that specific stage. In five obstructive sleep apnea (OSA) patients, the accurate apneaic epoch detection rates were over 89% for QDA and SVM.

**Title:** Evaluation of machine-learning approaches to estimate sleep apnea severity from at-home oximetry recordings

**Year:** 2018

**Author**: G. C. Gutierrez-Tobal, D. Alvarez, A. Crespo, F. Del Campo, and R. Hornero.

Complexity, costs, and waiting lists issues demand a simplified alternative for sleep apnea-hypopnea syndrome (SAHS) diagnosis. The blood oxygen saturation signal (SpO2 ) carries useful information about SAHS and can be easily acquired from overnight oximetry. In this study, SpO 2 single-channel recordings from 320 subjects were obtained at patients' home. They were used to automatically obtain statistical, spectral, non-linear, and clinical SAHS-related information. Relevant and non-redundant data from these analyses were subsequently used to train and validate four machine-learning methods with ability to classify SpO 2 signals into one out of the four SAHS-severity degrees (no-SAHS, mild, moderate, and severe). All the models trained (linear discriminant analysis, 1-vs-all logistic regression, Bayesian multi-layer perceptron, and Ada Boost), outperformed the diagnostic ability of the conventionally-used 3% oxygen desaturation index. An Ada Boost model built with linear discriminants as base classifiers reached the highest figures. It achieved 0.479 Cohen's a in the SAHS severity classification, as well as 92.9%, 87.4%, and 78.7% accuracies in binary classification tasks using increasing severity thresholds (apnea-hypopnea index: 5, 15, and 30 events/hour, respectively). These results suggest that machine learning can be used along with SpO 2 information acquired at patients' home to help in SAHS diagnosis simplification

The study involved 320 adult subjects referred to the Hospital Universitario Rio Hortega in Valladolid (Spain) due to SAHS suspicion. All of them were diagnosed through an in-lab overnight PSG (E-series, Compumedics). A physician computed AHI following the rules of the American Academy of Sleep Medicine (AASM) [10], which was used as the gold standard. Participants with an AHI < 5 events per hour (e/h) were considered as no-SAHS subjects. Those showing an AHI in the ranges [5, 15) e/h, and [15, 30) e/h, were diagnosed as mild and moderate SAHS patients, respectively. Finally, subjects with AHI ≥ 30 e/h were diagnosed as severe. Each participant also conducted an at-home NPO. This was randomly carried out within the 24 hours before or after PSG to minimize the night-to-night variability effect [20]. Participants were divided into two sets: a training set composed of the first 60% consecutive subjects (ntr=193, 19 no-SAHS, 31 mild, 35 moderate, 108 severe) and a test set composed of the remaining 40% (ntest=127, 10 no-SAHS, 24 mild, 21 moderate, 72 severe). All of them gave an informed consent. The Ethics Committee of the Hospital accepted the protocol (approval number: CEIC 7/13). Table I displays demographic and clinical data of the subjects (mean ± standard deviation). No statistically significant differences (p-value>0.01) were found in age, body mass index (BMI), or AHI. SpO2 signals were acquired during NPO (overnight length) by the use of a portable oximeter (Nonin WristOx2 3150, sampling rate 1 Hz). Artifacts due to movements were automatically removed during preprocessing. Thus, the SpO2 values equal to zero, as well as the differences between consecutive SpO2 samples ≥4%, were considered artifacts [15]. Figure 1 shows examples of SpO2 recordings from each SAHS severity degree (no-SAHS, mild, moderate, and severe). Different patterns are observed, showing a tendency of higher amount of baseline falls (desaturations) in SpO2 as SAHS severity increases. However, a comprehensive analysis is required to predict the class of each recording.

Common statistics: First-to-fourth order statistical moments were extracted from SpO2 in time domain: mean (Mt1), standard deviation (Mt2), skewness (Mt3), and kurtosis (Mt4). These features characterize central tendency, dispersion, asymmetry, and peakedness of a given time series [13], [18], [19]. Previous in-lab studies reported statistically significant lower values of Mt1, Mt3 and Mt4, as well as higher values of Mt2, in the SpO2 signals from SAHS positive subjects [18].

**Title**: Decision tree based diagnostic system for moderate to severe obstructive sleep apnea. Journal of medical systems

**Year:** 2014

**Author**: H. Ting, Y. T. Mai, H. C. Hsu, H. C. Wu, and M. H. Tseng

One of the major modern medical issues, obstructive sleep apnea (OSA), particularly at moderate to severe levels, may potentially cause cardiovascular morbidity and mortality. However, polysomnography (PSG), a gold standard tool in diagnosing OSA, is cumbersome, has limited availability, and is costly and time-consuming. Clinical prediction models thus are absolutely necessary in screening patients with OSA. Furthermore, the performance of the published prediction formulas is not satisfactory for Chinese populations. The aim of this study was to develop and validate a simple and accurate prediction system for the diagnosis of moderate to severe OSA by integrating an expert-based feature extraction technique with decision tree algorithms which have automatic feature selection capability in screening the moderate to severe OSA cases in Taiwan. Moreover, the backward stepwise multivariable logistic regression model and four other decision tree algorithms were also employed for comparison. The results showed that the proposed best prediction formula, with an overall accuracy reaching to 96.9 % in sensitivity = 98.2 % and specificity = 93.2 %, could present a good tool for screening moderate and severe Taiwanese OSA patients who require further PSG evaluation and medical intervention. Results also indicate that the proposed best prediction formula is simple, accurate, and reliable, and outperforms all the other prediction formulae considered in the present study. The proposed clinical prediction formula derived from three non-invasive features (Sex, Age, and AveSBP) may help prioritize patients for PSG studies as well as avoid a diagnosis of PSG in subjects who have a low probability of having the disease.

# CHAPTER 3 SYSTEM ANALYSIS

### 3.1 EXISTING SYSTEM

This work are related to systematic comparison of signal combinations and data mining techniques, and to data preprocessing.

First, we train our classifiers with only minimal preprocessing of raw-data and still achieve rather good classification results while the main body of existing works performs feature extraction before training.

Second, we conduct an exhaustive, systematic comparison of all possible combinations of signals that are relevant for home monitoring with five data mining techniques. We conduct an exhaustive, systematic comparison of all possible combinations of signals that are relevant for home monitoring. But it is the static output from raw data.so it id delay for find disease diagnosed. So our project we can take dataset for patient list apply algorithm. We can quickly diagnosed disease. Existing System we apply the five data mining techniques so we take time. Here we take monitoring dataset so we cannot find the exact output just we can find the neighbor for disease. Our input data is blood pressure form dataset.so we cannot find the other input data like sugar, age. Distance based learning is not clear which type of distance to use which attribute to use to produce the best result. Computation cost is quite high because we need compute distance of each query instance to all training samples. Our dataset contain particular information so our output is not dynamic.

The model cannot be interrupted. There is no description of the learned concepts. It is computationally expensive find the k nearest neighbor when the data is very large. Performance distance on the number of the dimensions. In order to record the data at home without any personnel, the data must be easy to record.

**PROPOSEDSYSTEM** Accurate and efficient data classification techniques are of vital importance to many problems, and are rapidly developing in recent decades. K-Nearest Neighbor algorithm (KNN), as one of the most important algorithms, is widely used in text categorization, predictive analysis, data mining and image recognition, etc. To accelerate the algorithm and to optimize the parallel implementation solution are two key issues of KNN.

It is the very important and lazy algorithm. Compare to other algorithm it is the very easy and we can find exact solution or issues.it is compare to two information and we can find the distance from solution or problems from our information.

Here we have kidney disease dataset. Upload the dataset to SQL database. We apply the data mining technique we will find the disease list compare blood pressure with KNN algorithm. To develop useful and acceptable solutions for sleep monitoring at home we investigate in this paper how many and which physiological signals should be collected. . From a user's point of view less sensors as well as comfortable sensors that are easy to use are better. These values are regarded to correspond to very reliable classification, also according to related works. Effective the data is large. Among algorithm of collaborative filtering recommendation knn has lowest time complexity.

very simple implementation. Robust with regard to the search space; for instance, classes don't have to be linearly separable. Classifier can be updated online at very little cost as new instances with known classes are presented. Few parameters to tune: distance metric and k.

* + 1. Advantages

1. Simple technique that is easily implemented

2. Building model is cheap

### REQUIREMENT ANALYSIS AND SPECIFICAITON

The requirement engineering process of feasibility study, requirements elicitation and analysis, requirement specification, requirements validation and requirement management. Requirement elicitation and analysis is am iterative process that can be represented as a spiral of activities, namely requirements discovery, requirements classification and organization, requirement negotiation and requirements documentation.

### INPUT REQUIREMENT

The user interface design is very important for any application. The interface design describes how the software communicated within itself, to system that interpreted with it and with humans who use it. The interface is a packing for computer software if the interface is easy to learn, simple to use. If the interface design is very good, the user will fall into an interactive software application.

The input design is the process of converting the user-oriented inputs into the computer-based format. Errors entered by data entry operations can be controlled by input design. The data is fed into the system using simple interactive forms. The forms have been supplied with messages so that user can enter data without facing any difficulty.

The data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into the system. The goal for designing input data is to make data entry as easy, logical and free from errors.

The objectives of input design are:

* To produce a cost effective method of input
* To make the input forms understandable to the user
* To ensure the validation of data input
* To achieve the highest position level of accuracy

The various activities to be performed for the overall input processors are:

* Data recording at its source.
* Data transfer to input form.
* Data conversation to computer acceptable mode.
* Data validation.
* Data flow control.
* Data correction if necessary.

### OUTPUT REQUIREMENT

The system output is the most important and direct source of information to the user. So intelligible output design improves the relationship with the user and helps in decision-making. Outputs from the computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of these results for later consultation.

A major form of output is a hard copy obtained from the printer. These printouts are designed to include the exact requirements of the user. The outputs required by the end-user are defined during the logical design stages.

Two phases of the output design are:

* Output specification

Computer outputs are the most important and direct source of information to the user. A quality output is one which meets the requirements of the end user and which presents information in a way which is clear, easy to read and visually attractive. The screens are designed in such a way that the outputs are provided to the user in an understandable form.

The objectives of output design are:

* Design output to serve the indented purpose.
* Provide output on time.
* Assume that output is where it is needed.
* Designoutputtofittheuser.

### FEASIBILITY STUDY

The feasibility study is carried out to test whether the proposed system is worth being implemented. The proposed system will be selected if it is best enough in meeting the performance requirements.

The feasibility carried out mainly in three sections namely.

**•** Economic Feasibility

• Technical Feasibility

• Behavioural Feasibility

### TECHNICAL FEASIBILITY

### This study center around the system’s department hardware, software and to what extend it can support the proposed system department is having the required hardware and software there is no question of increasing the cost of implementing the proposed system. The criteria, the proposed system is technically feasible and the proposed system can be developed with the existing facility.

### ECONOMIC FEASIBILITY

### Economic analysis is the most frequently used method for evaluating effectiveness of the proposed system. More commonly known as cost benefit analysis. This procedure determines the benefits and saving that are expected from the system of the proposed system. The hardware in system department if sufficient for system development.

### MINIMUM HARDWARE REQUIREMENTS

|  |  |
| --- | --- |
| Processor | Core i3, 2.4 GHz |
| Hard disk | 500 GB |
| RAM | 4GB |
| Monitor | 14/15 inches Color |

* 1. **SOFTWARE REQUIEMENTS**

|  |  |
| --- | --- |
| O/S | Windows7 |
| Language | Java |
| IDE | NetBeans8.2 |

|  |  |
| --- | --- |
| Database | MySql |
|  |  |

* 1. **SOFTWARE SPECIFICATION**

**Java**

Java is a programming language originally developed by James Gosling at Sun Microsystems (now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere." Java is currently one of the most popular programming languages in use, particularly for client-server web applications.

**Java Platform:**

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java byte code, instead of directly to platform-specific machine code. Java byte code instructions are analogous to machine code, but are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware.

End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets. Standardized libraries provide a generic way to access host-specific features such as graphics, threading, and networking.

A major benefit of using byte code is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executable would. Just-in-Time compilers were introduced from an early stage that compiles byte codes to machine code during runtime.

Just as application servers such as Glass Fish provide lifecycle services to web applications, the Net Beans runtime container provides them to Swing applications. All new shortcuts should be registered in "Key maps/Net Beans" folder. Shortcuts installed INS Shortcuts folder will be added to all key maps, if there is no conflict. It means that if the same shortcut is mapped to different actions in Shortcut folder and current key map folder (like Key map/Net Beans), the Shortcuts folder mapping will be ignored.

\* Database Explorer Layer API in Database Explorer

\* Loaders-text-dB schema-Actions in Database Explorer

\* Loaders-text-sq.-Actions in Database Explorer

\* Plug-in Registration in Java EE Server Registry

The keyword public denotes that a method can be called from code in other classes, or that a class may be used by classes outside the class hierarchy. The class hierarchy is related to the name of the directory in which the .java file is located.

The keyword static in front of a method indicates a static method, which is associated only with the class and not with any specific instance of that class. Only static methods can be invoked without a reference to an object. Static methods cannot access any class members that are not also static. The keyword void indicates that the main method does not return any value to the caller. If a Java program is to exit with an error code, it must call System. Exit () explicitly.

The method name "main" is not a keyword in the Java language. It is simply the name of the method the Java launcher calls to pass control to the program. Java classes that run in managed environments such as applets and Enterprise JavaBeans do not use or need a main () method. A Java program may contain multiple classes that have main methods, which means that the VM needs to be explicitly told which class to launch from.

The Java launcher launches Java by loading a given class (specified on the command line or as an attribute in a JAR) and starting its public static void main(String[]) method. Stand-alone programs must declare this method explicitly. The String [] rags parameter is an array of String objects containing any arguments passed to the class. The parameters to main are often passed by means of a command line.

**Java a High-level Language:**

A high-level programming language developed by Sun Microsystems. Java was originally called OAK, and was designed for handheld devices and set-top boxes. Oak was unsuccessful so in 1995 Sun changed the name to Java and modified the language to take advantage of the burgeoning World Wide Web.

Java source code files (files with a .java extension) are compiled into a format called byte code (files with a .class extension), which can then be executed by a Java interpreter. Compiled Java code can run on most computers because Java interpreters and runtime environments, known as Java Virtual Machines (VMs). Byte code can also be converted directly into machine language instructions by a just-in-time compiler (JIT).

Java is a general purpose programming language with a number of features that make the language well suited for use on the World Wide Web. Small Java applications are called Java applets and can be downloaded from a Web server and run on your computer by a Java-compatible Web browser, such as Netscape Navigator or Microsoft Internet Explorer.

Object-Oriented Software Development using Java: Principles, Patterns, and Frameworks contain a much applied focus that develops skills in designing software-particularly in writing well-designed, medium-sized object-oriented programs. It provides a broad and coherent coverage of object-oriented technology, including object-oriented modeling using the Unified Modeling Language (UML) object-oriented design using Design Patterns, and object-oriented programming using Java.

**Net Beans**

The **Net Beans Platform** is a reusable framework for simplifying the development of Java Swing desktop applications. The Net Beans IDE bundle for Java SE contains what is needed to start developing Net Beans plug-in and Net Beans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Canter module to allow users of the application to download digitally-signed upgrades and new features directly into the running application.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

* User interface management (e.g. menus and toolbars)
* User settings management
* Storage management (saving and loading any kind of data)
* Window management
* Wizard framework (supports step-by-step dialogs)
* Net Beans Visual Library
* Integrated Development Tools

**J2EE**

A **Java EE application** or a **Java Platform, Enterprise Edition application** is any deployable unit of Java EE functionality. This can be a single Java EE module or a group of modules packaged into an EAR file along with a Java EE application deployment descriptor.

Enterprise applications can consist of the following:

* EJB modules (packaged in JAR files);
* Web modules (packaged in WAR files);
* connector modules or resource adapters (packaged in RAR files);
* Session Initiation Protocol (SIP) modules (packaged in SAR files);
* application client modules
* Additional JAR files containing dependent classes or other components required by the application;

**Wimp Server**

**WAMP**s are packages of independently-created programs installed on computers that use a Microsoft Windows operating system.

Apache is a web server. MySQL is an open-source database. PHP is a scripting language that can manipulate information held in a database and generate web pages dynamically each time content is requested by a browser. Other programs may also be included in a package, such as phpMyAdmin which provides a graphical user interface for the MySQL database manager, or the alternative scripting languages Python or Perl.

**MySQL**

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

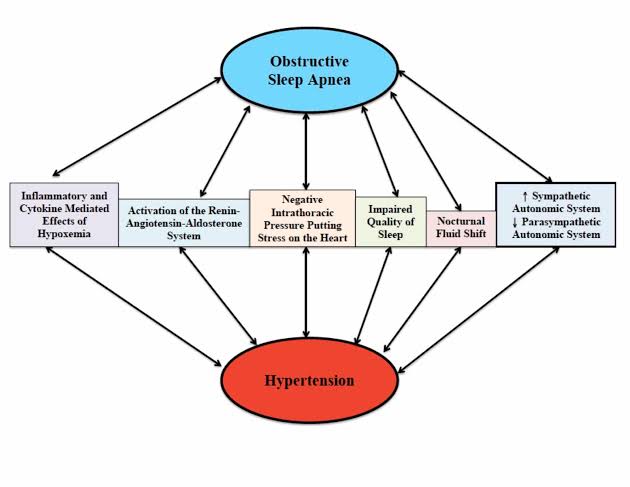
Free-software-open source projects that require a full-featured database management system often use MySQL. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, hob, Drupal and other software built on the LAMP software stack.

**Platforms and interfaces**

Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java. In addition, an ODBC interface called Modoc allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion. The MySQL server and official libraries are mostly impleme nted in ANSI C/ANSI C++.

# CHAPTER 4 SYSTEM DESIGN

### ER DIAGRAM

****

* 1. **DATA FLOW DIAGRAM**

A picture is worth a thousand words. A Data Flow Diagram (DFD) is traditional visual representation of the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

Dataset

Connected to Database

View and Upload

Output

Preprocessor

Fetch Data

KNN

nn

Start

Exit

### UML DIAGRAMS

UML stands for Unified Modeling Language. It’s a rich language to model software solutions, application structures, system behavior and business processes. There are 14 UML diagram types to help you model these behaviors. Unified Modeling Language™ (UML®) is a standard visual modeling language intended to be used for

* modeling business and similar processes,
* analysis, design, and implementation of software-based systems

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.

Specification explained that process:

* provides guidance as to the order of a team’s activities,
* specifies what artifacts should be developed,
* directs the tasks of individual developers and the team as a whole, and
* offers criteria for monitoring and measuring a project’s products and activities.

UML is intentionally process independent and could be applied in the context of different processes. Still, it is most suitable for use case driven, iterative and incremental development processes. An example of such process is Rational Unified Process (RUP).UML is not complete, and it is not completely visual. Given some UML diagram, we can't be sure to understand depicted part or behavior of the system from the diagram alone. Some information could be

intentionally omitted from the diagram, some information represented on the diagram could have different interpretations, and some concepts of UML have no graphical notation at all, so there is no way to depict those on diagrams. For example, semantics of multiplicity of actors and multiplicity of use cases on use case diagrams is not defined precisely in the UML specification and could mean either concurrent or successive usage of use cases.

Name of an abstract classifier is shown in italics while final classifier has no specific graphical notation, so there is no way to determine whether classifier is final or not from the diagram.

#### List of UML Diagram Types

So, what are the different UML diagram types? There are two main categories; structure diagrams and behavioral diagrams. Click on the links to learn more about a specific diagram type.

#### Structure Diagrams

Structure diagrams show the things in the modeled system. In a more technical term, they show different objects in a system. Behavioral diagrams show what should happen in a system. They describe how the objects interact with each other to create a functioning system.

#### Class Diagram

Class diagrams are the main building block of any object-oriented solution. It shows the classes in a system, attributes, and operations of each class and the relationship between each class. In most modeling tools, a class has three parts. Name at the top, attributes in the middle and operations or methods at the bottom.

In a large system with many related classes, classes are grouped together to create class diagrams. Different relationships between classes are shown by different types of arrows.

#### Component Diagram

A component diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems with many components. Components communicate with each other using interfaces. The interfaces are linked using connectors. The image below shows a component diagram.

#### Deployment Diagram

A deployment diagram shows the hardware of your system and the software in that hardware. Deployment diagrams are useful when your software solution is deployed across multiple machines with each having a unique configuration. Below is an example deployment diagram.

#### Package Diagram

As the name suggests, a package diagram shows the dependencies between different packages in a system. Check out this wiki article to learn more about the dependencies and elements found in package diagrams.

#### Composite Structure Diagram

Composite structure diagrams are used to show the internal structure of a class. For a detailed explanation of composite structure diagrams, click here.

#### Use Case Diagram

As the most known diagram type of the behavioral UML diagrams, use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions interact.

It’s a great starting point for any project discussion because you can easily identify the main actors involved and the main processes of the system. You can create use case diagrams using our tool and/or get started instantly using our use case templates.

#### Activity Diagram

Activity diagrams represent workflows in a graphical way. They can be used to describe the business workflow or the operational workflow of any component in a system. Sometimes activity diagrams are used as an alternative to State machine diagrams. Check out this wiki article to learn about symbols and usage of activity diagrams.

#### Sequence Diagram

Sequence diagrams in UML show how objects interact with each other and the order those interactions occur. It’s important to note that they show the interactions for a scenario. The processes are represented vertically, and interactions are shown as arrows. This article explains the purpose and the basics of Sequence diagrams. Also, check out this complete Sequence Diagram Tutorial to learn more about sequence diagrams. You can also instantly start drawing using our sequence diagram templates.

fig 4.4 Use case diagram

Load dataset

Id, blood pressure, age

,

Load()

FecthData

Blood pressure age

Count()

Upload and view

Data set

jFileChooser()

Output

Get the Disease list

ArrayList()

KNN Algorithm

Processed data

√∑(pi – qi)2

Fig 4.5 Class diagram

Load data

Connectedto Database

Pre-processor

KNN algorithm

Kidney Dataset

Avoid the empty

Space

Fetch the data

Interesting topic

Analysis

Fig 4.6 Sequence diagram

Start

KNN Algorithm

Preprocess

Fetch Data

Analyses

Kidney disease dataset

Upload

Fig 4.7 Activity Diagram

### CHAPTER 5 ARCHITECTURE

#### SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

**SYSTEM MODULE**

The System contains three modules functions namely:

* Load Dataset
* Preprocessor
* Fetch data
* KNN Algorithm
* Analysis

Dataset

Connected to Database

View and Upload

Preprocessor

classification

Output

Barchart

1. **MODULES EXPLAINATION:**

**Load Dataset**

In this module, we are using Kidney dataset based on patient information.

Dataset was collected and used as Document Dataset.

Our dataset contains id, blood pressure, age totally 15 records of the particular person.

Connected to sql form dataset information. Here we retrieve the data form sql to table view. You can read the file line by line and convert each line into an object representing that data. Actually there are couple of ways to read or parse CSV file in Java e.g. you can use a third party library like Apache commons CSV or you can use Scanner class, but in this example we will use traditional way of loading CSV file using Buffered Reader.

We take the string that we read from CSV file and split it up using the comma as the 'delimiter' (because it’s a CSV file). This creates an array with the all the columns of CSV file as we want, however values are still in Strings, so we need to convert them into proper type.  
  
That's all about how to load CSV file in Java without using any third partylibrary.  You have learned how to use Buffered Reader to read data from CSV file and then how to split comma separated String into String array by using String.split() method. Though you can do it even more easily by using third party library like Apache commons CSV, but knowing how to do it using pure Java will help you to learn key classes form JDK. 

Datasets

Id blood pressure

Age

Kidney.csv

Rc, Wc ,Pa

SG, AL,Su,Hemo,Pc

**Preprocessor:**

We upload the dataset after we need remove unnecessary space because it is problem for used algorithm calculation so we re-upload without empty space to sql database, not only empty space like “” “/” “’” because it the interrupted for calculation we upload with preprocessor. This section describes the options that the SQL preprocessor supports. It is the other way modify sql.

We will split dataset into train and test dataset for machine learning models we are going to build in future. Splinting the data into two parts i.e. Train and Test so that the test data we can use to test the machine learning model and train data will be the training data for machine learning model. To split the dataset we have again used the same library. We have used [train test split](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html) class which splits some % data into train and test. The best practice is to split 20% of your dataset into test data.

Our dataset contain 15 records for particular patient information. Preprocessing is nothing it can remove null values. Data pre-processing consists of a series of steps to transform raw data derived from data extraction. Therefore, these databases can have many quality control issues. Pre-processing aims at assessing and improving the quality of data to allow for reliable statistical analysis.Data “cleaning”—This step deals with missing data, noise, outliers, and duplicate or incorrect records while minimizing introduction of bias into the database. These methods are explored in detail in Chaps.

Data integration”—Extracted raw data can come from heterogeneous sources or be in separate datasets. This step reorganizes the various raw datasets into a single dataset that contain all the information required for the desired statistical analyses.

CSV Files

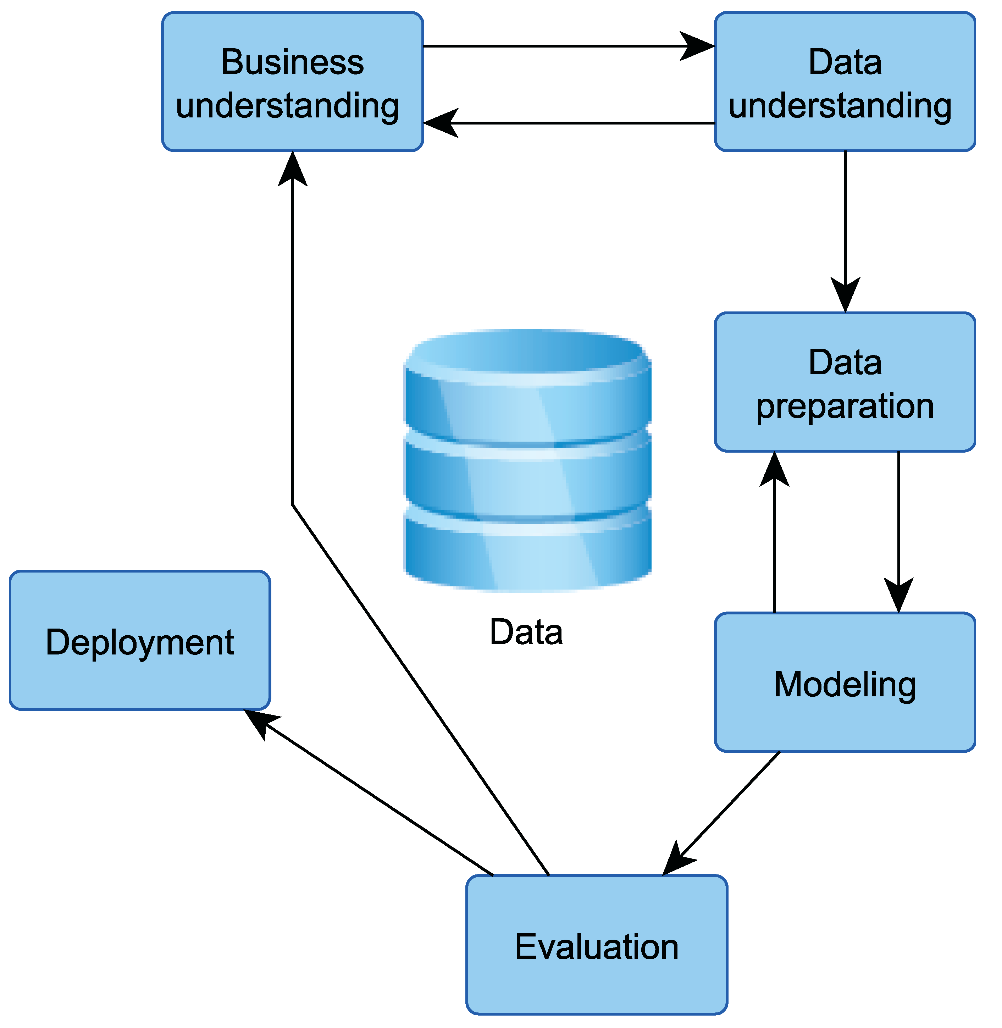
Preprocess

Database

Table View

**Fetch Data:**

  Data Fetching Sometimes you need to fetch data from the server when a route is activated. For example, before rendering a user profile, you need to fetch the user's data from the server. Providing information to help focus the search. Data mining is looking for hidden, valid, and potentially useful patterns in huge data sets. Mining is all about discovering unsuspected/ previously unknown relationships amongst the data.

A task that work same like information retrieval but more focuses on extracting relevant facts [5]. Machine Learning is support process that helps in mining data. We fetch the data from 10 columns for KNN algorithm for calculation. It is a process that has ability to learn user behavior and enhance the performance on specific task. Fetch information after we enter the KNN algorithm information. We get the particular information from database. It is clustering data. Data pre-processing and data mining techniques can lead to even better results. The two key contributions of this work are related to systematic comparison of signal combinations and data mining techniques, and to data preprocessing. First, we train our classifiers with only minimal preprocessing of raw-data and still achieve rather good classification results.

This image example of clustering. That image get the data understanding preprocessing information fetch data. Cluster the dataset information for KNN algorithm. Evaluation is here KNN. Deployment the particular information form dataset. Every null value ignore from dataset for while preprocessing. That particular information deployment for algorithm.

**KNN Algorithm:**

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970’s as a non-parametric technique. A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique we generally look at 3 important aspects.

1. Ease to interpret output

2. Calculation time

3. Predictive Power

First let us try to understand what exactly does K influence in the algorithm. If we see the last example, given that all the 6 training observation remain constant, with a given K value we can make boundaries of each class. These boundaries will segregate RC from GS. The same way, let’s try to see the effect of value “K” on the class boundaries. We put the input data blood pressure and age for algorithm calculation.

**Analysis**

In this module, we identify the chart related evaluation of our process.

We evaluate the following in a single graph.

* + Preprocess count
  + Input data
  + Knn
  + Find Distance.

We get distance from disease. It is using for KNN algorithm. Input data is the blood pressure and age our input data from dataset. Sort of the distance and determine

Analysis

KNN

Find distance

Our project output is who are getting blood pressure they are affected the disease. The work presented in this paper is a first step towards our long-term goal to reduce the number of individuals that are not diagnosed as OSA patients and to reduce the time until individuals are undergoing a clinical OSA diagnosis.

Performance is subject to future work. The achieved results are not only important for our overall goal, but they also bring new insights to the broader research field. To the best of our knowledge, our systematic comparison of using various combinations of respiration signals. . This indicates that it is possible to achieve with data mining of original sensor data good results if proper preprocessing is performed. Furthermore, we cannot see that the down-sampling. These values are regarded to correspond to very reliable classification, also according to related works. Every particular information the data blood pressure and age it is our input data. For SVM, the number of support vectors increases with the number of attributes, which combined with the added dimensions results in increasing classification times.

**CHAPTER 6 TESTING**

**Testing of Product:**

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system.  System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved.  The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test.  A series of tests are performed before the system is ready for the user acceptance testing.  Any engineered product can be tested in one of the following ways.  Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational.  Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

**UNIT TESTING:**

Unit testing is the testing of each module and the integration of the overall system is done.  Unit testing becomes verification efforts on the smallest unit of software design in the module.  This is also known as ‘module testing’.  The modules of the system are tested separately.  This testing is carried out during the programming itself.  In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module.  There are some validation checks for the fields.  For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included.  It is very easy to find error and debug the system.

**INTEGRATION TESTING:**

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function.  Integrated testing is systematic testing that can be done with sample data.  The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

1. Top-down integration testing.
2. Bottom-up integration testing.

**WHITE BOX TESTING:**

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases.  Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

**BLACK BOX TESTING:**

* + Black box testing is done to find incorrect or missing function
  + Interface error
  + Errors in external database access
  + Performance errors
  + Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’.  It tests the external behaviour of the system.  Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

**VALIDATION TESTING:**

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

# USER ACCEPTANCE TESTING:

User acceptance of the system is the key factor for the success of the system.  The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

# OUTPUT TESTING:

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format.  The output displayed or generated by the system under consideration.  Here the output format is considered in two ways.  One is screen and the other is printed format.  The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs.  For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

**System Implementation:**

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier.

* The active user must be aware of the benefits of using the system
* Their confidence in the software built up
* Proper guidance is impaired to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

**User Training:**

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important. Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

**Training on the Application Software:**

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

**Operational Documentation:**

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

**System Maintenance:**

The maintenance phase of the software cycle is the time in which software performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is to make adaptable to the changes in the system environment. There may be social, technical and other environmental changes, which affect a system which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only thru proper system maintenance procedures, the system can be adapted to cope up with these changes. Software maintenance is of course, far more than “finding mistakes”.

**Corrective Maintenance:**

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called Corrective Maintenance.

**Adaptive Maintenance:**

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore Adaptive maintenance termed as an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace

**Perceptive Maintenance:**

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancement are received from users. To satisfy requests in this category, Perceptive maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance.

**Preventive Maintenance:**

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques.

## CHAPTER 7

**CONCLUSION AND FUTURE ENHANCEMENT**

#### CONCLUSION

Thus our models can be either long-term dependency (KNN) topic models. We can manage big data set with KNN algorithm.

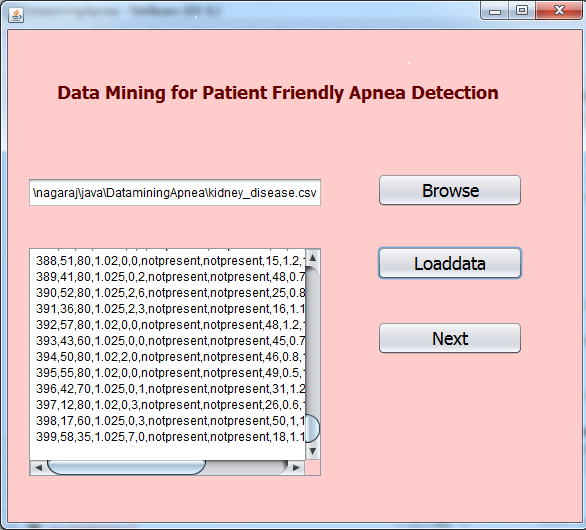
We can find the exact solution in our project and we get patient list and non-patient list. So we get awareness from this data and we defense the disease. Dynamic interests based on Blood pressure’ topic distributions at not only the last time period but also other multiple time periods in the past.

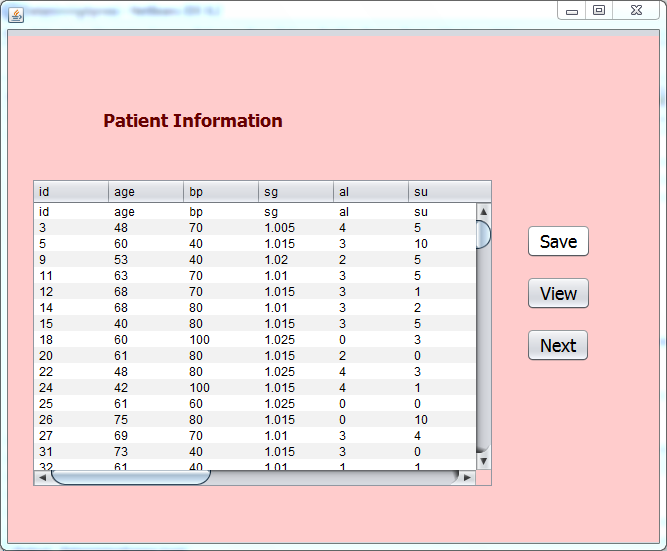
#### FUTURE ENHANCEMENT

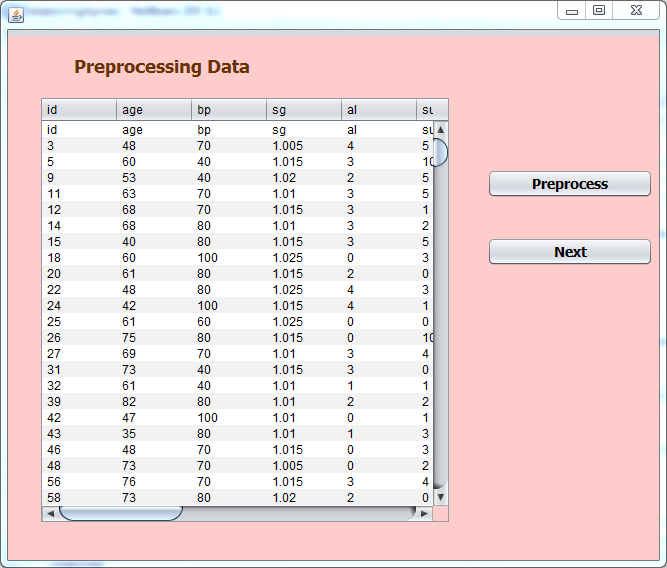
In terms of future work, we would like to investigate how injecting external knowledge would improve the results. We investigate which signals from classical PSG diagnosis could be used to achieve reliable classification. The results show that classifying data from only a single sensor.

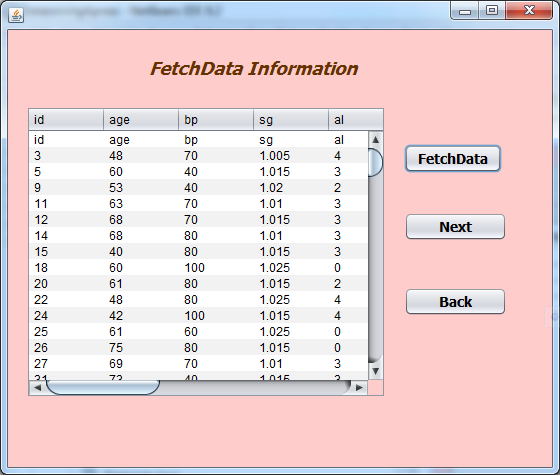
**APPENDICES**

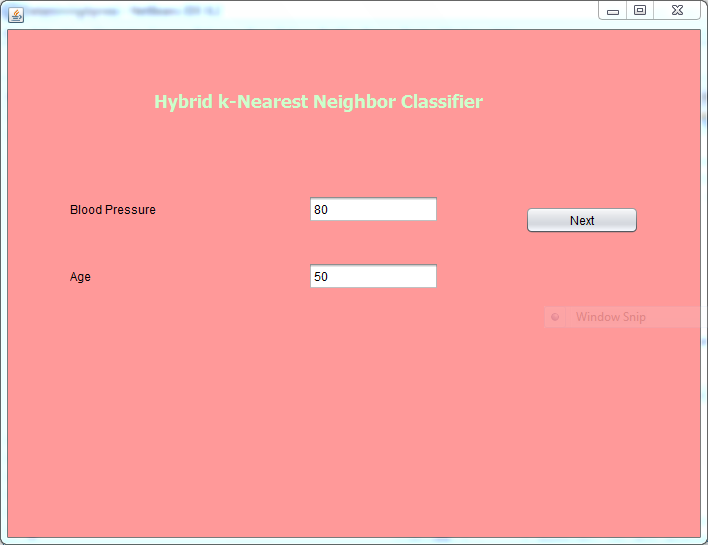
**A1.SAMPLE SCREENS**

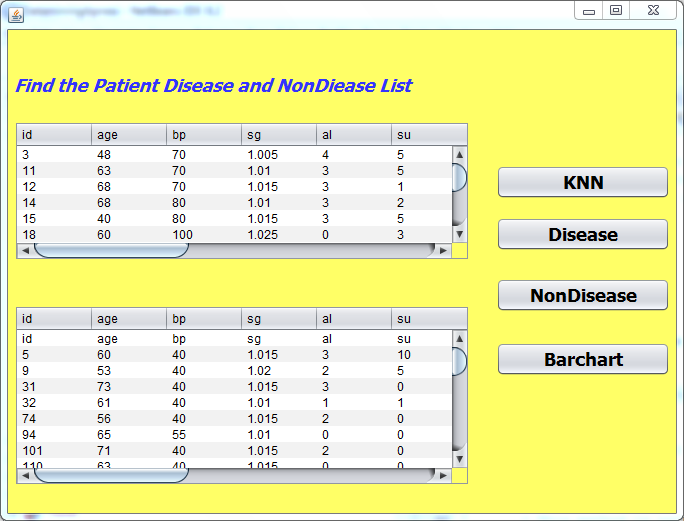


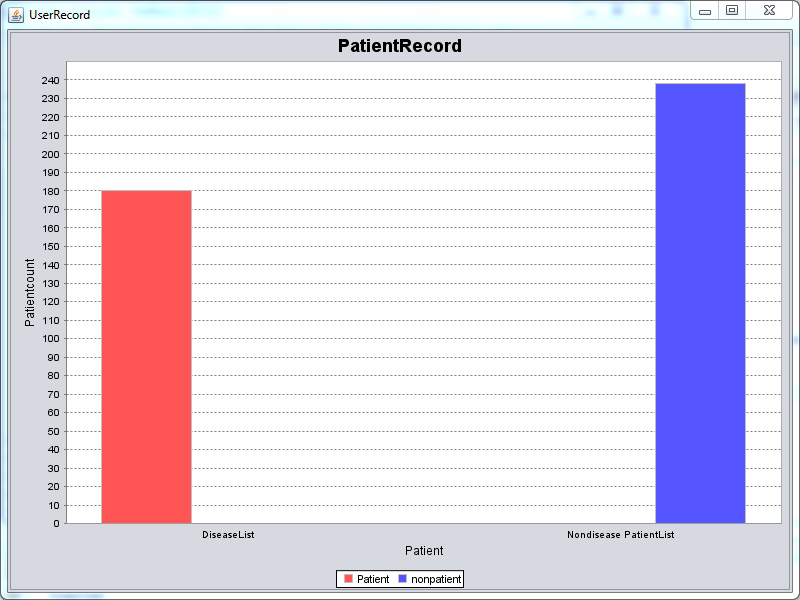












**A2.SAMPLE CODE**

package dataminingapnea;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.util.Arrays;

import java.util.Scanner;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author EGC

\*/

public class Loaddata extends javax.swing.JFrame {

/\*\*

\* Creates new form Loaddata

\*/

public static String fileName;

public Loaddata() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jButton1 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

fileRead = new javax.swing.JTextArea();

file\_path = new javax.swing.JTextField();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jButton1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jButton1.setText("Loaddata");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

fileRead.setColumns(20);

fileRead.setRows(5);

jScrollPane1.setViewportView(fileRead);

file\_path.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

file\_pathActionPerformed(evt);

}

});

jButton2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jButton2.setText("Browse");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jButton3.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jButton3.setText("Next");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

jPanel1.setBackground(new java.awt.Color(204, 255, 51));

jLabel1.setBackground(new java.awt.Color(153, 255, 102));

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel1.setForeground(new java.awt.Color(255, 0, 255));

jLabel1.setText("Data Mining for Patient Friendly Apnea Detection");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(31, 31, 31)

.addComponent(jLabel1)

.addContainerGap(117, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1,javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, 73, Short.MAX\_VALUE)

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane1)

.addGap(40, 40, 40)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jButton1,javax.swing.GroupLayout.DEFAULT\_SIZE,javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton3,javax.swing.GroupLayout.DEFAULT\_SIZE,javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(66, 66, 66))

.addGroup(layout.createSequentialGroup()

.addGap(42, 42, 42)

.addComponent(file\_path,javax.swing.GroupLayout.PREFERRED\_SIZE,236,javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jButton2)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(jPanel1,javax.swing.GroupLayout.DEFAULT\_SIZE,javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(16, 16, 16)

.addComponent(jPanel1,javax.swing.GroupLayout.PREFERRED\_SIZE,javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 26, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(layout.createSequentialGroup()

.addComponent(jButton1)

.addGap(28, 28, 28)

.addComponent(jButton3)

.addGap(211, 211, 211))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(file\_path,javax.swing.GroupLayout.PREFERRED\_SIZE,25,javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2))

.addGap(18, 18, 18)

.addComponent(jScrollPane1,javax.swing.GroupLayout.PREFERRED\_SIZE,352,javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())));

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

File file=new File(fileName);

try{

Scanner inputStream=new Scanner(file);

inputStream.next();

while(inputStream.hasNext()){

String data=inputStream.next();

//dataFieldtxt.setText(data+"\n");

String []values=data.split("\\t");

// System.out.println(values[4]);

fileRead.append(data+"\n");

}

inputStream.close();

}

catch(Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

JFileChooser chooser=new JFileChooser("./");

chooser.showOpenDialog(null);

File f= chooser.getSelectedFile();

fileName=f.getAbsolutePath();

file\_path.setText(fileName);

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

new Dataview().setVisible(true);

this.setVisible(false);

}

private void file\_pathActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for(javax.swing.UIManager.LookAndFeelInfoinfo: javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Loaddata.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Loaddata.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Loaddata.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Loaddata.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Loaddata().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JTextArea fileRead;

private javax.swing.JTextField file\_path;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JLabel jLabel1;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

// End of variables declaration

}

**Dataview.java**

package dataminingapnea;

import java.io.BufferedReader;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.InputStreamReader;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.Statement;

import java.util.Arrays;

import java.util.Vector;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.JScrollPane;

import javax.swing.table.DefaultTableModel;

import net.proteanit.sql.DbUtils;

/\*\*

\*

\* @author EGC

\*/

public class Dataview extends javax.swing.JFrame {

/\*\*

\* Creates new form Dataview

\*/

java.sql.Statement st;

Connection con;

// Vector<Object> columnNames = new Vector<Object>();

// Vector<Object> data = new Vector<Object>();

ResultSet rs;

private Object addfile;

public Dataview() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jButton4 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jPanel2 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jButton4.setBackground(new java.awt.Color(153, 255, 153));

jButton4.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jButton4.setText("Save");

jButton4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton4ActionPerformed(evt);

}

});

jButton2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jButton2.setText("View");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jButton3.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jButton3.setText("Next");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null},

{null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null},

{null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null},

{null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null}

},

new String [] {

"Title 1", "Title 2", "Title 3", "Title 4", "Title 5", "Title 6", "Title 7", "Title 8", "Title 9", "Title 10", "Title 11", "Title 12", "Title 13", "Title 14", "Title 15", "Title 16"

}

));

jScrollPane1.setViewportView(jTable1);

jPanel2.setBackground(new java.awt.Color(153, 255, 153));

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel1.setForeground(new java.awt.Color(255, 153, 153));

jLabel1.setText("Patient Information");

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(68, 68, 68)

.addComponent(jLabel1,javax.swing.GroupLayout.PREFERRED\_SIZE,318,javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(212, Short.MAX\_VALUE))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addContainerGap(28, Short.MAX\_VALUE)

.addComponent(jLabel1,javax.swing.GroupLayout.PREFERRED\_SIZE,51,javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(21, 21, 21))

);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(19, 19, 19)

.addComponent(jScrollPane1,javax.swing.GroupLayout.PREFERRED\_SIZE,475,javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton2)

.addComponent(jButton4)

.addComponent(jButton3))

.addGap(0, 0, Short.MAX\_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel2,javax.swing.GroupLayout.PREFERRED\_SIZE,javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(139, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(44, 44, 44)

.addComponent(jPanel2,javax.swing.GroupLayout.PREFERRED\_SIZE,javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(75, 75, 75)

.addComponent(jButton4)

.addGap(18, 18, 18)

.addComponent(jButton2)

.addGap(18, 18, 18)

.addComponent(jButton3))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(28, 28, 28)

.addComponent(jScrollPane1,javax.swing.GroupLayout.PREFERRED\_SIZE,360,javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(45, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1,javax.swing.GroupLayout.DEFAULT\_SIZE,javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel1,javax.swing.GroupLayout.PREFERRED\_SIZE,javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

try {

con=(com.mysql.jdbc.Connection) DriverManager.getConnection("jdbc:mysql:///Apenadata","root","");

st = con.createStatement();

// Statement stt;

BufferedReader br = new BufferedReader(new FileReader(Loaddata.fileName));

String s,r,k;

while ((s = br.readLine()) != null) {

r= s.replaceAll("'", "");

k= r.replaceAll("`", "");

String[] var = k.split(",");

// s = s.replace("/", "-");

if (var.length == 15) {

String s1 = var[0];

String s2 = var[1];

String s3 = var[2];

String s4 = var[3];

String s5 = var[4];

String s6 = var[5];

String s7 = var[6];

String s8 = var[7];

String s9 = var[8];

String s10 = var[9];

String s11 = var[10];

String s12 = var[11];

String s13 = var[12];

String s14 = var[13];

String s15 = var[14];

if(s1.equals("-")||s2.equals("-")||s3.equals("-")||s4.equals("-")||s5.equals("-")||s6.equals("-")||s7.equals("-")||s8.equals("-")||s9.equals("-")||s10.equals("-")||s11.equals("-")||s12.equals("-")||s13.equals("-")||s14.equals("-")||s15.equals("-"))

continue;

else

st.executeUpdate("INSERT INTO minetable values ('" + s1 + "','" + s2 + "','" + s3 + "','" + s4 + "','" + s5 + "','" + s6 + "','" + s7 + "','" + s8 + "','" + s9 + "','" + s10 + "','" + s11 + "','" + s12 + "','" + s13 + "','" + s14 + "','" + s15 + "')");

}

System.out.println("Inserted");

}

br.close();

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, ex);

System.out.println("done");

}

JOptionPane.showMessageDialog(this, " Loaded Completed");

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

jTable1.setAutoResizeMode(jTable1.AUTO\_RESIZE\_OFF);

try{

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/apenadata","root","");

String sql = "SELECT \* FROM minetable";

PreparedStatement pst = con.prepareStatement(sql);

rs=pst.executeQuery();

jTable1.setModel(DbUtils.resultSetToTableModel(rs));

}catch(Exception e){ System.out.println(e);}

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

new Preproces().setVisible(true);

this.setVisible(false);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Dataview.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Dataview.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Dataview.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Dataview.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

@Override

public void run() {

new Dataview().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JLabel jLabel1;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

// End of variables declaration

}

**Preproces.java**

import com.mysql.jdbc.PreparedStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.Statement;

import java.util.Vector;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author EGC

\*/

public class Preproces extends javax.swing.JFrame {

/\*\*

\* Creates new form Preproces

\*/

Statement st;

Vector<Object> columnNames = new Vector<Object>();

Vector<Object> data = new Vector<Object>();

ResultSet rs;

private Object addfile;

public Preproces() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jButton2 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jButton1 = new javax.swing.JButton();

jPanel1 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jButton2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

jButton2.setText("Preprocess");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null},

{null, null, null, null},

{null, null, null, null},

{null, null, null, null}

},

new String [] {

"Title 1", "Title 2", "Title 3", "Title 4"

}

));

jScrollPane1.setViewportView(jTable1);

jButton1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

jButton1.setText("Next");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jPanel1.setBackground(new java.awt.Color(51, 255, 102));

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel2.setText("Preprocessing Data");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(66, 66, 66)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 395, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(183, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 48, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(20, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(30, 30, 30)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jButton2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)))

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(79, 79, 79)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 20, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(66, 66, 66))

.addGroup(layout.createSequentialGroup()

.addGap(121, 121, 121)

.addComponent(jButton2)

.addGap(28, 28, 28)

.addComponent(jButton1)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))))

);

pack();

}// </editor-fold>

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Connection con=null;

jTable1.setAutoResizeMode(jTable1.AUTO\_RESIZE\_OFF);

try {

Class.forName("com.mysql.jdbc.Driver");

con=(com.mysql.jdbc.Connection) DriverManager.getConnection("jdbc:mysql:///apenadata","root","");

st=con.createStatement();

rs = st.executeQuery("Select \* from minetable");

ResultSetMetaData md = rs.getMetaData();

int columns = md.getColumnCount();

for (int i = 1; i <= columns; i++) {

columnNames.addElement(md.getColumnName(i));

}

while (rs.next()) {

Vector<Object> row = new Vector<Object>(columns);

for (int i = 1; i <= columns; i++) {

row.addElement(rs.getObject(i));

}

data.addElement(row);

}

rs.close();

st.close();

con.close();

} catch (Exception e) {

e.printStackTrace();

}

DefaultTableModel model = new DefaultTableModel(data, columnNames);

//DefaultTableModel myTableModel = (DefaultTableModel) this.myjTable.getModel();

jTable1.setModel(model);

jTable1.removeAll();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

new Fetchdata().setVisible(true);

this.setVisible(false);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Preproces.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Preproces.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Preproces.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Preproces.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Preproces().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel2;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

// End of variables declaration

}

**Finddiease.java**

package dataminingapnea;

//import java.awt.List;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.Collections;

import java.util.StringTokenizer;

import java.util.logging.Level;

import java.util.logging.Logger;

import java.util.List;

import net.proteanit.sql.DbUtils;

import sun.security.jca.GetInstance.Instance;

/\*\*

\*

\* @author EGC

\*/

public class Finddiease extends javax.swing.JFrame {

/\*\*

\* Creates new form Finddiease

\*/

public Finddiease() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jButton1 = new javax.swing.JButton();

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jButton2 = new javax.swing.JButton();

jScrollPane2 = new javax.swing.JScrollPane();

jTable2 = new javax.swing.JTable();

jButton3 = new javax.swing.JButton();

jPanel2 = new javax.swing.JPanel();

jButton4 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jButton1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jButton1.setText("Result");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jPanel1.setBackground(new java.awt.Color(255, 255, 102));

jLabel1.setBackground(new java.awt.Color(255, 255, 0));

jLabel1.setFont(new java.awt.Font("Tahoma", 3, 18)); // NOI18N

jLabel1.setForeground(new java.awt.Color(51, 51, 255));

jLabel1.setText("Find the Patient Disease and NonDiease List");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(24, 24, 24)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 475, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(21, 21, 21)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 57, Short.MAX\_VALUE)

.addGap(22, 22, 22))

);

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null},

{null, null, null, null},

{null, null, null, null},

{null, null, null, null}

},

new String [] {

"Title 1", "Title 2", "Title 3", "Title 4"

}

));

jScrollPane1.setViewportView(jTable1);

jButton2.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jButton2.setText("Disease");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jTable2.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null},

{null, null, null, null},

{null, null, null, null},

{null, null, null, null}

},

new String [] {

"Title 1", "Title 2", "Title 3", "Title 4"

}

));

jScrollPane2.setViewportView(jTable2);

jButton3.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jButton3.setText("NonDisease");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

jPanel2.setBackground(new java.awt.Color(255, 255, 0));

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 618, Short.MAX\_VALUE)

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 100, Short.MAX\_VALUE)

);

jButton4.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jButton4.setText("Exit");

jButton4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton4ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(23, 23, 23)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jButton2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)))

.addGroup(layout.createSequentialGroup()

.addGap(12, 12, 12)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton3)

.addComponent(jButton4, javax.swing.GroupLayout.PREFERRED\_SIZE, 103, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(685, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(19, 19, 19)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(39, 39, 39)

.addComponent(jButton1)

.addGap(32, 32, 32)

.addComponent(jButton2)

.addGap(86, 86, 86)

.addComponent(jButton3)

.addGap(37, 37, 37)

.addComponent(jButton4))

.addGroup(layout.createSequentialGroup()

.addGap(26, 26, 26)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 140, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(33, 33, 33)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 181, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

try{

BufferedReader br=new BufferedReader(new FileReader(Loaddata.fileName));

String line="";

List<Knn> li=new ArrayList();

br.readLine();

while((line=br.readLine())!=null){

String k[]=line.split(",");

Knn fr=new Knn(Integer.parseInt(k[2]),Integer.parseInt(k[1]));

li.add(fr);

//System.out.println(li);

}

Knn f1=new Knn(50,80);

List<Knn> so=new ArrayList();

so=Knn.callDistance(li,f1);

Collections.sort(so);

System.out.println(so);

for(int i=0;i<2;i++){

System.out.println(so.get(i));

}

// double k=6.02;

Knn n=new Knn();

// k=n.getdistance(k);

// System.out.println(k);

// int s=n.getsu();

// int o=n.getbp();

// System.out.println(n);

// System.out.println(o);

}

catch(Exception e){

System.out.println(e);

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

jTable1.setAutoResizeMode(jTable1.AUTO\_RESIZE\_OFF);

Connection con=null;

ResultSet rs;

try{

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/apenadata","root","");

String sql = "SELECT \* FROM minetable WHERE bp>=60";

PreparedStatement pst = con.prepareStatement(sql);

rs=pst.executeQuery();

jTable1.setModel(DbUtils.resultSetToTableModel(rs));

}catch(Exception e){ System.out.println(e);}

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

jTable2.setAutoResizeMode(jTable1.AUTO\_RESIZE\_OFF);

Connection con=null;

ResultSet rs;

try{

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/apenadata","root","");

String sql = "SELECT \* FROM minetable WHERE bp<60";

PreparedStatement pst = con.prepareStatement(sql);

rs=pst.executeQuery();

jTable2.setModel(DbUtils.resultSetToTableModel(rs));

}catch(Exception e){ System.out.println(e);}

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

System.exit(0);

}

/\*\*

\* @param args the command line arguments

\*/

/\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Finddiease.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Finddiease.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Finddiease.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Finddiease.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Finddiease().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JLabel jLabel1;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JTable jTable1;

private javax.swing.JTable jTable2;

// End of variables declaration

}

### REFERENCES

1. Z. Abedi, N. Naghavi, and F. Rezaeitalab. Detection and classification of sleep apnea using genetic algorithms and SVM-based classification of thoracic respiratory effort and oximetric signal features. Computational Intelligence, 33(4):1005–1018, 2017. COIN-OA-07-15-2237.R2
2. H. Ting, Y. T. Mai, H. C. Hsu, H. C. Wu, and M. H. Tseng. Decision tree based diagnostic system for moderate to severe obstructive sleep apnea. Journal of medical systems, 38(9):1–10, 2014
3. H. M. Al-Angari and A. V. Sahakian. Automated recognition of obstructive sleep apnea syndrome using support vector machine classifier. IEEE Transactions on Information Technology in Biomedicine, 16(3):463–468, 2012.
4. ApneaECG.https://www.physionet.org/physiobank/database/apneaecg/additional-information.txt. Accessed: September 2018.
5. H. Ting, Y. T. Mai, H. C. Hsu, H. C. Wu, and M. H. Tseng. Decision tree based diagnostic system for moderate to severe obstructive sleep apnea. Journal of medical systems, 38(9):1–10, 2014.