A Project Report on

CANCER PREDICTION USING DATAMINING

Project report submitted in partial fulfillment of the requirement for the award of the degree of B.Tech in Information Technology

BY

T.Tejasree (14251A12B0)

Under the guidance of

C. Sudhakar Reddy,
Associate Professor,
Department of IT

Department of Information Technology

G. Narayanamma Institute of Technology & Science (for Women)

Accredited by NBA & NAAC &Affiliated to JNTUH

(ISO 9001 Certified Institution)

Hyderabad-500104, T. S, INDIA

April,2018

G. NARAYANAMMA INSTITUTE OF TECHNOLOGY & SCIENCE
(For Women)
(Sponsored by G. Pulla Reddy Charities Trust, Hyderabad.)
Accredited by NBA & NAAC
(Approved by AICTE & Affiliated to J N T U H)
AN ISO 9001 : 2015 Certified Institution
Estd : 1997

NSF
ISR

CERTIFICATE

This is to certify that the project report entitled CANCER PREDICTION USING DATA MINING is bonafide work done by

T.Tejasree (14251A12B0)

during January to April, in partial fulfillment for the award of degree in B.Tech in Information Technology, from G.Narayanamma Institute of Technology and Science, Affiliated to JNTUH, Accredited by NBA & NAAC of UGC, Hyderabad.

Internal Guide	**Head of the Department**
C. Sudhakar Reddy,	Dr. I. Ravi Prakash Reddy
Associate Professor	

Acknowledgements

We would like to express our sincere thanks to **Dr. K. Ramesh Reddy, Principal**, GNITS, for providing the working facilities in the college. Our sincere thanks and gratitude to **Dr. I. Ravi Prakash Reddy, Professor and HOD**, Dept. of IT, GNITS for all the timely support and valuable suggestions during the period of our Major project.

We are extremely thankful to the major project coordinator **V. Sesha Bhargavi, Assistant Professor**, Department of IT, GNITS for her encouragement and support throughout the project.

We are extremely thankful and indebted to our internal guide, **C. Sudhakar Reddy**, **Associate Professor,** Deptartment of IT, GNITS for his constant guidance, continuous advice, encouragement and moral support throughout the Major project.

Finally, we would also like to thank all the faculty and staff of IT Department who helped us directly or indirectly and for their cooperation in completing the Major project work.

ABSTRACT

Cancer is the most important cause of death for both men and women. The early detection of cancer can be helpful in curing the disease completely. So, the requirement of techniques to detect the occurrence of cancer in early stage is increasing. Therefore, a cancer risk prediction system is proposed here. Through this Cancer Disease Prediction System, we try to predict cancer in early stages based on the symptoms stated by the user. Here, we propose a system that allows users to get instant guidance on their cancer disease through an intelligent system. The application is fed with various symptoms and the cancer disease associated with those symptoms. The application allows user to share their personal information, health related issues and their habits for cancer prediction. It then processes user specific details to check if there is a chance of them getting affected with a cancer. In health care industry, Data mining plays an important role for predicting diseases. This project uses data mining technology such as classification, clustering and prediction to identify potential cancer patients. The gathered data is preprocessed, fed into the database and classified to yield significant patterns using some classification algorithms. Then the data is clustered using clustering algorithms to separate cancer and non-cancer patient data. Finally, a prediction system is developed to analyze risk levels which help in prognosis. This system helps in detection of a person's predisposition for cancer before going for clinical and lab tests which is cost and time consuming.

Table of Contents

S.No Topic Page No.

Appendix

1.INTRODUCTION

Cancer is one of the most common diseases in the world that results in majority of death. It is caused by uncontrolled growth of cells in any of the tissues or parts of the body. Cancer may occur in any part of the body and may spread to several other parts. Only early detection of cancer at the benign stage and prevention from spreading to other parts in malignant stage could save a person's life. There are several factors that could affect a person's predisposition for cancer. Education is an important indicator of socioeconomic status through its association with occupation and life-style factors. A number of studies in developed countries have shown that cancer incidence varies between people with different levels of education. A high incidence of breast cancer has been found among those with high levels of education whereas an inverse association has been found for the incidence of cancers of the stomach, lung. Such differences in cancer risks associated with education also reflect in the differences in life-style factors and exposure to both environmental and work-related carcinogens. This project uses the data mining techniques to describe the association between cancer incidence pattern and risk levels of various factors and devises a risk prediction system for different types of cancer which helps in prognosis.

1.1 Domain Description

Data mining is the practice of automatically searching large stores of data to discover patterns and trends that go beyond simple analysis. Data mining uses sophisticated mathematical algorithms to segment the data and evaluate the probability of future events. Data mining is also known as Knowledge Discovery in Data (KDD).

The key properties of data mining are:

- Automatic discovery of patterns
- Prediction of likely outcomes
- Creation of actionable information
- Focus on large data sets and databases

Data mining is accomplished by building models. A model uses an algorithm to act on a set of data.

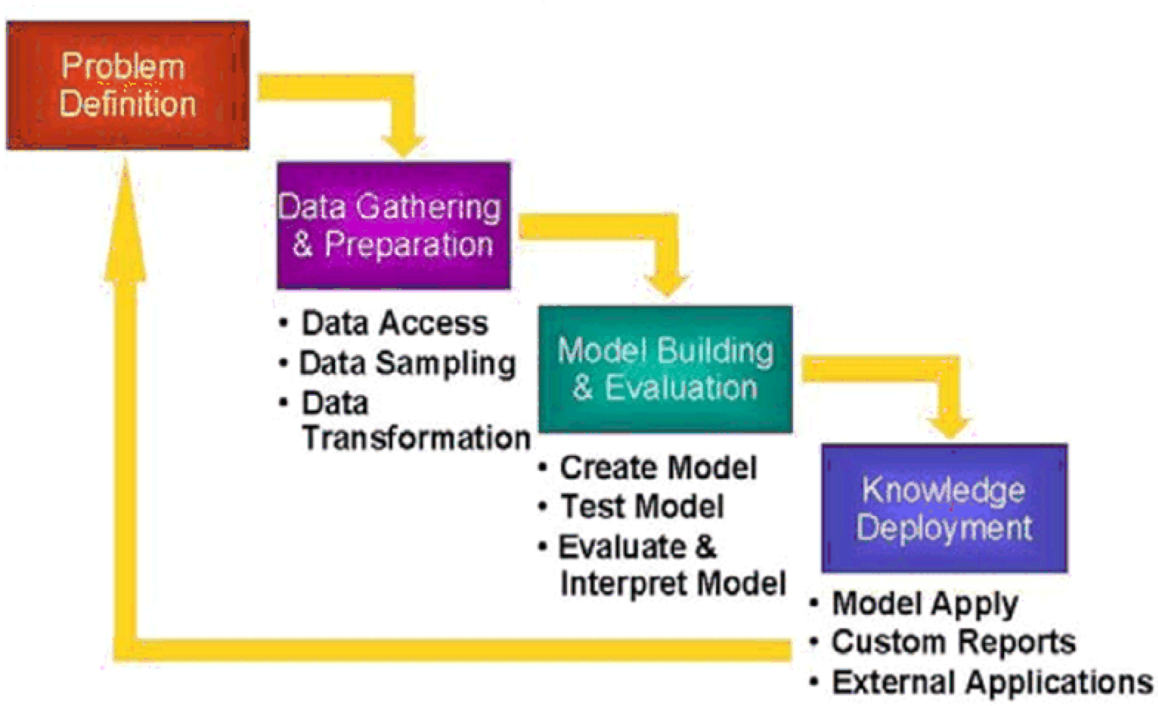


Fig. 1.1 Data mining Process

The data mining techniques used here are: Classification and Clustering.

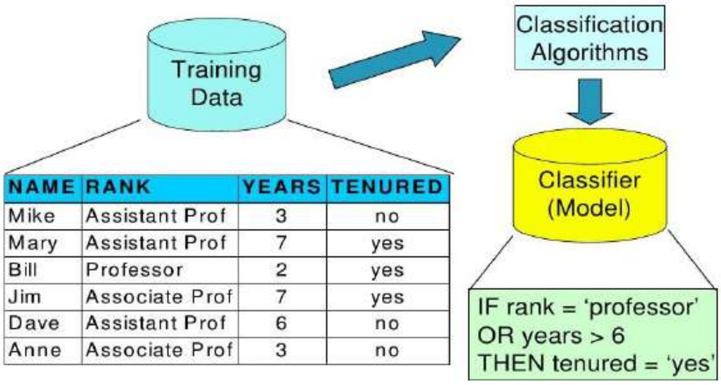
1.1.1 Classification

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. A classification task begins with a data set in which the class assignments are known.

The simplest type of classification problem is binary classification. In binary classification, the target attribute has only two possible values: for example, high credit rating or low credit rating. Multiclass targets have more than two values: for example, low, medium, high, or unknown credit rating.

In the model build (training) process, a classification algorithm finds relationships between the values of the predictors and the values of the target. Different classification algorithms use different techniques for finding relationships. These relationships are summarized in a model, which can then be applied to a different data set in which the class assignments are unknown.

Classification models are tested by comparing the predicted values to known target values in a set of test data. The historical data for a classification project is typically divided into two data sets: one for building the model; the other for testing the model.



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Fig. 1.1.1 Classification example

1.1.2 Clustering

Clustering analysis finds clusters of data objects that are similar in some sense to one another. The members of a cluster are more like each other than they are like members of other clusters. The goal of clustering analysis is to find high-quality clusters such that the inter-cluster similarity is low and the intra-cluster similarity is high.

Clustering, like classification, is used to segment the data. Unlike classification, clustering models segment data into groups that were not previously defined. Classification models segment data by assigning it to previously-defined classes, which are specified in a target. Clustering models do not use a target.

Clustering is useful for exploring data. If there are many cases and no obvious groupings, clustering algorithms can be used to find natural groupings. Clustering can also serve as a useful data pre-processing step to identify homogeneous groups on which to build supervised models.

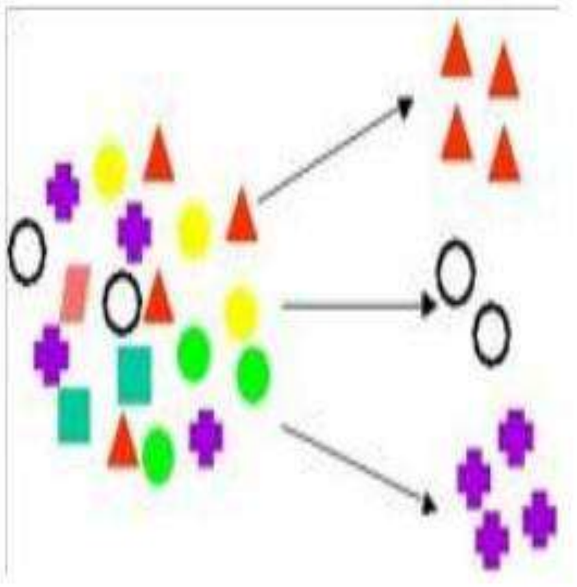


Fig.1.1.2 Clustering example

1.2 Objective

The main objective of the project is as follows: To study the implementation of Decision Tree Algorithm and Clustering Algorithm to obtain risk levels caused by cancer.

The sub objectives formulated are as follows:

1. Creating a Web Page using Java.
2. Creating Database through MySQL.
3. To classify the data and to mine frequent patterns in dataset Decision Tree Algorithm is used
4. To cluster the data obtained from the Decision Tree Algorithm using K-means Algorithm.
5. Providing report to cancer effected people according to their risk levels.

1.3 Project Definition

Cancer is one of the leading causes for death world-wide. However, early detection of cancer plays an important role in reducing deaths. Relationship between a person's life style and the probability of he/she being affected with cancer can be established by using data mining techniques. In this project, we use those data mining techniques and help people to predict if they have cancer as soon as they recognize the changes in their body.

The data mining techniques we use here are Classification and Clustering.

1.4 Organization of Project Report

User module

User can login as old user or new user.

New user can fill details in the entry form and user id will be generated.

Old user can check his/her details using user id.

Report

Risk score is calculated based on his/her details in the entry form. Along with that diagnosis test is suggested, cancer and type of cancer is also predicted

Clustering

Based on the risk score of the patient, each instance is clustered into different types of cancer clusters using k-means algorithm.

Decision tree

Based on risk scores and weights, significant patterns are generated from decision tree using ID-3 algorithm.

2.LITERATURE SURVEY

2.1 Existing System and its Drawbacks:

Cancer can be diagnosed with the help of Screening and Diagnostic tests. These tests are useful in checking if the person has cancer. Screening tests aim at finding the disease even before the symptoms become noticeable. But, there are some disadvantages of these methods.

- Screening tests include X-Ray scanning and thus, the patient gets exposed to radiations and their harmful side effects.
- These tests are costly and not accessible to everyone.
- This process is more time consuming.

2.2 Motivation for Proposed System:

Thus, we use a computerized system to predict if a person has cancer based on his/her personal information, habits and the symptoms noticed. The benefits of using this system are:

- Any user can get access to this system from any remote location.
- Time taken to get the report is very less. Thus, using this system, a user can submit his details and instantly could know if he has cancer.
- Managing the data is easy too.
- The system becomes more intelligent as the number of people entering the data increases.

- A report is generated by the system which suggests the lab tests to be done based on the type of cancer the user is predicted with. The user can use this information and confirm if he has cancer by undergoing those suggested tests.

3. REQUIREMENT SPECIFICATION

3.1 Proposed System

The following is the model of the proposed work. The collected data is pre-processed and stored in the knowledge base to build the model. Seventy five percent of the entire data is taken as training set to build the classification and clustering model the remaining of which is taken for testing purpose. The decision tree model is build using the classification rules, the significant frequent pattern and its corresponding weightage. The clustering model is build using the k-means clustering algorithm. The model is then tested for accuracy, sensitivity and specificity using test data along with merging it to the knowledge base.

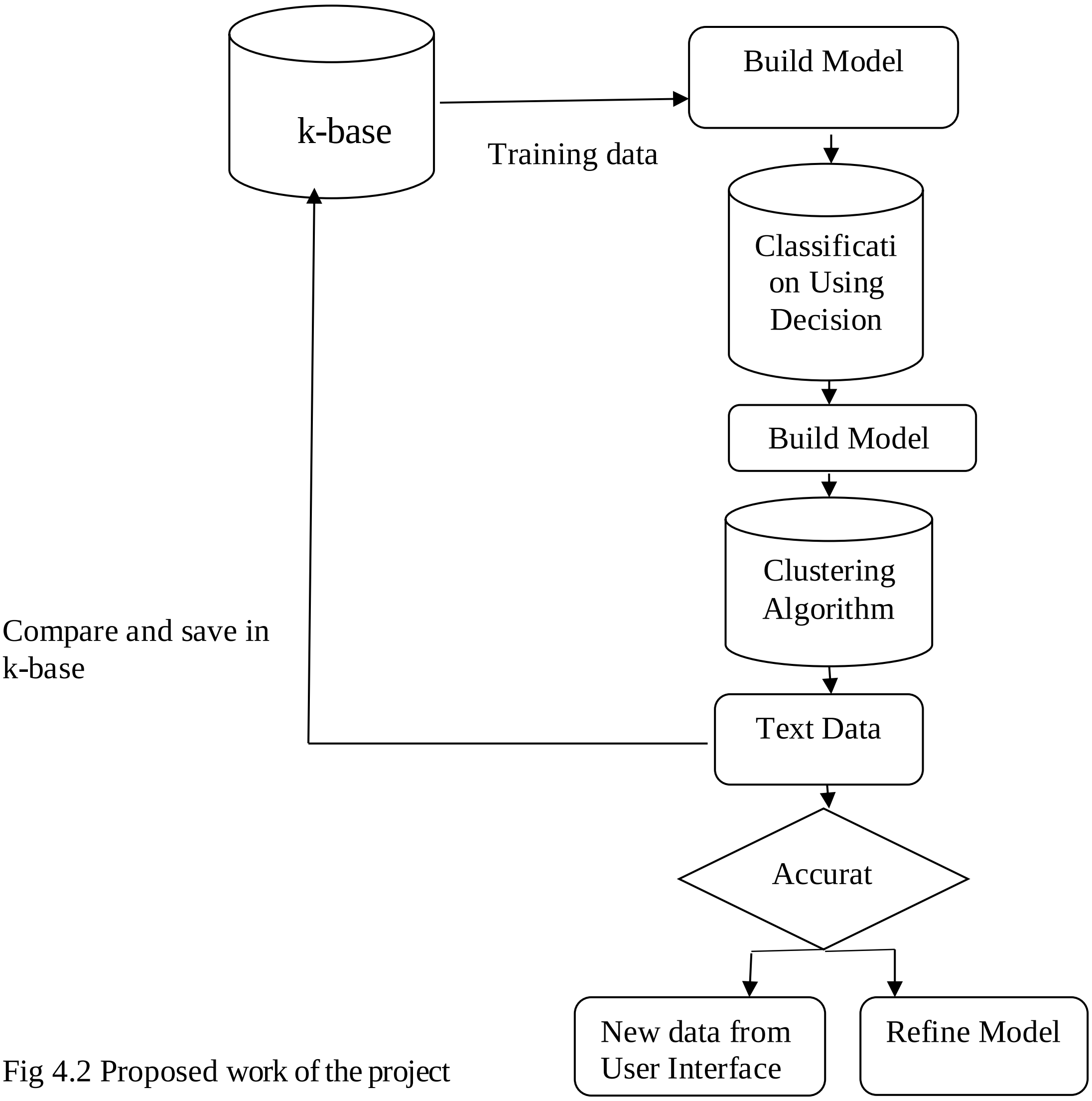


Fig 4.2 Proposed work of the project

3.2 Overall Description of Project

Use case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

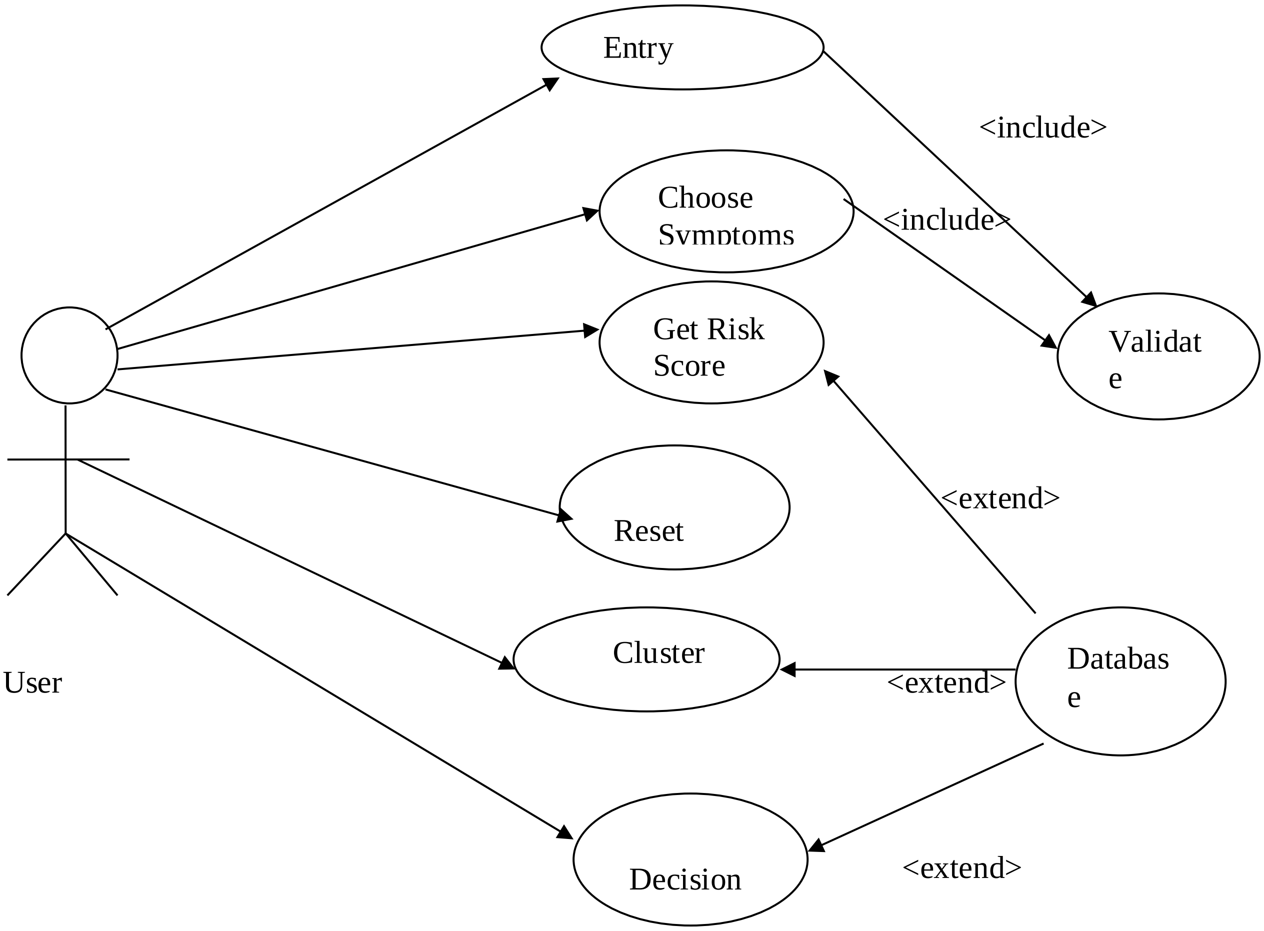


Fig 3.2.1 Use case diagram

Activity Diagram

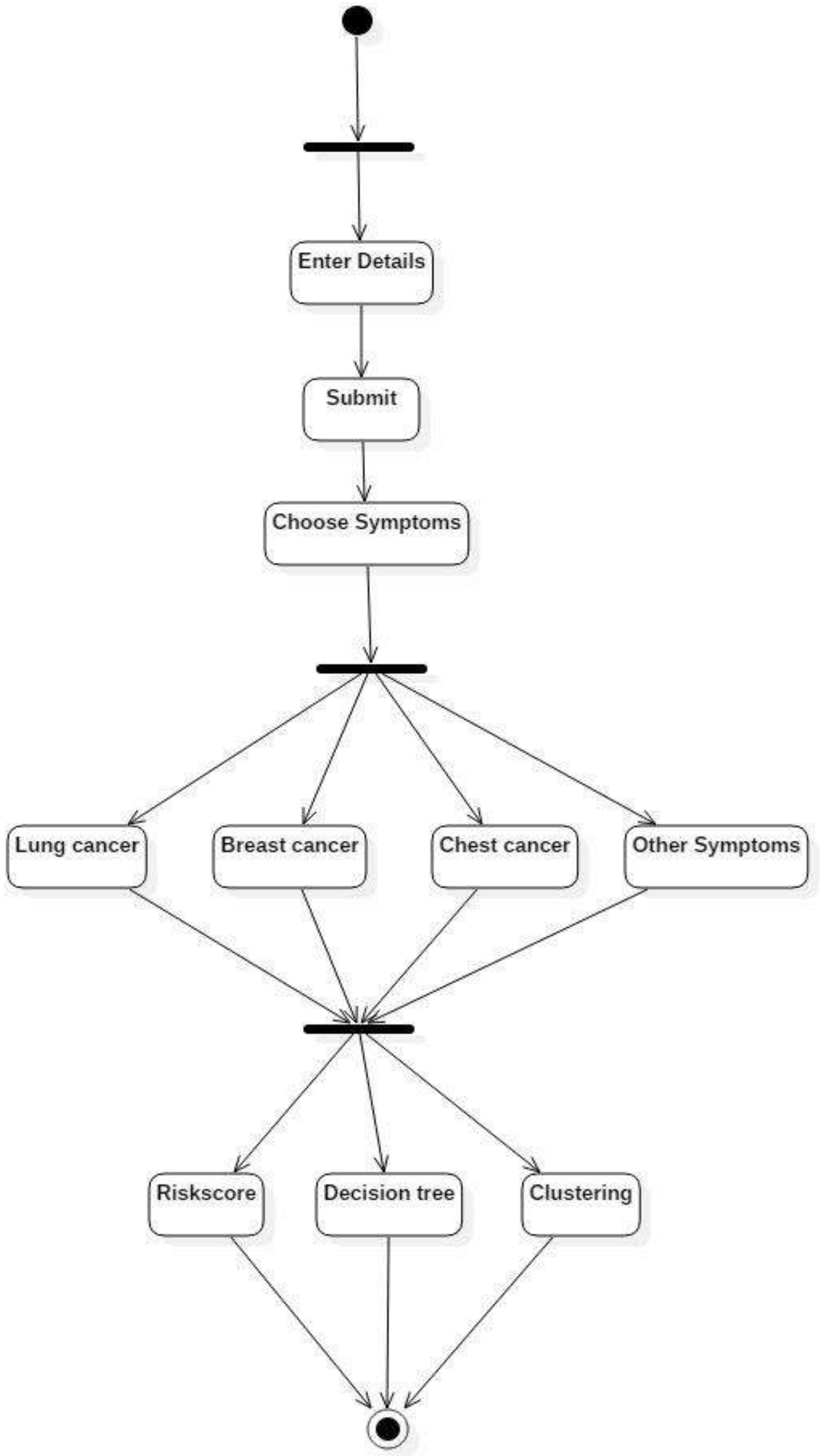


Fig 3.2.2 Activity Diagram for User Interface

3.3Functional and Non-Functional Requirements:

Functional Requirements

1.Login Page

Field Name	Field Description	Field Type	Field Business Rule
Old user	Leads to report page	Button	Report is displayed
New user	Leads to entry form page	Button	Authenticated login is provided

2.Entry Form Page

Field Name	Field Description	Field Type	Field Business Rule
Name	Text in the field is considered as a login credential	TextField	Generic in format
Age	User age is to be entered in field	TextField	Only numerals are to be entered
Gender	Select the respective item in list	Drop down list	Select an item from the list provided ones
Marital Status	Select the respective item in list	Drop down list	Select an item from the list provided ones
Children	Select the respective item in list	Drop down list	Select an item from the list provided ones
Education	Select the respective item in list	Drop down list	Select an item from the list provided ones
Occupational hazards	Select the respective item in list	Drop down list	Select an item from the list provided ones
Smoking	Select the respective item in list	Drop down list	Select an item from the list provided ones
Alcohol	Select the respective item in list	Drop down list	Select an item from the list provided ones

Chewing	Select the respective item in list	Drop down list	Select an item from the list provided ones
Hot beverages	Select the respective item in list	Drop down list	Select an item from the list provided ones
Diet	Select the respective item in list	Drop down list	Select an item from the list provided ones
Fast food addiction	Select the respective item in list	Drop down list	Select an item from the list provided ones
Family history of cancer	Select the respective item in list	Drop down list	Select an item from the list provided ones
Relation	Select the respective item in list	Drop down list	Select an item from the list provided ones
Weight loss	Select the respective item in list	Drop down list	Select an item from the list provided ones
Submit	Submit the details in the form to system	Submit button	Submit form
Symptoms	Select the respective item in list	Drop down list	Select an item from the list provided ones

Non-Functional Requirements

1.Usability

The system is designed to be easily understood and usable by the users.

2.Reliability

The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using java is more reliable.

3.Performance

This system is developed in the high-level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

4.Supportability

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platforms, which are having JVM, built into the system.

5.Implementation

The system is implemented using java applet. The Net beans is used as server and windows 8.1 is used as the platform.

3.4 Design Specification (Class Diagram/E-R Diagram):

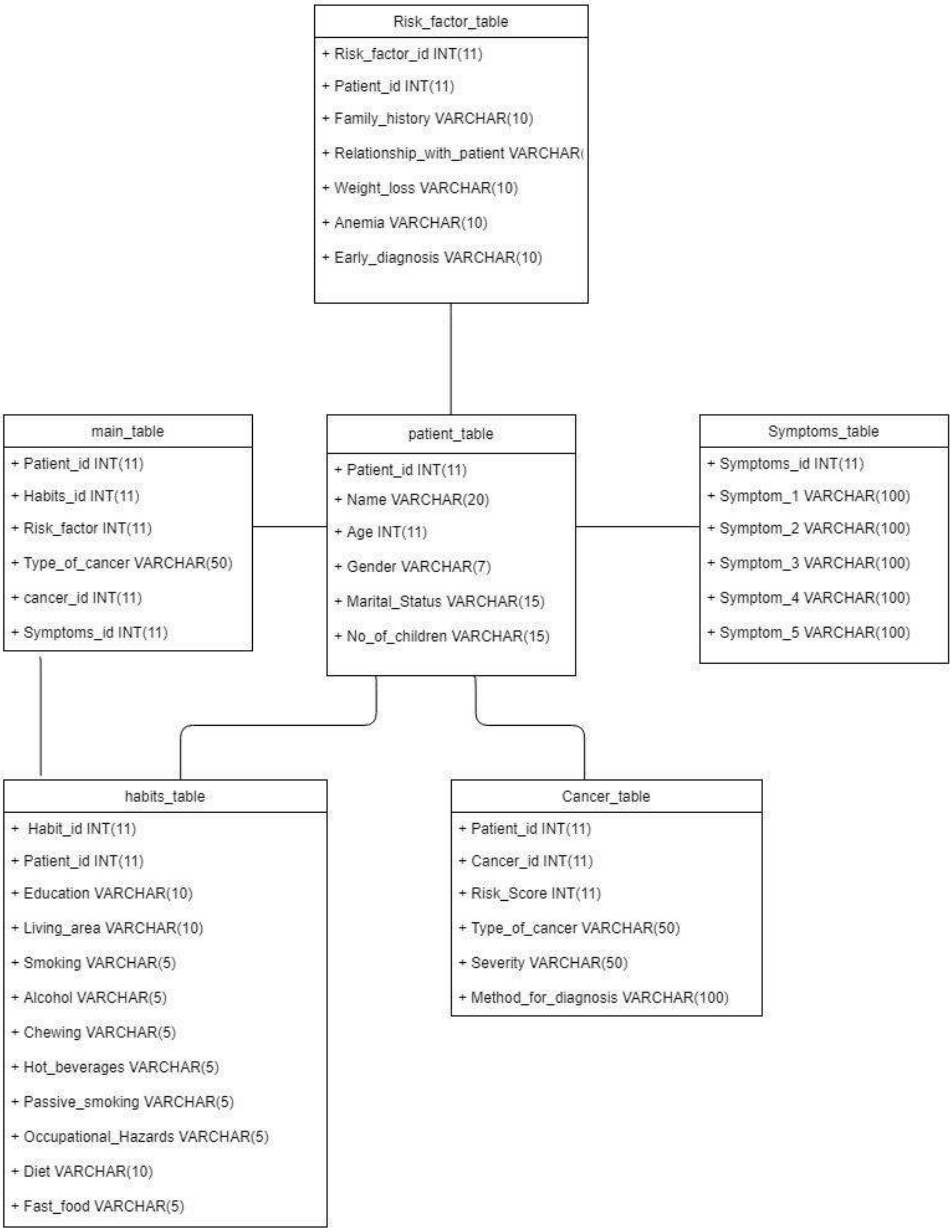


Figure.3.4.1 class diagram

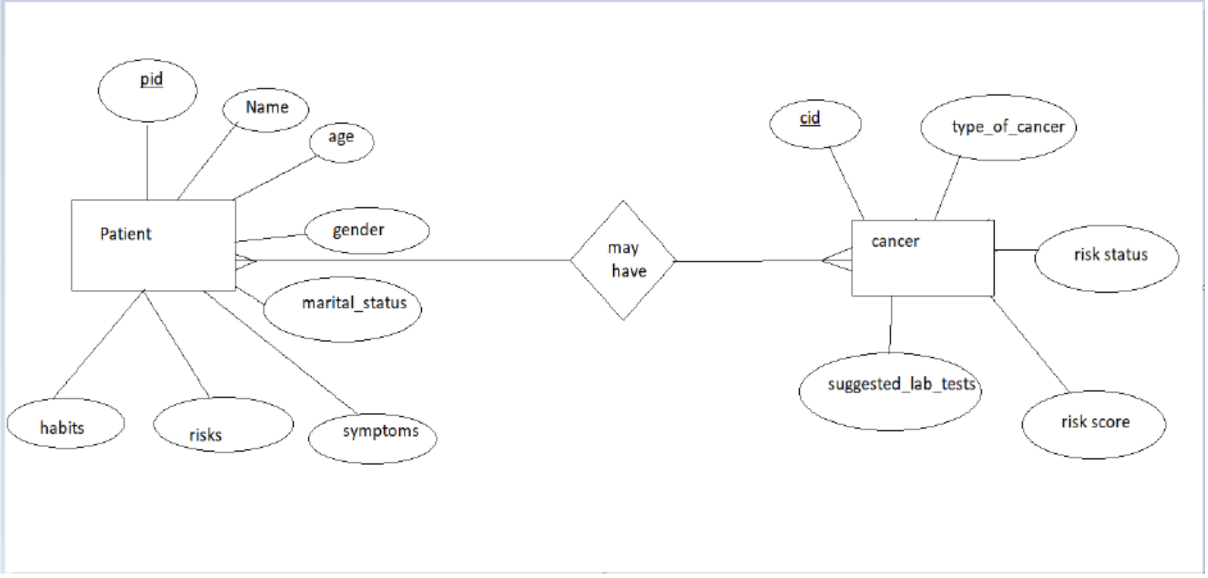


Fig. 3.4.2 E-R Diagram

3.5 Software Requirements and Hardware Requirements:

Software Requirements:

Platform	- Windows
Front End	- Java Applets
Development Tool	- NetBeans IDE
Back End	- Java, MySQL

Hardware Requirements:

Name of the Processor	- I3
Hard Disk Capacity	- 1TB
RAM Capacity	- 4GB

Development Environment:

1.NetBeans

NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended .The NetBeans Platform is a framework for simplifying the development of Java Swing desktop applications. Application's can install modules dynamically. Any application can include the Update centre module to allow users of the application to download digitally signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

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)
- NetBeans Visual Library
- Integrated development tools

A jFrame Form acts as a container to place other components like the button, text field and text area etc. We can create as many forms we want in a Netbeans Project. A jFrame containing jLabel, jTextField and jButton.

File Handling

Filesystem is used to control how data is stored and retrieved. File handling is used to read, write, append or update a file without directly opening it.

Types of File

1. Text File
2. Binary File

Text File contains only textual data. Text files may be saved in either a plain text (TXT) format and rich text (.RTF) format like files in our Notepad while Binary Files contains both textual data and custom binary data like font size, text color and text style etc. After the termination of program all the entered data is lost because primary memory is volatile. If the data has to be used later, then it becomes necessary to keep it in permanent storage device. So the concept of file through which data can be stored on the disk or secondary storage device. The stored data can be read whenever required.

2.Lists

The `java.util.List` interface is a subtype of the `java.util.Collection` interface. It represents an ordered list of objects, meaning you can access the elements of a `List` in a specific order, and by an index too.We can also add the same element more than once to a `List`.

List Implementations

Being a Collection subtype all methods in the Collection interface are also available in the List interface.

Since List is an interface we need to instantiate a concrete implementation of the interface in order to use it. we can choose between the following List implementations in the Java Collections API:

1. java.util.ArrayList
2. java.util.LinkedList
3. java.util.Vector
4. java.util.Stack

There are also List implementations in the java.util.concurrent package

Syntax;

```
List listA = new ArrayList();
List listB = new LinkedList();
List listC = newVector();
List listD = newStack();
```

Adding and Accessing Elements

To add elements to a `List` you call it's `add()` method. This method is inherited from the `Collection` interface

```
listA.add ("element 1");
listA.add (0, "element 0");
```

The first `add()` calls add a `String` instance to the end of the list. The last `add()` call adds a `String` at index 0, meaning at the beginning of the list.

The order in which the elements are added to the `List` is stored, so we can access the elements in the same order. We can do so using either the `get (int index)` method, or via the `Iterator` returned by the `iterator()` method.

Removing Elements

We can remove elements in two ways:

1. remove(Object element)
2. remove(int index)

Remove (Object element) removes that element in the list, if it is present. All subsequent elements in the list are then moved up in the list. Their index thus decreases by 1.

Remove (int index) removes the element at the given index. All subsequent elements in the list are then moved up in the list. Their index thus decreases by 1.

Clearing a List

The Java List interface contains a clear() method which removes all elements from the list when called. Here is simple example of clearing a List with clear():

list.clear();

List Size

We can obtain the number of elements in the `List` by calling the `size()` method**.**

int size = list.size();

Generic Lists

By default we can put any `Object` into a `List`, but from Java 5, Java Generics makes it possible to limit the types of object you can insert into a `List`. Here is an example:

List<MyObject> list = new ArrayList<MyObject>();

This `List` can now only have `MyObject` instances inserted into it. We can then access and iterate its elements without casting them

Hashmap

Java HashMap class implements the map interface by using a hashtable. It inherits AbstractMap class and implements Map interface.

The important points about Java HashMap class are:

- A HashMap contains values based on the key.
- It contains only unique elements.
- It may have one null key and multiple null values.
- It maintains no order.

Hierarchy of HashMap class

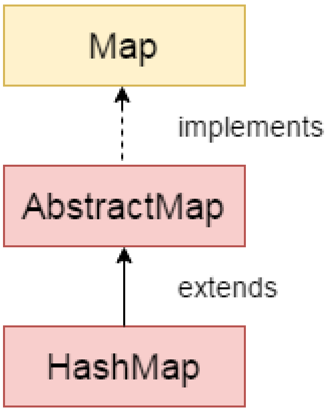


Fig.3.5.2HashMap hierarchy

HashMap class Parameters

1. **K**: It is the type of keys maintained by this map.
2. **V**: It is the type of mapped values.

Constructors of Java HashMap class

Constructor	**Description**

HashMap()	It is used to construct a default HashMap.
HashMap(Map m)	It is used to initializes the hash map by using the elements of the given Map object m.
HashMap(int capacity)	It is used to initializes the capacity of the hash map to the given integer value, capacity.

Methods of Java HashMap class

Method	**Description**
void clear()	It is used to remove all of the mappings from this map.
booleancontainsKey(Object key)	It is used to return true if this map contains a mapping for the specified key.
booleancontainsValue(Object value)	It is used to return true if this map maps one or more keys to the specified value.
booleanisEmpty()	It is used to return true if this map contains no key-value mappings.
Object clone()	It is used to return a shallow copy of this HashMap instance: the keys and values themselves are not cloned.
Set entrySet()	It is used to return a collection view of the mappings contained in this map.
Set keySet()	It is used to return a set view of the keys contained in this map.
Object put(Object key, Object value)	It is used to associate the specified value with the specified key in this map.
intsize()	It is used to return the number of key-value mappings in this map.

3. MySQL

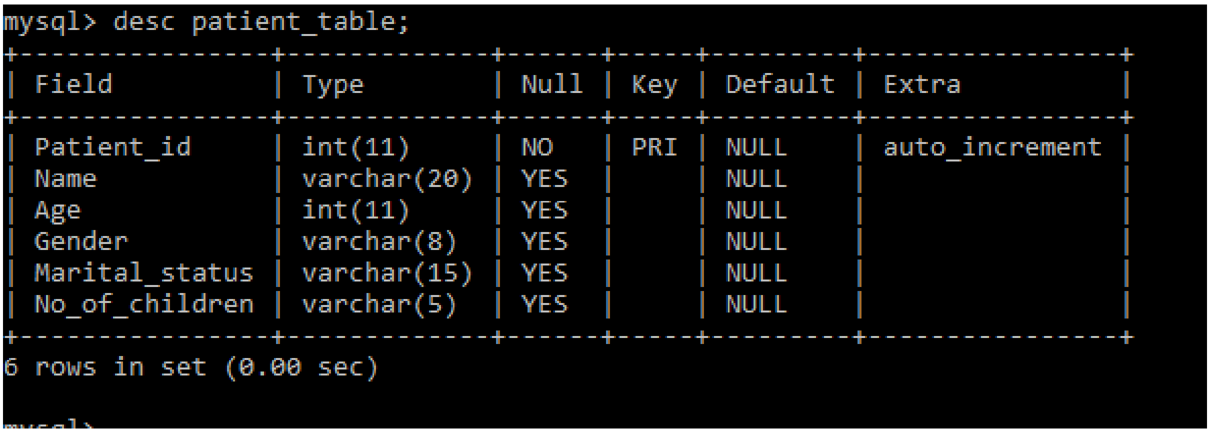
A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds. Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. It uses a standard form of the well-known SQL data language. It works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. It works very quickly and works well even with large data set.

DATABASE

In this project, database includes six tables.

They are

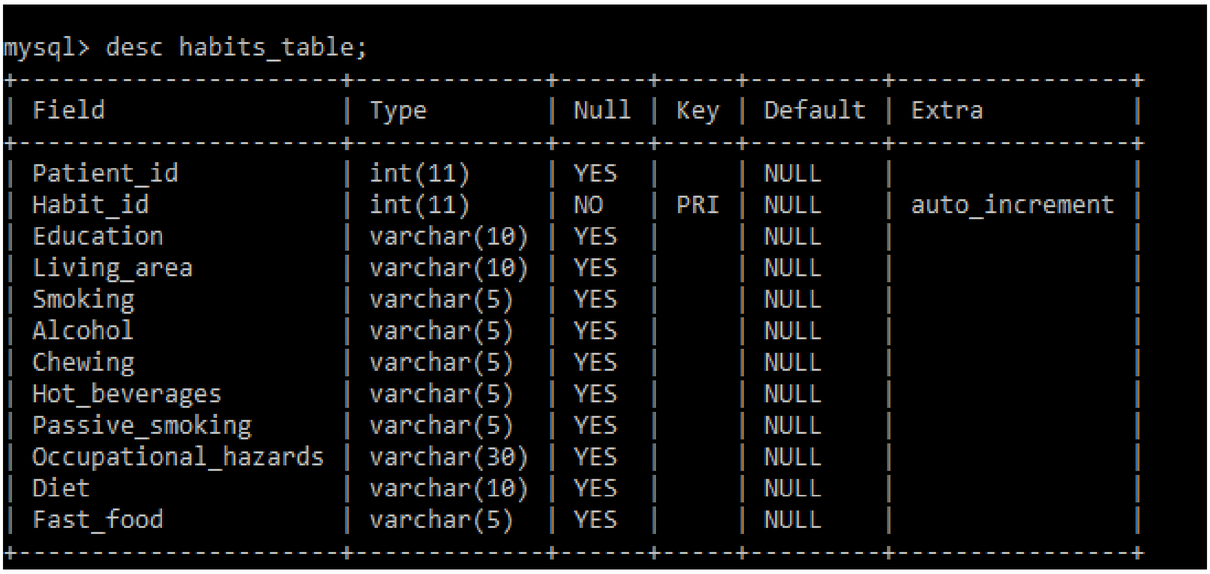
1.Patient table which includes patient information.



```
mysql> desc patient_table;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| Patient_id     | int(11)     | NO   | PRI | NULL    | auto_increment |
| Name           | varchar(20) | YES  |     | NULL    |                |
| Age            | int(11)     | YES  |     | NULL    |                |
| Gender         | varchar(8)  | YES  |     | NULL    |                |
| Marital_status | varchar(15) | YES  |     | NULL    |                |
| No_of_children | varchar(5)  | YES  |     | NULL    |                |
+----------------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

Table.1. patient_table

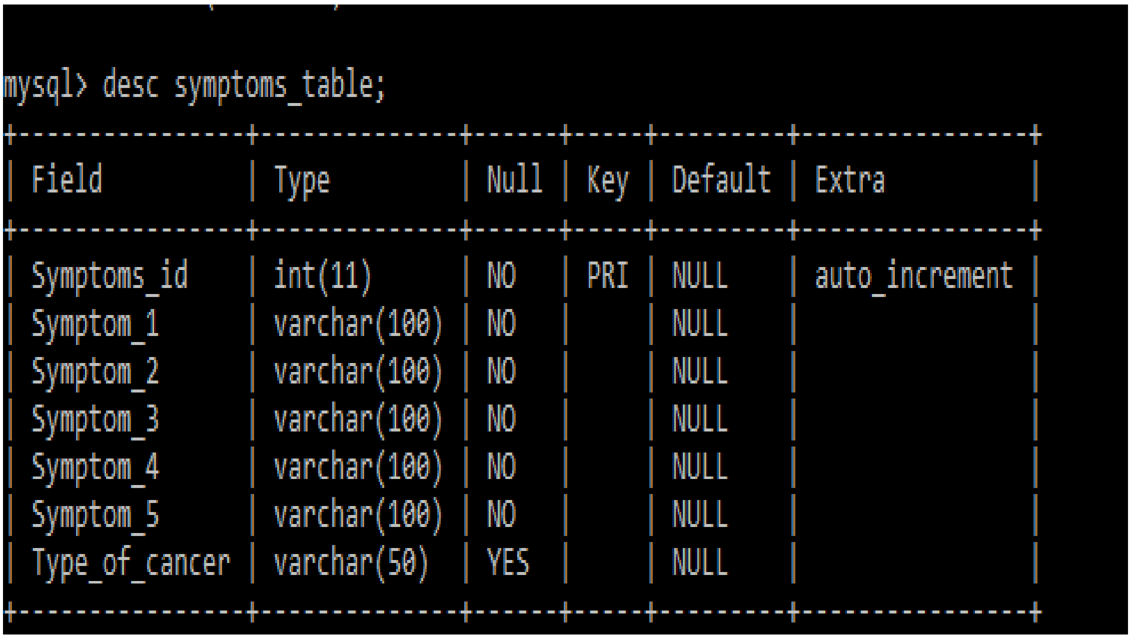
2. Habits table which includes several habits of a person



```
mysql> desc habits_table;
+---------------------+-------------+------+-----+---------+----------------+
| Field               | Type        | Null | Key | Default | Extra          |
+---------------------+-------------+------+-----+---------+----------------+
| Patient_id          | int(11)     | YES  |     | NULL    |                |
| Habit_id            | int(11)     | NO   | PRI | NULL    | auto_increment |
| Education           | varchar(10) | YES  |     | NULL    |                |
| Living_area         | varchar(10) | YES  |     | NULL    |                |
| Smoking             | varchar(5)  | YES  |     | NULL    |                |
| Alcohol             | varchar(5)  | YES  |     | NULL    |                |
| Chewing             | varchar(5)  | YES  |     | NULL    |                |
| Hot_beverages       | varchar(5)  | YES  |     | NULL    |                |
| Passive_smoking     | varchar(5)  | YES  |     | NULL    |                |
| Occupational_hazards| varchar(30) | YES  |     | NULL    |                |
| Diet                | varchar(10) | YES  |     | NULL    |                |
| Fast_food           | varchar(5)  | YES  |     | NULL    |                |
+---------------------+-------------+------+-----+---------+----------------+
```

Table.2. habits_table

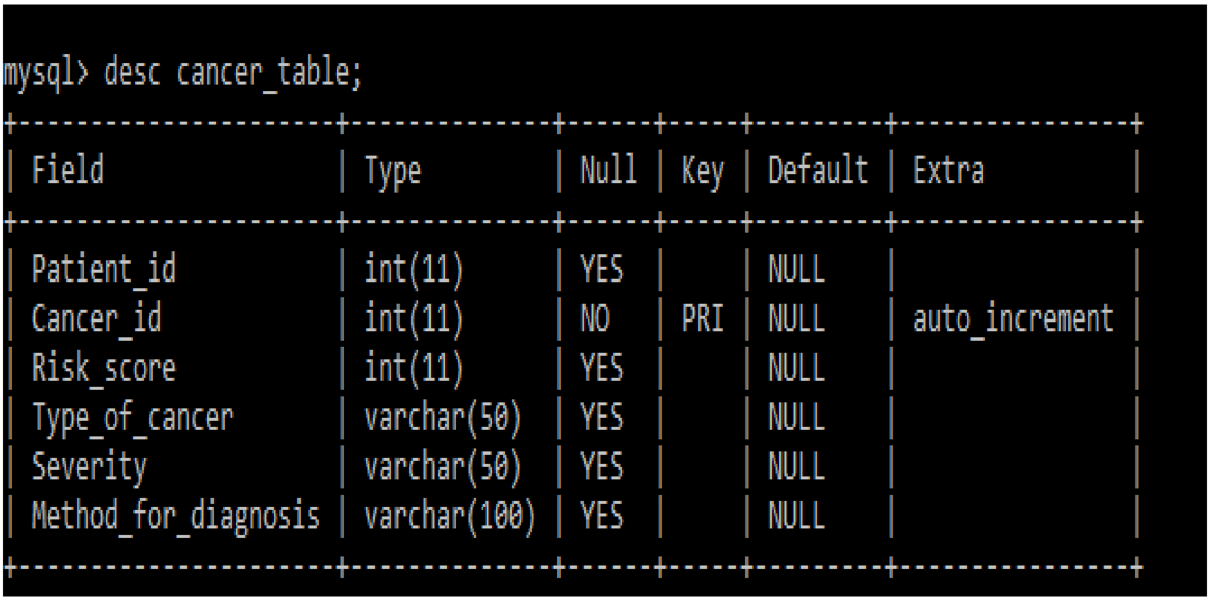
3.Symptoms table which includes various symptoms a person may have.



```
mysql> desc symptoms_table;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| Symptoms_id    | int(11)      | NO   | PRI | NULL    | auto_increment |
| Symptom_1      | varchar(100) | NO   |     | NULL    |                |
| Symptom_2      | varchar(100) | NO   |     | NULL    |                |
| Symptom_3      | varchar(100) | NO   |     | NULL    |                |
| Symptom_4      | varchar(100) | NO   |     | NULL    |                |
| Symptom_5      | varchar(100) | NO   |     | NULL    |                |
| Type_of_cancer | varchar(50)  | YES  |     | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
```

Table.3. symptoms_table

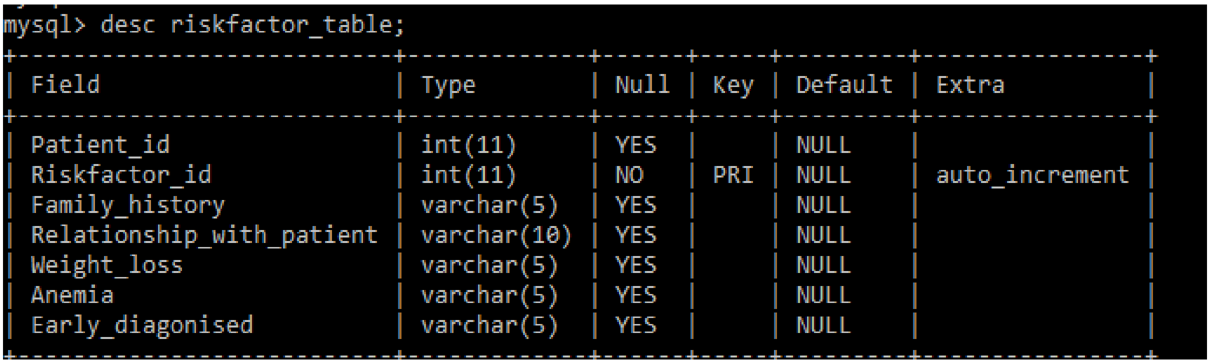
4. Cancer table which includes types of cancer and shown to person based on his/her symptoms.



```
mysql> desc cancer_table;
+---------------------+-------------+------+-----+---------+----------------+
| Field               | Type        | Null | Key | Default | Extra          |
+---------------------+-------------+------+-----+---------+----------------+
| Patient_id          | int(11)     | YES  |     | NULL    |                |
| Cancer_id           | int(11)     | NO   | PRI | NULL    | auto_increment |
| Risk_score          | int(11)     | YES  |     | NULL    |                |
| Type_of_cancer      | varchar(50) | YES  |     | NULL    |                |
| Severity            | varchar(50) | YES  |     | NULL    |                |
| Method_for_diagnosis | varchar(100) | YES |     | NULL    |                |
+---------------------+-------------+------+-----+---------+----------------+
```

4.Table. cancer_table

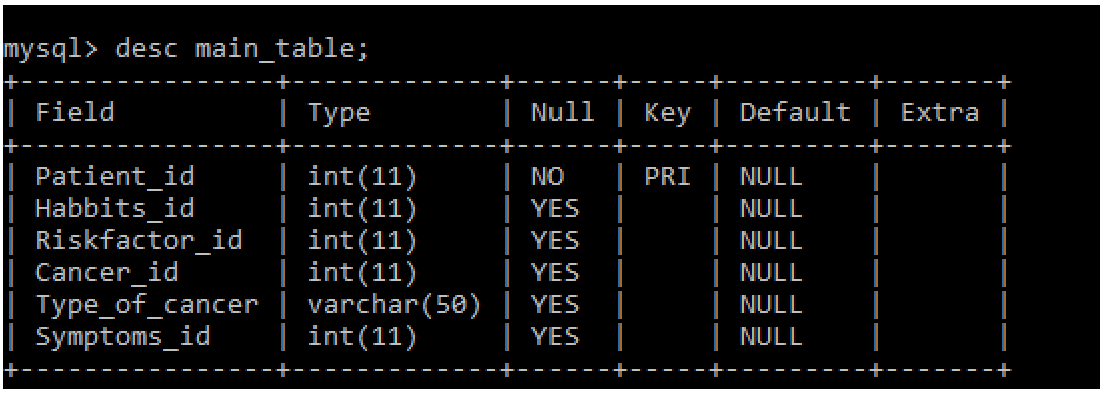
5. Risk factor table which includes calculated risk scores for every attribute that is been assigned.



```
mysql> desc riskfactor_table;
+--------------------------+-------------+------+-----+---------+----------------+
| Field                    | Type        | Null | Key | Default | Extra          |
+--------------------------+-------------+------+-----+---------+----------------+
| Patient_id               | int(11)     | YES  |     | NULL    |                |
| Riskfactor_id            | int(11)     | NO   | PRI | NULL    | auto_increment |
| Family_history           | varchar(5)  | YES  |     | NULL    |                |
| Relationship_with_patient | varchar(10) | YES  |     | NULL    |                |
| Weight_loss              | varchar(5)  | YES  |     | NULL    |                |
| Anemia                   | varchar(5)  | YES  |     | NULL    |                |
| Early_diagonised         | varchar(5)  | YES  |     | NULL    |                |
+--------------------------+-------------+------+-----+---------+----------------+
```

5.Table. riskfactor_table

6. Main table that includes all the required information that to be shown to the user



```
mysql> desc main_table;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| Patient_id     | int(11)     | NO   | PRI | NULL    |       |
| Habbits_id     | int(11)     | YES  |     | NULL    |       |
| Riskfactor_id  | int(11)     | YES  |     | NULL    |       |
| Cancer_id      | int(11)     | YES  |     | NULL    |       |
| Type_of_cancer | varchar(50) | YES  |     | NULL    |       |
| Symptoms_id    | int(11)     | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
```

6.Table. main_table

4. IMPLEMENTATION

4.1 Methodology

Decision Tree Algorithm

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making). This page deals with decision trees in data mining. Each element of the domain of the classification is called a class. A decision tree or a classification tree is a tree in which each internal (non-leaf) node is labelled with an input feature. The arcs coming from a node labeled with an input feature are labeled with each of the possible values of the target or output feature or the arc leads to a subordinate decision node on a different input feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes.

A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning.

The recursion is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the prediction.

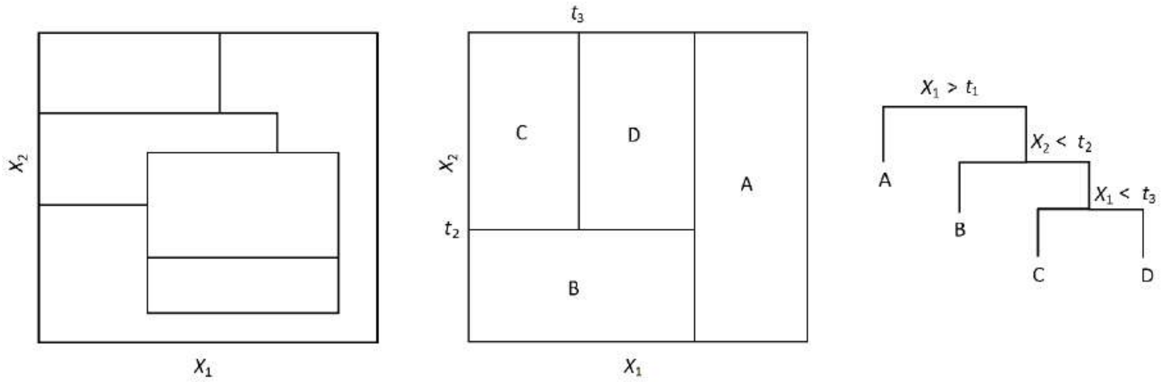


Fig. 4.1 Decision Tree Example

Left: A partitioned two-dimensional feature space. These partitions could not have resulted from recursive binary splitting.

Middle: A partitioned two-dimensional feature space with partitions that did result from recursive binary splitting.

Right: A tree corresponding to the partitioned feature space in the middle. Notice the convention that when the expression at the split is true, the tree follows the left branch. When the expression is false, the right branch is followed.

In data mining, decision trees can be described also as the combination of mathematical and computational techniques to aid the description, categorization and generalization of a given set of data.

Data comes in records of the form:

(x,Y)=(x1, x2, x3, x4…..xk, Y)

The dependent variable, Y, is the target variable that we are trying to understand, classify or generalize. The vector **x** is composed of the input variables, x_1, x_2, x_3 etc., that are used for that task.

Information gain

Information gain is based on the concept of entropy from information theory.

Information Gain = Entropy (parent) - Weighted Sum of Entropy (Children)
Information gain is used to decide which feature to split on at each step in building the tree. Simplicity is best, so we want to keep our tree small. To do so, at each step we should choose the split that results

in the purest daughter nodes. A commonly used measure of purity is called information which is measured in bits, not to be confused with the unit of computer memory. For each node of the tree, the information value "represents the expected amount of information that would be needed to specify whether a new instance should be classified yes or no "A data set with some attributes and to construct a decision tree on this data, we need to compare the information gain of each of these trees, each split on one of the these features. The split with the highest information gain will be taken as the first split and the process will continue until all children nodes are pure, or until the information gain is 0.

Clustering

Clustering in pattern recognition is the process of partitioning a set of pattern vectors into subsets called clusters. Clustering is an unsupervised method for dividing data into disjoint subsets with high intra-cluster similarity and low inter-cluster similarity. Over the past decades, many clustering algorithms have been proposed, including k-means clustering, mixture models, spectral clustering ,and locality-sensitive hashing and maximum margin clustering. Most of these approaches perform hard clustering, that is, they assign each item to a single cluster. This works well when clustering compact and data is well- seperated and grouped.

K-Means Clustering

This algorithm is guaranteed to coverage, but it may not return the optimal solution. The quality of the solution depends on the initial set of clusters and the value of K.K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain numbers of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending the first step is completed and an early group page is done. At this point we need to re-calculate k new centroids as barycentres of clusters resulting from the previous step. After we have these k means centroids, a new binding has to be done between the same data set points and nearest new centroid .A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done .In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function.

4.2 System Architecture

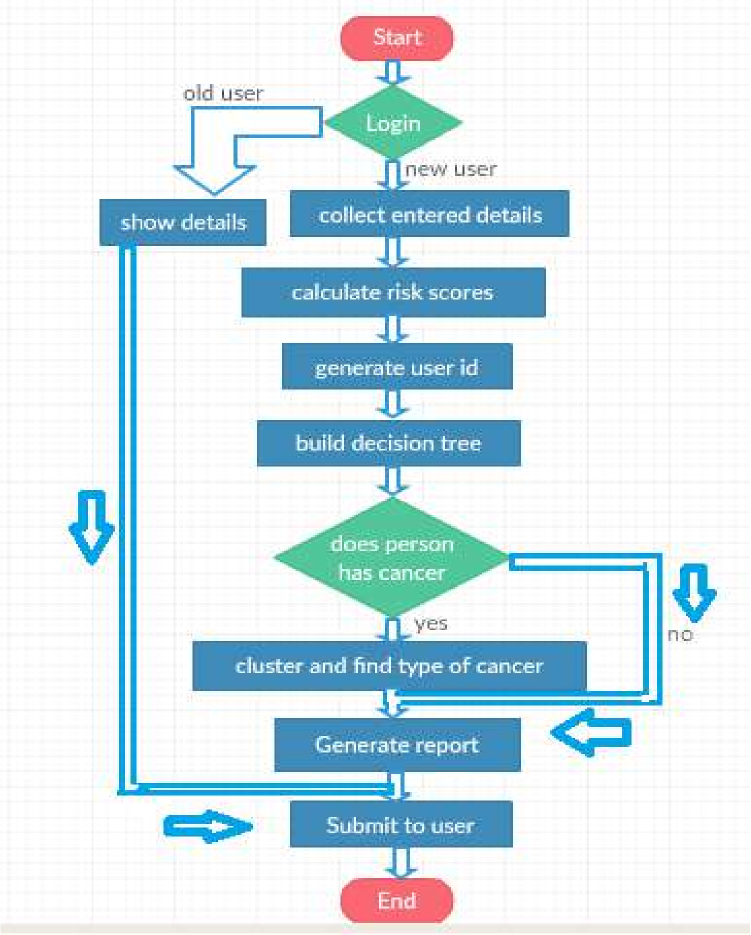


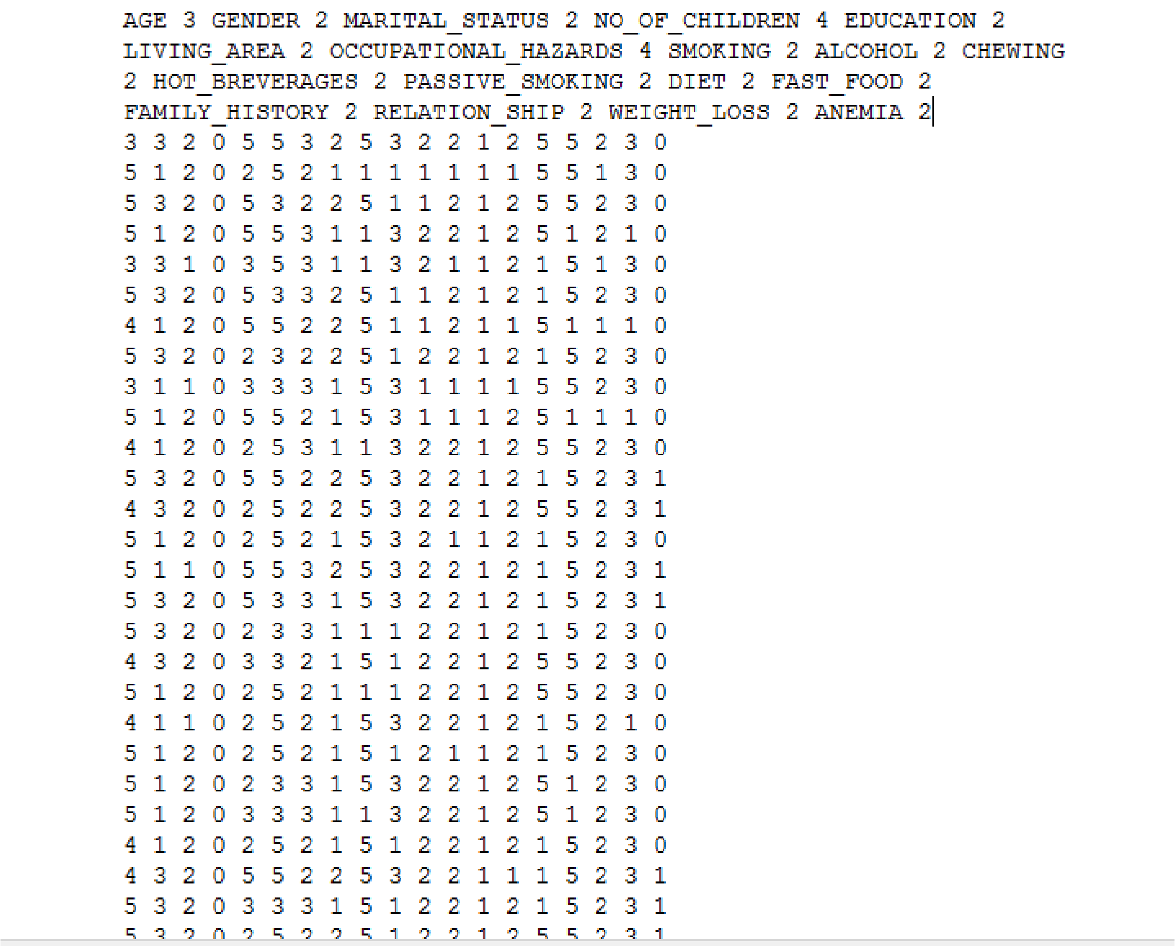
Fig 4.2 Flow chart diagram

4.3 Material and Methods

Extensive literature reviews, case studies and discussions with medical experts show that there are number of factors influenc0ing cancer. These factors are identified and taken as attributes for this project.

4.3.1 Data Source

The data for this study was collected from various websites, consisting of cancer and non cancer patients data and they are preprocessed to suit this project This data consists of more than 18 attributes such as Age, Marital status, Symptoms relating to cancer, occupational hazards, family history of cancer etc. These attributes are used to train and develop the system and a part is used to test the significance of the system. These attributes play an important role in diagnosing cancer in all the cases. This data is stored in a knowledge base which has the ability to expand itself as new data enters the system through front end from which new knowledge is gained and thus the system becomes intelligent.



```
AGE 3 GENDER 2 MARITAL_STATUS 2 NO_OF_CHILDREN 4 EDUCATION 2
LIVING_AREA 2 OCCUPATIONAL_HAZARDS 4 SMOKING 2 ALCOHOL 2 CHEWING
2 HOT_BREVERAGES 2 PASSIVE_SMOKING 2 DIET 2 FAST_FOOD 2
FAMILY_HISTORY 2 RELATION_SHIP 2 WEIGHT_LOSS 2 ANEMIA 2
3 3 2 0 5 5 3 2 5 3 2 2 1 2 5 5 2 3 0
5 1 2 0 2 5 2 1 1 1 1 1 1 1 5 5 1 3 0
5 3 2 0 5 3 2 2 5 1 1 2 1 2 5 5 2 3 0
5 1 2 0 5 5 3 1 1 3 2 2 1 2 5 1 2 1 0
3 3 1 0 3 5 3 1 1 3 2 1 1 2 1 5 1 3 0
5 3 2 0 5 3 3 2 5 1 1 2 1 2 1 5 2 3 0
4 1 2 0 5 5 2 2 5 1 1 2 1 1 5 1 1 1 0
5 3 2 0 2 3 2 2 5 1 2 2 1 2 1 5 2 3 0
3 1 1 0 3 3 3 1 5 3 1 1 1 1 5 5 2 3 0
5 1 2 0 5 5 2 1 5 3 1 1 1 2 5 1 1 1 0
4 1 2 0 2 5 3 1 1 3 2 2 1 2 5 5 2 3 0
5 3 2 0 5 5 2 2 5 3 2 2 1 2 1 5 2 3 1
4 3 2 0 2 5 2 2 5 3 2 2 1 2 5 5 2 3 1
5 1 2 0 2 5 2 1 5 3 2 1 1 2 1 5 2 3 0
5 1 1 0 5 5 3 2 5 3 2 2 1 2 1 5 2 3 1
5 3 2 0 5 3 3 1 5 3 2 2 1 2 1 5 2 3 1
5 3 2 0 2 3 3 1 1 1 2 2 1 2 1 5 2 3 0
4 3 2 0 3 3 2 1 5 1 2 2 1 2 5 5 2 3 0
5 1 2 0 2 5 2 1 1 1 2 2 1 2 5 5 2 3 0
4 1 1 0 2 5 2 1 5 3 2 2 1 2 1 5 2 1 0
5 1 2 0 2 5 2 1 5 1 2 1 1 2 1 5 2 3 0
5 1 2 0 2 3 3 1 5 3 2 2 1 2 5 1 2 3 0
5 1 2 0 3 3 3 1 1 3 2 2 1 2 5 1 2 3 0
4 1 2 0 2 5 2 1 5 1 2 2 1 2 1 5 2 3 0
4 3 2 0 5 5 2 2 5 3 2 2 1 1 1 5 2 3 1
5 3 2 0 3 3 3 1 5 1 2 2 1 2 1 5 2 3 1
5 3 2 0 2 5 2 2 5 1 2 2 1 2 5 5 2 3 1
```

Fig.4.3.1 Data Set with attributes

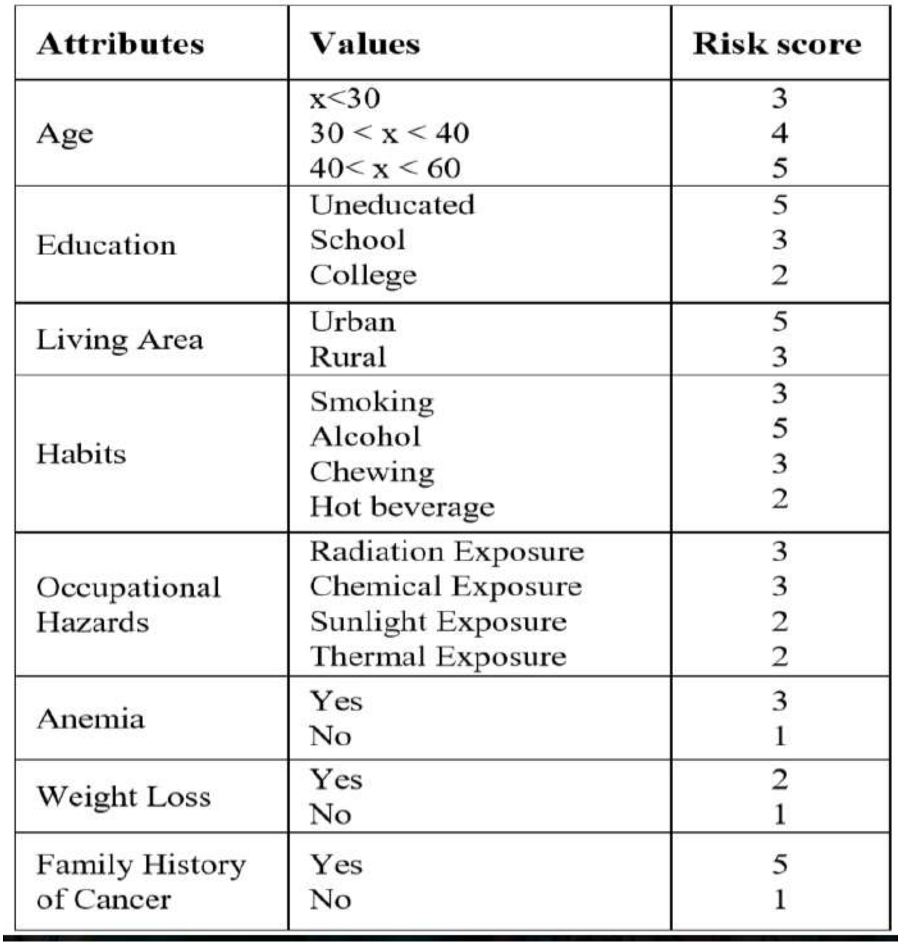
4.3.2 Classification and Significant Pattern Generation

Decision tree algorithm is used to mine frequent patterns from the data set. The frequent item sets that occur throughout the data base and have a significant link to cancer status are mined as significant patterns. The data is fed into the decision tree algorithm to obtain the significant patterns related to cancer and non cancer data sets. In other words the patterns that are mined by the decision tree are well defined and

distinguished to be separated as cancer and non cancer datasets. The following pseudo code is used to generate frequent pattern using decision tree.

Pseudo Code:

```
ID3 {Examples, Target_Attribute, Attributes}
Create a root node for the tree.
If all examples are positive, Return the single-node tree root, with label = +.
If all examples are negative, Return the single-node Root, with label= -.
If number of predicting attributes is empty, then return the single node tree Root,
With label = most common value of the target attribute in the example,
Otherwise Begin
  A = The Attribute the best classifies examples,
  Decision Tree Attribute for Root = A,
  For each possible value, Vi, of A,
      Add a new tree branch below Root, corresponding to the test A =vi,
      Let examples (Vi) be the subset of examples that have the value Vi for A
      If examples (Vi) is empty
          Then below this new branch add a leaf node with label = most common target value
In the examples
            Else below this new branch add the sub tree ID3 {Examples (Vi), Target_Attribute,
Attributes – {a})
End
Return Root
```



Attributes	Values	Risk score
Age	x<30 30 < x < 40 40< x < 60	3 4 5
Education	Uneducated School College	5 3 2
Living Area	Urban Rural	5 3
Habits	Smoking Alcohol Chewing Hot beverage	3 5 3 2
Occupational Hazards	Radiation Exposure Chemical Exposure Sunlight Exposure Thermal Exposure	3 3 2 2
Anemia	Yes No	3 1
Weight Loss	Yes No	2 1
Family History of Cancer	Yes No	5 1

Figure 4.3.2. Risk scores for the attributes that represent the significant patterns

4.3.2.1 Significant Pattern mined using Decision tree algorithm

1. Age - gender - living area - family history- anemia-symptoms -> none- Cancer Type -> None. Weightage =100.55

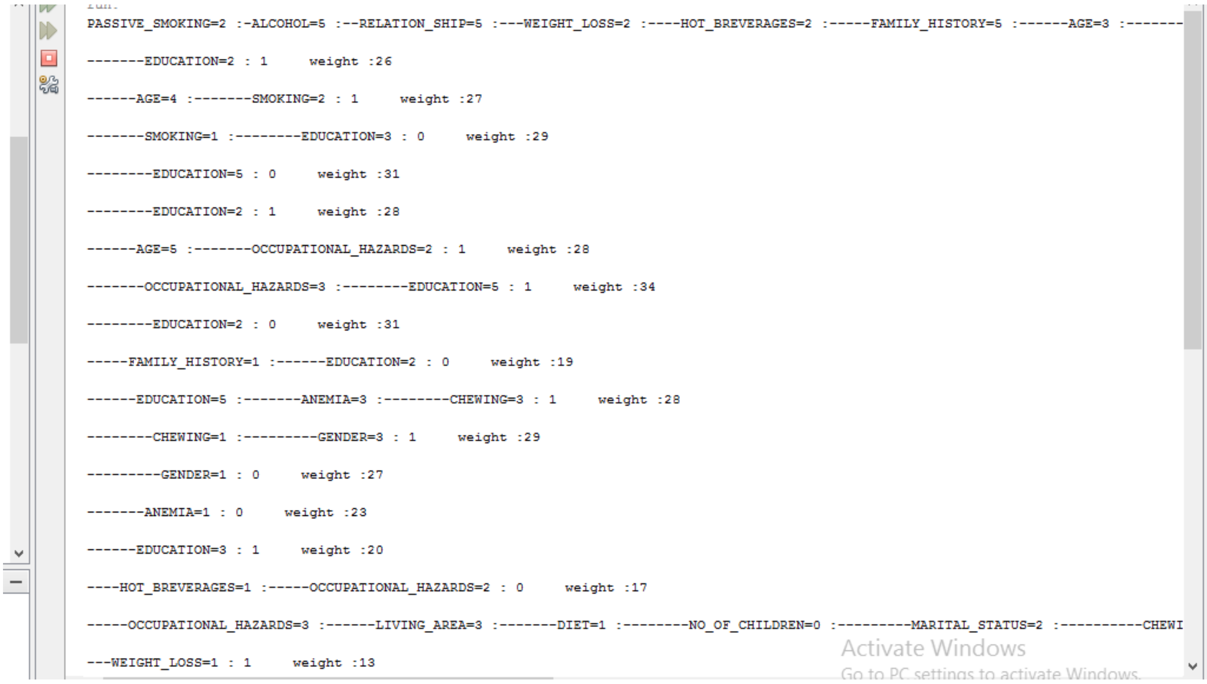
2.Age - gender- marital status-education-smoking-diet- symptoms-> Pain in chest, back, shoulder or arm->Shortness of breath and hoarseness-Cancer Type->Lung Weightage =200.50

3.Gender-Education-Occupational hazards- Alcohol-Family history- Weight loss- symptoms-> severe abdominal pain or bloating-> abdominal pain with blood in stool- Cancer Type ->Stomach Weightage = 180.05

4. Age- gender- no of children- occupational hazards- Family history- relationship with cancer patient- symptoms-> swelling or mass in armpit -> discharge or pain in nipple -> Cancer Type -> Breast. Weightage = 170.55

5.Gender- education- living area- Smoking- Hot beverage- Diet- fast food addiction- Earlier cancer diagnosis- symptoms-> Ulcers in mouth or pain of teeth and jaw-> White or red patches in tongue, gums- Cancer Type -> Oral. Weightage =190.50

Numerical values are given as risk scores to the attributes that have a direct link to the significant patterns mined.



```
PASSIVE_SMOKING=2 :-ALCOHOL=5 :--RELATION_SHIP=5 :---WEIGHT_LOSS=2 :----HOT_BREVERAGES=2 :-----FAMILY_HISTORY=5 :------AGE=3 :-------

-------EDUCATION=2 : 1     weight :26

------AGE=4 :-------SMOKING=2 : 1     weight :27

-------SMOKING=1 :--------EDUCATION=3 : 0     weight :29

--------EDUCATION=5 : 0     weight :31

--------EDUCATION=2 : 1     weight :28

------AGE=5 :-------OCCUPATIONAL_HAZARDS=2 : 1     weight :28

-------OCCUPATIONAL_HAZARDS=3 :--------EDUCATION=5 : 1     weight :34

--------EDUCATION=2 : 0     weight :31

-----FAMILY_HISTORY=1 :------EDUCATION=2 : 0     weight :19

------EDUCATION=5 :-------ANEMIA=3 :--------CHEWING=3 : 1     weight :28

--------CHEWING=1 :---------GENDER=3 : 1     weight :29

---------GENDER=1 : 0     weight :27

-------ANEMIA=1 : 0     weight :23

------EDUCATION=3 : 1     weight :20

----HOT_BREVERAGES=1 :-----OCCUPATIONAL_HAZARDS=2 : 0     weight :17

-----OCCUPATIONAL_HAZARDS=3 :------LIVING_AREA=3 :-------DIET=1 :--------NO_OF_CHILDREN=0 :---------MARITAL_STATUS=2 :----------CHEWI

---WEIGHT_LOSS=1 : 1     weight :13
```

Fig 4.3.2.1 Weight calculation of Decision Tree

4.3.2.2 Rules for Decision Tree

If symptoms = none and risk score = x <45 then result = you don't have cancer, tests = do simple clinical tests to confirm. If symptoms = none and risk score = 45 < x < 60 then result = you may have cancer, tests = do blood test and x ray to confirm

Else if symptom= related to stomach and risk score = x > 45 then result = you have cancer, cancer type = stomach, tests = endoscopy of stomach

If symptom= related to breast and shoulder and risk score =x > 45 then result = you have cancer, cancer type = breast, tests= mammogram and PET scan of breast

If symptom= related to chest and shoulder and risk score =x > 40 then result = you have cancer, cancer type = lung, tests = take CT scan of chest.

If symptom= related to pelvis and lower hip and risk score= x >55 then result = you have cancer, cancer type = cervix, tests = do pap smear test

If symptom= related to head and throat and risk score =x > 40 then result = you have cancer, cancer type = oral, tests =biopsy of tongue and inner mouth.

Else symptom= other symptoms and risk score =x > 40 then result = you have cancer, cancer type = leukemia, tests = biopsy of bone marrow

Based on the above mentioned rules and the calculated risk scores the severity of cancer is known as well as some tests were prescribed to confirm the presence of cancer.

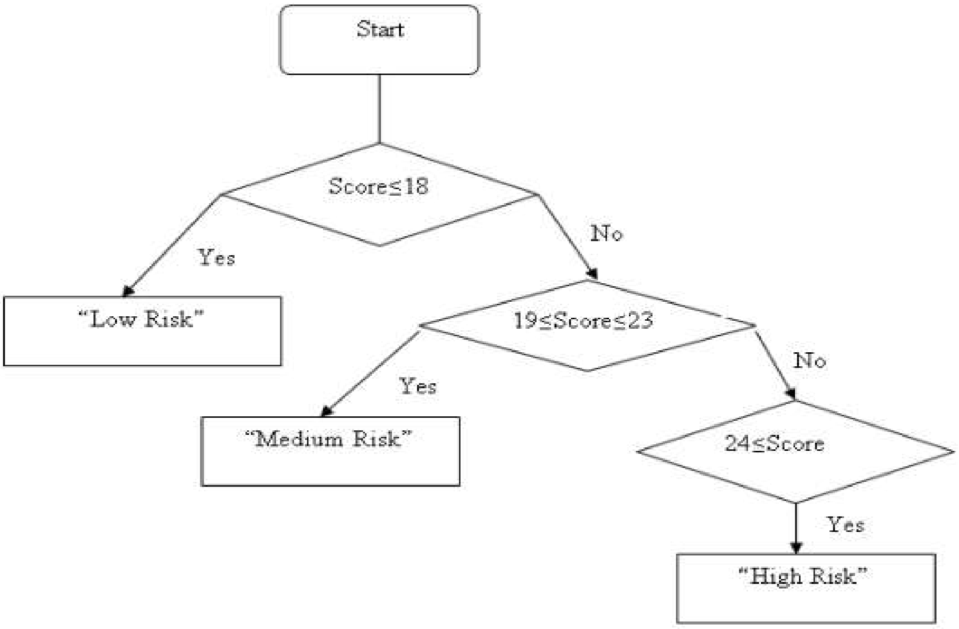


Fig.4.3.2Decision Tree Tracing to predict Risk Score

4.3.3 Clustering using K-Means Algorithm

The instance are now clustered into a number of classes where each class is identified by a unique feature based on the significant patterns mined by the decision tree algorithm. The aim of clustering is that the data object is assigned to unknown classes that has a unique feature and hence maximize the intra class similarity and minimize the interclass similarity. The weightage scores of the significant patterns mined are fed into K- means clustering algorithm to cluster and divide it into cancer and non cancer groups. The cancer group is further subdivided into six groups with each cluster representing a type of cancer. At the beginning the data is assigned to a non cancer cluster and then based on the intensity of the cancer given by its weightage it is either moved to the cancer cluster or gets retained in the non cancer cluster, further the data object is moved between the subgroups of the hierarchical cancer cluster based on the symptoms the data object contains. To calculate the mean of the cluster center the symptoms are given certain values the average of which represents each distinguished cluster. The data objects are distributed to the cluster based on the cluster center to which it is nearest.

It also searches the entire database to find a match to a single input data. The data is moved to that particular cluster if and only if an exact match is found. This technique minimizes the error rate of clustering. The data in the first cluster are all similar with little or no symptoms; no risk factors associated with cancer and low risk scores. Hence the cluster is labeled as Non cancer cluster. The top cluster of the second hierarchical cluster contains all the data that has high risk factors associated with cancer along with distinguished symptoms and high risk scores. The data in the cluster is again fed into k – means clustering algorithm to further subdivide it. The resulting six clusters are separated based on particular symptoms associated with any one type of cancer i.e. lung, cervix, breast, stomach, oral and leukemia. Finally all the data is partitioned into two types of clusters and six sub clusters of the cancer cluster.

4.3.3.1 Clustering algorithm

Algorithm: The k-means clustering algorithm is used for partitioning the data into cancer and non cancer clusters, where the initial cluster centers is represented by the mean value of the weightage of significant patterns.

Input: k: the number of clusters. D: data set containing n objects.

Output: A set of hierarchical clusters

Begin 1) choose two mean values from weightage of significant patterns as the initial cluster centers 2) assign each object to the cluster to which it is most similar based on the mean value of the weightage. 3) Update the cluster means by calculating mean value of all the objects in the cluster. 4) End

Now two clusters have been generated based on the weightage scores of the significant pattern. The two clusters are named as Non cancer and Cancer clusters. The mean weightage of the non cancer cluster is significantly lower than the cancer cluster. Again partition the cancer cluster to generate five sub clusters each representing a type of cancer.

Begin 1) arbitrarily chooses k objects from cancer cluster S with distinguished values for its symptoms. 2) Assign each object in S to the cluster whose mean value is closer to its symptom. 3) Update the cluster means and 4) Repeat step 2 and 3 until no change 5) End.

The output is five clusters with each representing a type of cancer.

4.3.4 Creating a Web Page using Jframes

Create a JFrame container

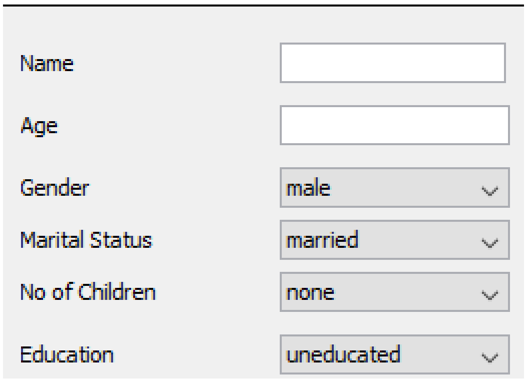
1. In the Projects window, right-click the `Number Addition` node and choose `New` > `Other`.
2. In the New File dialog box, choose the `Swing GUI Forms` category and the `JFrame Form` file type. Click Next.
3. Enter `Number Addition UI` as the class name.
4. Enter `my number addition` as the package.
5. Click Finish.

The IDE creates the `Number Addition UI` form and the `Number Addition UI` class within the `Number Addition` application, and opens the `Number Addition UI` form in the GUI Builder. The `Number Addition` package replaces the default package.

Adding Components: Making the Front End

Next we will use the Palette to populate our application's front end with a JPanel. Then we will add three JLabels, three JTextFields, and three JButtons. If you have not used the GUI Builder before, you might find information in the Designing a Swing GUI in NetBeans IDE tutorial on positioning components useful.

Once we are done dragging and positioning the aforementioned components, the JFrame should look something like the following screenshot.



Name	
Age	
Gender	male
Marital Status	married
No of Children	none
Education	uneducated

Fig 4.3.4 Example of User Interface Diagram using jframes

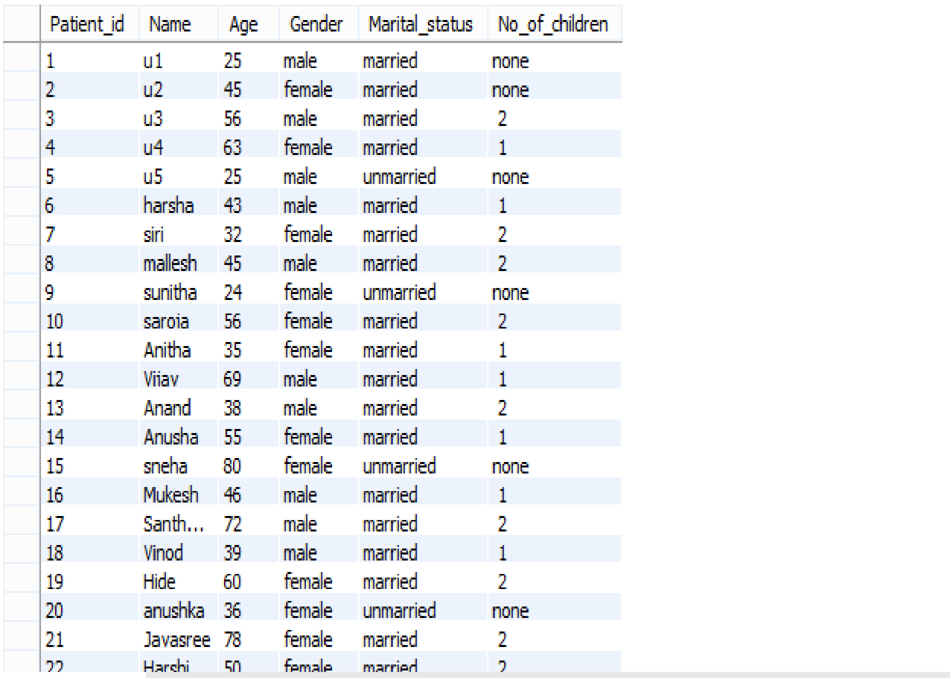
4.3.4.1 Input:

When the user opens the web page he/she can find the login form where the form contains both old user and a new user

If the user selects Old user then it will redirect to a page wherein the user have to login with his register id

When the user logins with register id it will display all the previous data stored regarding his health status.

The data is recollected from the database where all the data is divided into number of tables



Patient_id	Name	Age	Gender	Marital_status	No_of_children
1	u1	25	male	married	none
2	u2	45	female	married	none
3	u3	56	male	married	2
4	u4	63	female	married	1
5	u5	25	male	unmarried	none
6	harsha	43	male	married	1
7	siri	32	female	married	2
8	mallesh	45	male	married	2
9	sunitha	24	female	unmarried	none
10	saroia	56	female	married	2
11	Anitha	35	female	married	1
12	Viiav	69	male	married	1
13	Anand	38	male	married	2
14	Anusha	55	female	married	1
15	sneha	80	female	unmarried	none
16	Mukesh	46	male	married	1
17	Santh...	72	male	married	2
18	Vinod	39	male	married	1
19	Hide	60	female	married	2
20	anushka	36	female	unmarried	none
21	Javasree	78	female	married	2
22	Harshi	50	female	married	2

Fig 4.3.4.1 Database of the User with Attributes

If the user selects new user then it will redirect to a new page where he/she has to enter all the details regarding their health status

After entering all the details it will generate an individual id for every patient so that he can further verify his report from old user login

4.4 Sequence Diagram with Timelines

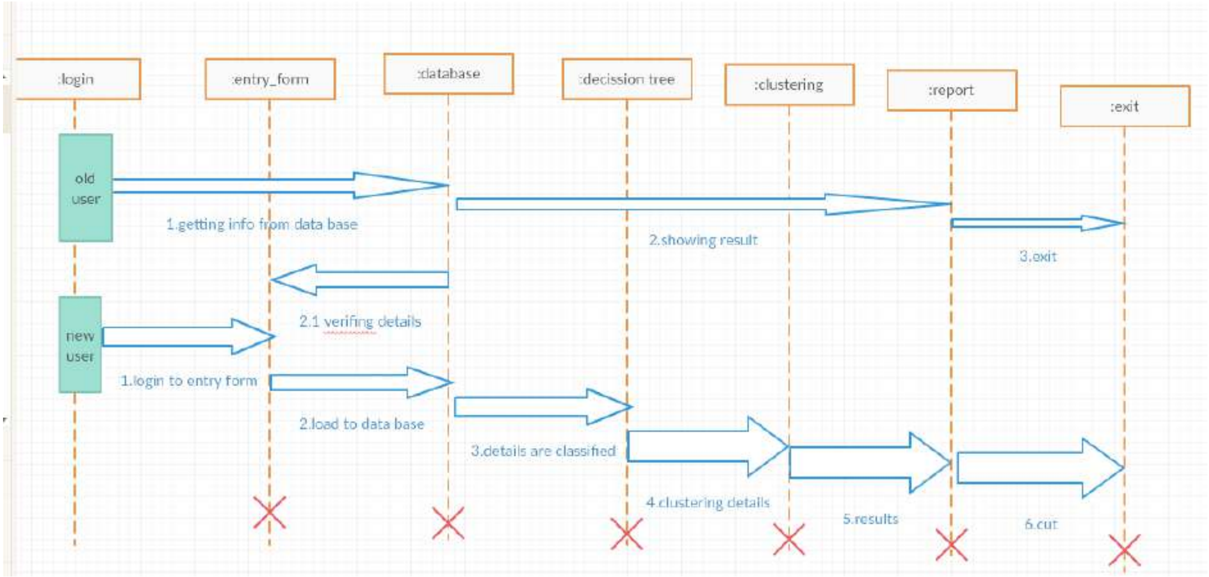


Figure 4.5 sequence diagram with timelines

5.TESTING

5.1Test Cases

Test Case ID	Gender	Age	Habits	Symptoms	Expected result (may have)	Actual result (may have)	Pass/Fail
1	Female	35	a	1,2	Stomach cancer	Stomach cancer	pass
2	Male	51	a, b	3,4	Lung cancer	Lung cancer	Pass
3	Female	49	a, d	5,6	Breast cancer	Breast cancer	Pass
4	Female	40	a, b, c	7,8	Head and throat	Head and throat	Pass
5	Male	56	--	9	No cancer	No cancer	Pass

Habits:

a. smoking

b. alcohol

c. chewing

d. hot beverages

e. passive smoking

Symptoms:

1. Feeling bloated after eating

2. Severe persistent heart burn

3. Cough that doesn’t go away

4. Feeling tired or weak with a chest pain

5. Lump or swelling

6. Cyst in arm pit

7.thick patches in mouth and cheeks

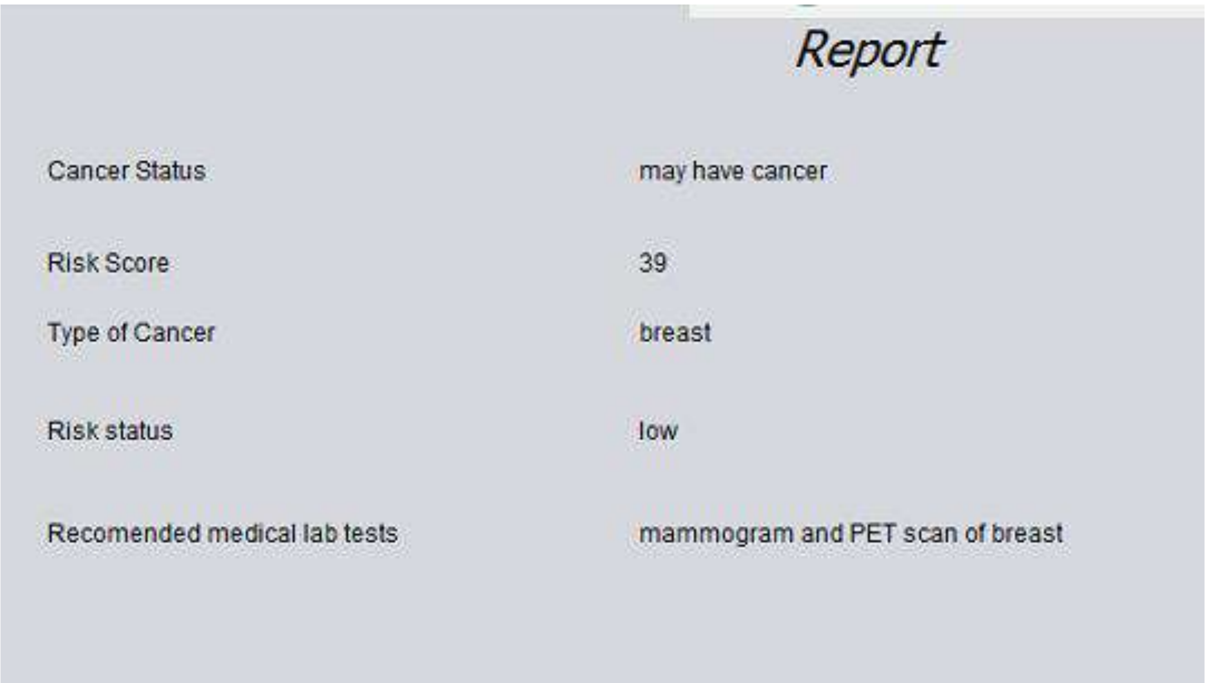
8. Problem in swallowing

9.Indigestion and Pain in upper abdomen

6.RESULTS AND CONCLUSION

6.1 Result Analysis

The results are separated into three parts. The first is the frequent and significant pattern discovery. The second is mapping the cancer to its cluster and the third is prediction by giving risk score as output. At the beginning all the input data is stored in the non cancer cluster further it gets classified and clustered by the model. A single user input data is fed into the system and gets classified according to the significant pattern to which it matches through decision tree, gets analyzed for its risk score merged with either one of the Non cancer and cancer clusters. This gives the result whether the patient has cancer or not. Again the data is merged with any one of the subsequent cancer clusters to which its symptoms belong. The type of cancer the patient has is found out from this step. It is also compared with the entire database to find its exact or relevant match so that a data with severe cancer related symptoms gets a pair only in the cancer cluster and it cannot get merged with non cancer cluster even by mistake. With each new entry getting appended to the model the process becomes intelligent and ensures accurate results

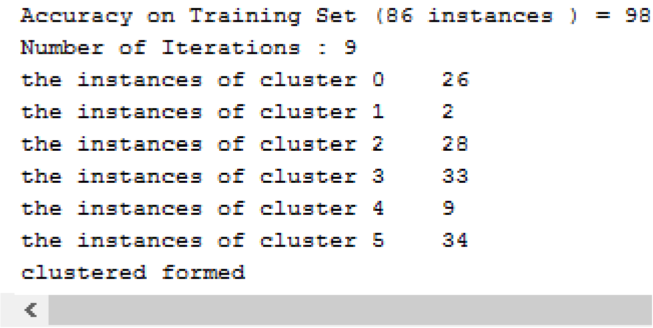


Report

Cancer Status	may have cancer
Risk Score	39
Type of Cancer	breast
Risk status	low
Recomended medical lab tests	mammogram and PET scan of breast

Fig 6.1.1 Overall Report Analysis

Clustering Output:



```
Accuracy on Training Set (86 instances ) = 98
Number of Iterations : 9
the instances of cluster 0     26
the instances of cluster 1     2
the instances of cluster 2     28
the instances of cluster 3     33
the instances of cluster 4     9
the instances of cluster 5     34
clustered formed
```

Fig 6.1.2 Clustering of all types Cancers

Graph output:

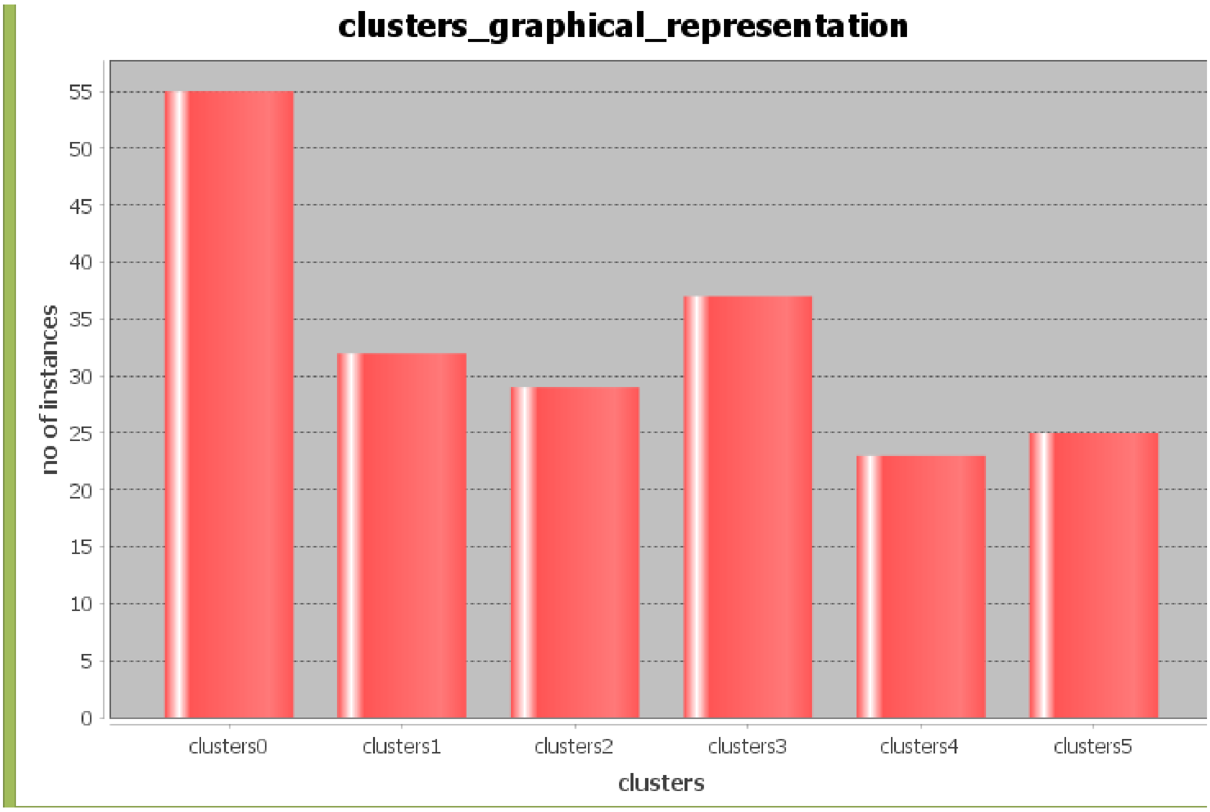


Fig 6.1.3 Graphical Representation of Clustering

6.2 Conclusion and Future Scope

In this paper a multi layered method combining clustering and decision tree techniques to build a cancer risk prediction system is proposed. Cancer has become the leading cause of death worldwide. The most effective way to reduce cancer deaths is to detect it earlier. Many people avoid cancer screening due to the cost involved in taking several tests for diagnosis. This prediction system may provide easy and a cost effective way for screening cancer and may play a pivotal role in earlier diagnosis process for different types of cancer and provide effective preventive strategy. This system can also be used as a source of record with detailed patient history in hospitals as well as help doctors to concentrate on particular therapy for any patient

In future, more data will be inserted to get more accuracy regarding risk score so that it might be helpful in the medical stream to predict cancer more accurately.

APPENDIX

I. Screenshots

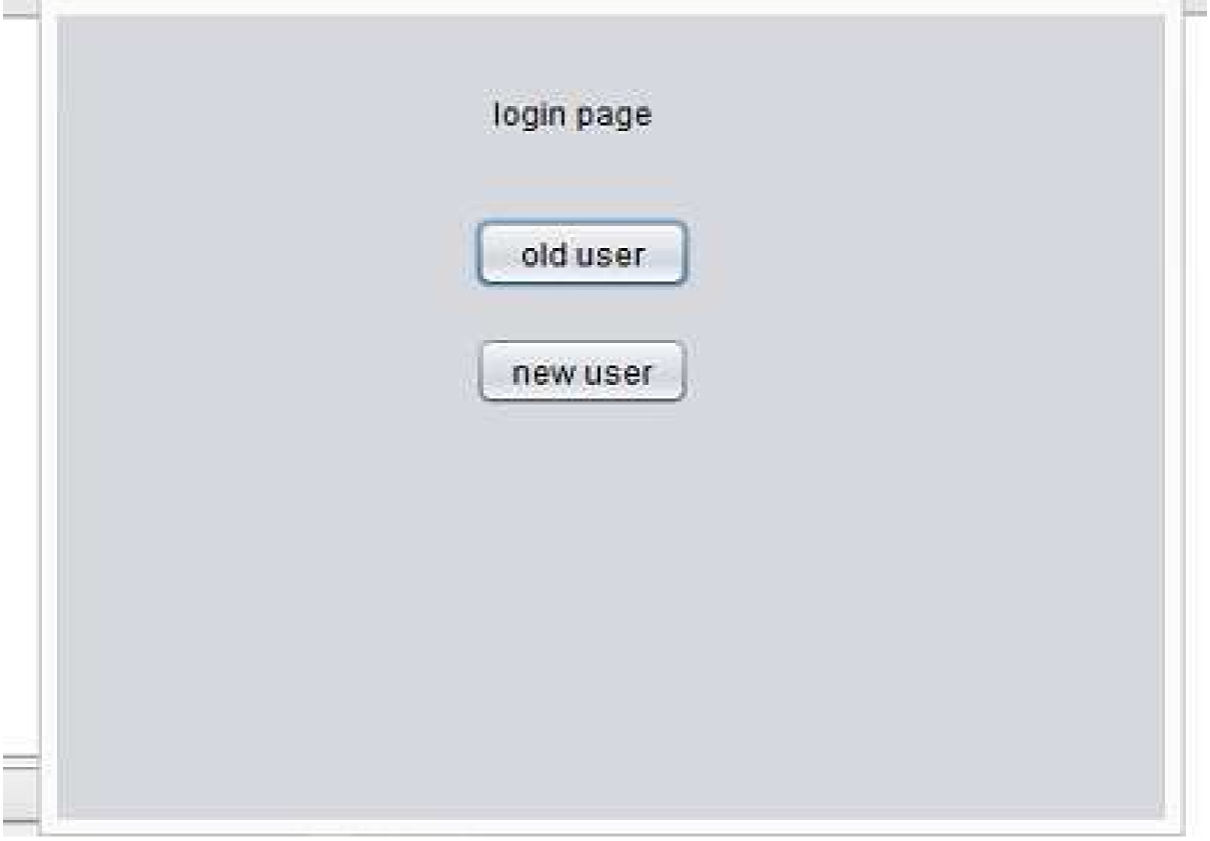
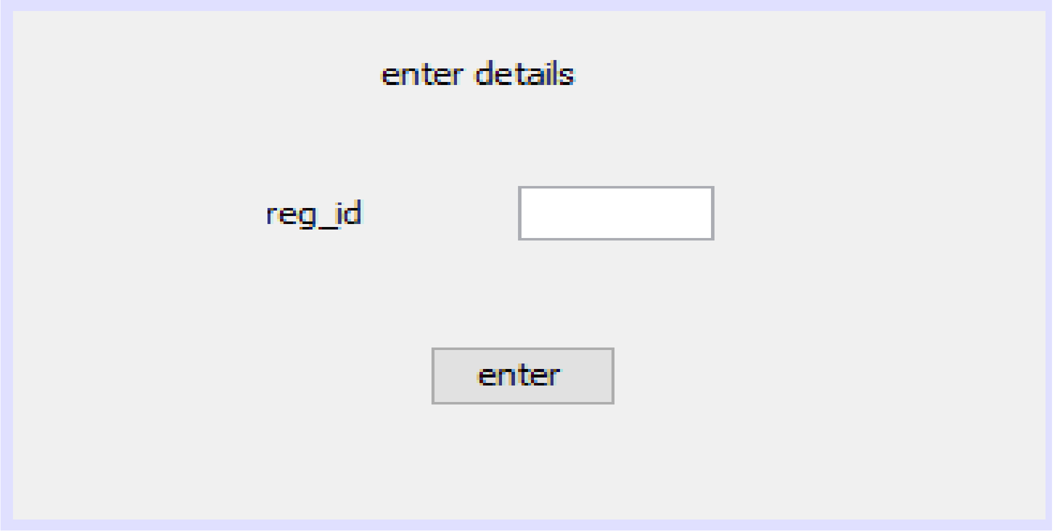


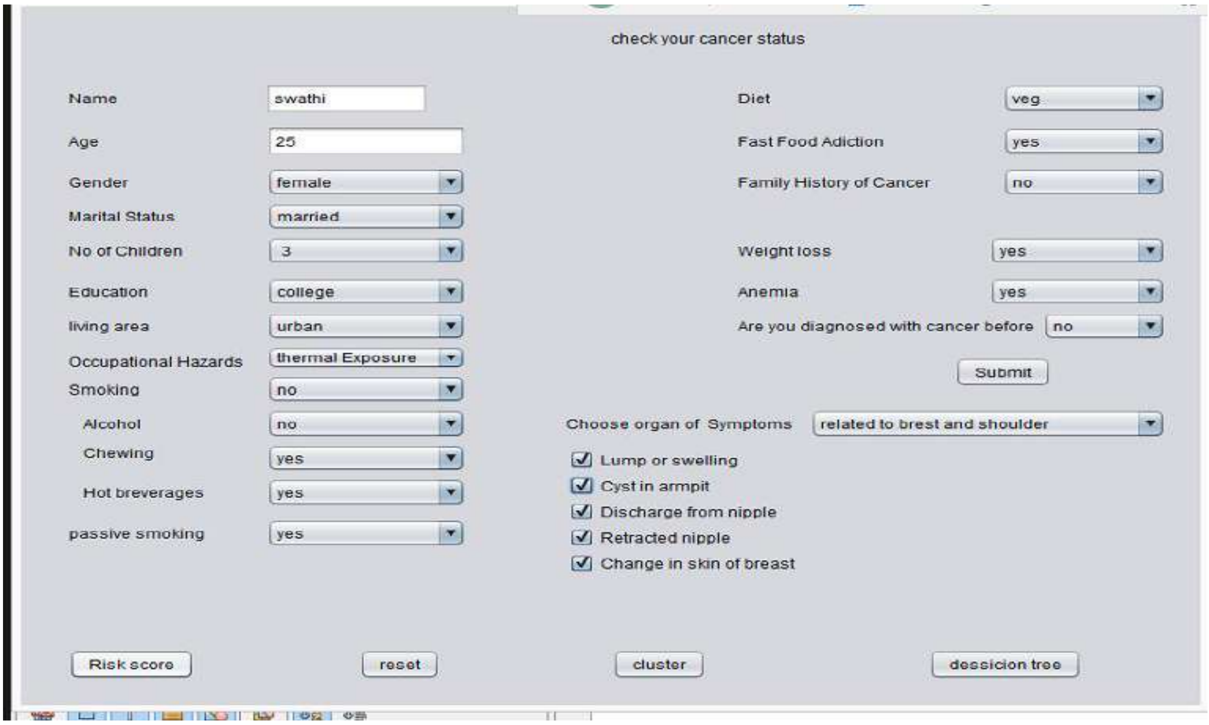
Fig 7.1.1 Login Page for the user



enter details

reg_id

Fig 7.1.2 Old User login Form



check your cancer status

Name swathi

Age 25

Gender female

Marital Status married

No of Children 3

Education college

living area urban

Occupational Hazards thermal Exposure

Smoking no

Alcohol no

Chewing yes

Hot breverages yes

passive smoking yes

Diet veg

Fast Food Adiction yes

Family History of Cancer no

Weight loss yes

Anemia yes

Are you diagnosed with cancer before no

Choose organ of Symptoms related to brest and shoulder

☑ Lump or swelling

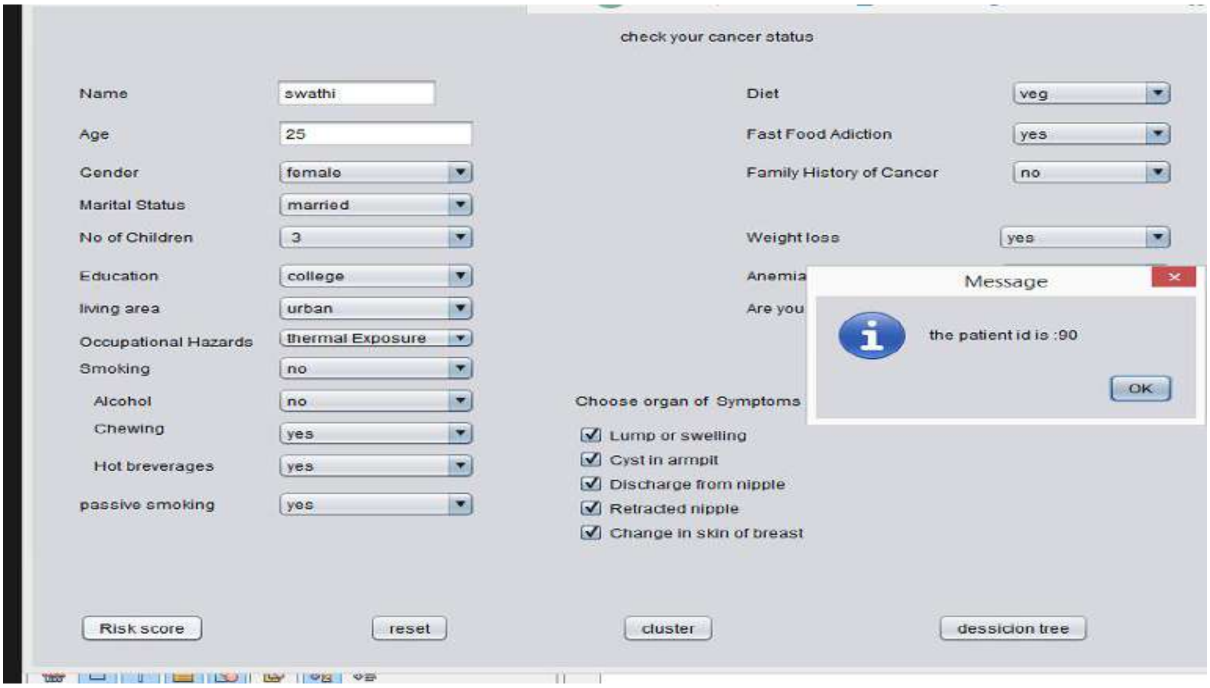
☑ Cyst in armpit

☑ Discharge from nipple

☑ Retracted nipple

☑ Change in skin of breast

Fig 7.1.3 New User entry form



check your cancer status

Name	swathi
Age	25
Cender	female
Marital Status	married
No of Children	3
Education	college
living area	urban
Occupational Hazards	thermal Exposure
Smoking	no
Alcohol	no
Chewing	yes
Hot breverages	yes
passive smoking	yes
Diet	veg
Fast Food Adiction	yes
Family History of Cancer	no
Weight loss	yes
Anemia	
Are you	

Choose organ of Symptoms

☑ Lump or swelling
☑ Cyst in armpit
☑ Discharge from nipple
☑ Retracted nipple
☑ Change in skin of breast

Message

the patient id is :90

Fig 7.1.4Generating patient id

II. Code

ID3 Algorithm

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
public class ID3Learner {
  String FileNameToRead;// filename to read the training data  set
  int PercentageOfDataToLEarnFrom;// this is the percent , which shows how
    public ID3Learner(String FileName, int percs) {
  FileNameToRead = FileName;
  PercentageOfDataToLEarnFrom = percs;
  }
    public TreeNode startLearning() {
                if (FileNameToRead == null) {
                        System.out.println("---- Error ------");
                        System.out.println("---- Please Specify test data set ------");
                }
                if (PercentageOfDataToLEarnFrom < 0) {
                        System.out.println("---- Error ------");
```

```
                                System.out.println("---- Please Specify %correctly ------");
}
MatrixData matrix = new MatrixData();
matrix.prepareMatrix(FileNameToRead, PercentageOfDataToLEarnFrom);
HashMap<String, int[]> setTrainingVector = new HashMap<String, int[]>();
// Now i need a set of R training vectors
for (int i = 0; i < matrix.coloumns - 1; i++) {
int[] trainingVector = new int[matrix.Numrows];
matrix.fillArray(trainingVector, i);
setTrainingVector.put(matrix.Headers.get(i), trainingVector);
}
int[] FinalClass = new int[matrix.Numrows];
matrix.fillArray(FinalClass, matrix.coloumns - 1);
TreeNode rootNode = new TreeNode();
rootNode.setAtrvalue(-1);learnTree(setTrainingVector, FinalClass, rootNode, matrix);
return rootNode;}
            public void learnTree(HashMap<String, int[]> setTrainingVector,
int[] FinalClass, TreeNode node, MatrixData matrix) {
        if (checkFinalClass(FinalClass, 0)) {
node.fClass = 0;
return;
} else if (checkFinalClass(FinalClass, 1)) {
node.fClass = 1;
return;
}
if (setTrainingVector.entrySet().size() == 1) {
int cPos = getCountPositives(FinalClass);
int cNeg = FinalClass.length - cPos;
if (cPos >= cNeg) {
node.fClass = 0;
return;
} else {
node.fClass = 1;
return;
```

```
  }
  } else {
            HashMap<String, Double> attributesGains = new HashMap<String, Double>();
            HashMap<String,     ArrayList<Integer>>     mapAttributesValuesInListUnique     =     new
HashMap<String, ArrayList<Integer>>();
                              ouble entropyS = getEntropy(FinalClass);
            for (Map.Entry entry : setTrainingVector.entrySet()) {
            HashMap<Integer, Integer> atrPositive = new HashMap<Integer, Integer>();
                    HashMap<Integer, Integer> atrNegative = new HashMap<Integer, Integer>();
                    ArrayList<Integer> atrUnique = new ArrayList<Integer>();

                    int[] trainingClass = (int[]) entry.getValue();
                    for (int i = 0; i < trainingClass.length; i++) {
                              addOnlyUnique(atrUnique, trainingClass[i]);
                              if (FinalClass[i] == 0)// its a positive
                              {
                                        if (atrPositive.containsKey(trainingClass[i])) {
                                                  atrPositive.put(trainingClass[i],
                                                            atrPositive.get(trainingClass[i]) + 1);
                              } else {
                                        atrPositive.put(trainingClass[i], 1);
                              }
                              } else {
                                        if (atrNegative.containsKey(trainingClass[i])) {
                                                  atrNegative.put(trainingClass[i],
                                                            atrNegative.get(trainingClass[i]) + 1);
                              } else {
                                        atrNegative.put(trainingClass[i], 1);
                              }
                    }
  }

  mapAttributesValuesInListUnique.put((String) entry.getKey(),atrUnique)
  {
```

```
double gain = entropyS;
for (int tempAttr : atrUnique) {
double entropyTemp = 0.0;
int positives = 0;
int negatives = 0;
if (atrPositive.get(tempAttr) != null)
positives = atrPositive.get(tempAttr);
if (atrNegative.get(tempAttr) != null)
negatives = atrNegative.get(tempAttr);
double val1 = (double) (positives)
/ (positives + negatives);
double val2 = (double) (negatives)
/ (positives + negatives);
entropyTemp = -(val1 * log2(val1))
- (val2 * log2(val2));
   gain = gain - ((((double) positives + negatives) / trainingClass.length) * entropyTemp);
}
attributesGains.put((String) entry.getKey(), gain);
}

}String attributeWithMAxGain = "";
double maxGainValue = 0.0;
int indexToChoose = 0;
for (Map.Entry entry : setTrainingVector.entrySet()) {
double tempGain = attributesGains.get((String) entry.getKey());
if (indexToChoose == 0) {
maxGainValue = tempGain;
attributeWithMAxGain = (String) entry.getKey();
indexToChoose++;
}

if (tempGain > maxGainValue) {
maxGainValue = tempGain;
attributeWithMAxGain = (String) entry.getKey();
```

```
}
}
node.setAttributeName(attributeWithMAxGain);
node.setfClass(-1);
node.setGain(maxGainValue);

ArrayList<Integer>atrUniqueValuesForAttrMaxGain=mapAttributesValuesInListUnique
.get(attributeWithMAxGain);

for (int tempAtrUniqueValue : atrUniqueValuesForAttrMaxGain) {

TreeNode NodeChild = new TreeNode();
NodeChild.setAtrvalue(tempAtrUniqueValue);
node.getBranches().add(NodeChild);
MatrixData matrixChild = matrix.splitMatrix(
attributeWithMAxGain, tempAtrUniqueValue);
HashMap<String, int[]> setTrainingVectorChild = new HashMap<String, int[]>();
for (int i = 0; i < matrixChild.coloumns - 1; i++) {
int[] trainingVectorChild = new int[matrixChild.Numrows];
matrixChild.fillArray(trainingVectorChild, i);
setTrainingVectorChild.put(matrixChild.Headers.get(i),
trainingVectorChild);
}
int[] FinalClassChild = new int[matrixChild.Numrows];
matrixChild
.fillArray(FinalClassChild, matrixChild.coloumns - 1);
learnTree(setTrainingVectorChild, FinalClassChild, NodeChild,
matrixChild);

}

return;

}
```

```
}
public boolean checkFinalClass(int[] FinalClass, int valueToChecked) {
for (int i = 0; i < FinalClass.length; i++) {
if (FinalClass[i] != valueToChecked)
return false;
}
return true;
}
public int getCountPositives(int[] FinalClass) {
int countPos = 0;
for (int i = 0; i < FinalClass.length; i++) {
if (FinalClass[i] == 0)
countPos++;
}
return countPos;
}
public double getEntropy(int[] vector) {
double entropy = 0.0;
int positives = 0;
int negatives = 0;
for (int i = 0; i < vector.length; i++) {
if (vector[i] == 0)// its a positive
{
positives++;
} else {// FinalClass is negative
negatives++;
}
}
double val1 = (double) (positives) / (positives + negatives);
double val2 = (double) (negatives) / (positives + negatives);
entropy = -(val1 * log2(val1)) - (val2 * log2(val2));
return entropy;
}
```

```
 public static double log2(double num) {
 if (num <= 0)
 return 0.0;
 return (Math.log(num) / Math.log(2));
 }
 public void addOnlyUnique(ArrayList<Integer> data, int val) {
 if (!data.contains(val))
 data.add(val);
 }

}
```

K-Means Algorithm

```
import java.util.ArrayList;
import java.util.List;
import java.util.Random;
public class KMeans {
    private final List<Point> dataPoints;
    public List<Cluster> clusters;
    private boolean hasntChangedClusters;
        public KMeans(){
        int k=6;
    int seed=34;
    Data data = new Data();
    dataPoints = data.dataSet;
    clusters = new ArrayList<>();
    generateClusterCentroids(k,seed);
        int numIterations = 0;
        while (!hasntChangedClusters) {
            hasntChangedClusters = true;
            assignPointsToClusters();
            updateCentroids();
            numIterations++;
        }
        System.out.println("Number of Iterations : " + numIterations);
        getCurrentCluster();
    }
    private void generateClusterCentroids(int numClusters , int seed) {
        Random random = new Random(seed);
        for (int i = 0; i < numClusters; i++) {
            Cluster newCluster = new Cluster();
            newCluster.centroid = dataPoints.get(random.nextInt(dataPoints.size()));
            clusters.add(newCluster);
        }
    }
    void getCurrentCluster(){
     int sizeofdatpoints=dataPoints.size();
     Point p=dataPoints.get(sizeofdatpoints-1);
     int currentClusterIndex=p.getClusterIndex();
     System.out.println("the total number of instances :"+dataPoints.size());
     System.out.println("the present data value belongs to cluster"+currentClusterIndex+":
"+clusters.get(currentClusterIndex).points.size());
    }
    private void assignPointsToClusters() {
        double x, y, cx, cy;
        double distance;
        double lowestDistance = Double.MAX_VALUE;
        int indexOfClosestCluster = 0;
        for (int i = 0; i < dataPoints.size(); i++) {
            Point currentPoint = dataPoints.get(i);
            x = currentPoint.getX();
            y = currentPoint.getY();
            lowestDistance = Double.MAX_VALUE;
            for (int j = 0; j < clusters.size(); j++) {
                Point currentCentroid = clusters.get(j).centroid;
                cx = currentCentroid.getX();
                cy = currentCentroid.getY();
                distance = Distance.calculateDistance(x, y, cx, cy);
                if (distance < lowestDistance) {
                    indexOfClosestCluster = j;
                    lowestDistance = distance;
                }
            }
            if (indexOfClosestCluster != currentPoint.getClusterIndex()) {
                clusters.get(indexOfClosestCluster).points.add(currentPoint);
                if (currentPoint.getClusterIndex() != -1) {
                    clusters.get(currentPoint.getClusterIndex()).points.remove(currentPoint);
                }
                currentPoint.setClusterIndex(indexOfClosestCluster);
                hasntChangedClusters = false;
            }
        }
    }
 private void updateCentroids() {
        for (Cluster c : clusters) {
            double xAccumulated = 0, yAccumulated = 0, num = 0;
            for (Point p : c.points) {
                xAccumulated += p.getX();
                yAccumulated += p.getY();
                num++;
            }

            c.centroid.setX(xAccumulated / num);
            c.centroid.setY(yAccumulated / num);
        }
    }
}
```

III.References

[1] Ada and Rajneet Kaur "Using Some Data Mining Techniques to Predict the Survival Year of Lung Cancer Patient" International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 2, Issue. 4, April 2013, pg.1 – 6, ISSN 2320–088X

[2] V.Krishnaiah "Diagnosis of Lung Cancer Prediction System Using Data Mining Classification Techniques" International Journal of Computer Science and Information Technologies, Vol. 4 (1) 2013, 39 – 45 www.ijcsit.Com ISSN: 0975-9646.

[3] Charles Edeki "Comparative Study of Data Mining and Statistical Learning Techniques for Prediction of Cancer Survivability" Mediterranean journal of Social Sciences Vol 3 (14) November 2012, ISSN: 2039-9340.

[4] A. Sahar "Predicting the Serverity of Breast Masses with Data Mining Methods" International Journal of Computer Science Issues, Vol. 10, Issues 2, No 2, March2013 ISSN (Print):1694-0814| ISSN (Online):1694-0784 www.IJCSI.org

[5] Rajashree Dash "A hybridized K-means clustering approach for high dimensional dataset" International Journal of Engineering, Science and Technology Vol. 2, No. 2, 2010, pp. 59-66

[6] Ritu Chauhan"Data clustering method for Discovering clusters in spatial cancer databases" International Journal of Computer Applications (0975-8887) Volume10-No.6, November 2010.

[7] Dechang Chen "Developing Prognostic Systems of Cancer Patients by Ensemble Clustering" Hindawi publishing corporation, Journal of Biomedicine and Biotechnology Volume 2009, Article Id 632786.

[8] S M Halawani "A study of digital mammograms by using clustering algorithms" Journal of Scientific & Industrial Research Vol. 71, September 2012, pp. 594-600.

[9] Zakaria Suliman zubi "Improves Treatment Programs of Lung Cancer using Data Mining Techniques" Journal of Software Engineering and Applications, February 2014, 7, 69-77

[10] Labeed K Abdulgafoor "Detection of Brain Tumor using Modified K-Means Algorithm and SVM" International Journal of Computer Applications (0975 –8887) National Conference on Recent Trends in Computer Applications NCRTCA 2013

[11] K.Kalaivani "Childhood Cancer-a Hospital based study using Decision Tree Techniques" Journal of Computer Science 7(12): 1819-1823, 2011 ISSN: 1549-3636

[12] Boris Milovic "Prediction and Decision Making in Health Care using Data Mining" International Journal of Public Health Science Vol. 1, No. 2, December 2012, pp. 69-78 ISSN: 2252-8806

[13] T.Revathi "A Survey on Data Mining Using Clustering Techniques" International

Journal of Scientific & Engineering Research Http://Www.Ijser.Org, Volume 4, Issue 1, January-2013, Issn 2229-5518

[14] Shomona Gracia Jacob "Data Mining in Clinical Data Sets: A. Review" International Journals of Applied Information System (IJAIS) - ISSN: 2249-0868

[15] G. Rajkumar "Intelligent Pattern Mining and Data Clustering for Pattern Cluster Analysis using Cancer Data" International journal of Engineering Science and Technology Vol. 2(12), 2010, ISSN: 7459-7469.

[16] M. Durairaj "Data Mining Applications in Healthcare Sector: A Study" International journal of Scientific & Technology Research, Volume 2, Issue 10, October 2013, ISSN: 2277-8616

[17] Vikas Chaurasia "Data Mining Techniques: To Predict and Resolve Breast Cancer Survivability" International journal of Computer Science and Mobile Computing (IJCSMC), Vol.3, Issue. 1, January 2014, pg.10-22, ISSN: 2320-088X

[18] T.Sridevi "An Intelligent Classifier for Breast Cancer Diagnosis based on K-Means Clustering and Rough Set" International Journal of Computer Applications (0975 – 8887) Volume 85 – No 11, January 2014

[19] Reeti Yadav "Chemotheraphy Prediction of Cancer Patient by Using Data Mining Techniques" International Journal of Computer Applications (0975-8887), Volume76-No.10, August 2013

[20] K.Balachandran "Classifiers based Approach for Pre- Diagnosis of Lung Cancer Disease" International Journal of Computer Applications® (IJCA) (0975 – 8887)