A Final Year Project Report on:

“**Detecting High Risk Taxpayers Using Data Mining Techniques**”

*Submitted by:*

Under guidance of:

**Mrs.**

**Department of Computer Engineering**

**XYZ OF ENGINEERING**

2016-2017

**CERTIFICATE**

This is to certify that the pre report on the project entitled

“**Detecting High Risk Taxpayers Using Data Mining Techniques**”

*Submitted by:*

A partial fulfillment for BACHELOR OF COMPUTER ENGINEERING degree course of Mumbai University for year 2016-2017

INTERNAL GUIDE HOD

**( Prof. ) (Prof. )**

INTERNAL EXAMINER PRINCIPAL

EXTERNAL EXAMINER



# **ACKNOWLEDGEMENT**

No project is ever complete without the guidance of those experts who have already traded this past before and hence become master of it and as a result, our leader. So we would like to take this opportunity to take all those individuals who have helped us in visualizing this project.

We express our deep gratitude to our project guide Mrs. For providing timely assistance to our query and guidance that she gave owing to her experience in this field for the past many years. She had indeed been a lighthouse for us in this journey.

We would also take this opportunity to thank our project coordinator Mr. For his guidance in selecting this project and also for providing us all this details on proper presentation of this project.

We extend our sincerity appreciation to our entire Professor from COLLEGE OF ENGINEERING for their valuable inside and tip during the designing of the project. Their contributions have been valuable in so many ways that we find it difficult to acknowledge them individually.

We are also grateful to our HOD Mrs. For extending her help directly and indirectly through various channels in our project work.

.

Thanking You,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ABSTRACT**

Taxpayer means as the name refers to a person or organization who wants to pay a specific amount or tax to the government on its income. Risk refers to loss of anything, but look at from the angle of government tax its a avoid of paying tax on time. Sometimes, On purpose people show their income low to avoid the government tax. Hiding information of buying ,selling of properties in the form of lands it's also one type of risk in government tax. Because of these risks, the government suffers from a lot of difficulties during the audit period. Government faces the challenge of identifying and collecting taxes that have successfully hidden from paying the proper tax; it's also called as Tax fraud. Sometimes, Taxpayers misrepresent the financial facts to the government, is also tax fraud.

The main purpose of this system is to design and find out the high risk taxpayers and notify the amount of tax to the high risk taxpayers so that it would never forget to pay the government tax. In this system, various methods are used like Classification, Association, Regression, Data mining to detect the high risk taxpayers. This system classifies the risk according to how much tax the person wants to pay. This tax is associated with the person based on their income. High risk taxpayers means the person who wants to pay a high amount of tax. If these taxpayers are increasing day by day, the government wants to suffer economic problems in our country and how the government works for our country. Therefore, this system is helpful not only to detect the high risk taxpayers but also to resolve this drawback.

**Keywords:** Taxpayer, Risk, Tax fraud detection, Classification, Association, Regression, Data mining, High risk taxpayers.

INDEX

|  |  |  |
| --- | --- | --- |
| SR.NO | TITLE | PG.NO |
| 1) | INTRODUCTION | 1 |
| 2) | LITERATURE SURVEY | 5 |
| 3) | PROBLEM DEFINITION | 8 |
| 4) | REQUIREMENT ANALYSIS | 11 |
| 5) | PLANNING AND ESTIMATION | 13 |
| 6) | ALGORITHM | 15 |
| 7) | IMPLEMENTATION | 22 |
| 8) | ADVANTAGES & DISADVANTAGES | 27 |
| 9) | FUTURE MODIFICATIONS | 29 |
| 10) | APPLICATION | 31 |
| 11) | BIBLIOGRAPHY | 33 |
| 12) | SCREENSHOTS | 48 |
| 13) | SOURCE CODE |  |

**Chapter 1**

**INTRODUCTION**

**INTRODUCTION**

Tax fraud is a major issue that incurs expenses in terms of the loss of government revenues, those results in less economical tax programs and therefore the inequity between fraud taxpayers and honest taxpayers. Tax Administration is under increasing pressure, since the monetary crisis of 2008 and therefore the massive deficits that followed, to collect extra tax revenues and cut back commercial enterprise fraud. Effective management of tax fraud requires addressing a basic applied mathematics drawback of non-detection, which may bias estimates of the quantity of fraud and therefore the relative fraud propensities of various social economic teams. Tax fraud detection involves processing an oversized quantity of information in search of fraud behavior that needs quick and economical algorithms, among that data processing provides relevant techniques that may facilitate tax administration to take preventive measures and improve tax style.

Financial fraud detection tools are delivered to scenic in order to deal with this drawback and to produce reliable solutions to business. is often fraud is generally discovered through outlier detection method enabled by data processing techniques, that also determine valuable info by revealing hidden trends, relationships, patterns found in an exceedingly large database . data processing, defined as “a method that uses statistical, mathematical, artificial intelligence, and machine learning techniques to extract and determine helpful information and subsequently gain information from an outsized database”, could be a major contributor for detecting differing types of financial fraud through its numerous ways, such as, logistic regression, decision tree, support vector machine (SVM), neural network (NN) and naïve bayes. a number of these techniques exceed the others in specific financial contexts. Glancy and Yadav (2011) divide those contexts to a few main areas: internal, insurance and credit. Jans et al. (2011) additional classify internal fraud into 2 categories: budget fraud and dealings fraud . They define financial statement fraud as “the intentional statement of sure money values to reinforce the appearance of profitability and deceive shareholders or creditors” whereas transaction fraud captures the method of snatching organizational assets.

Auditing tax declarations could be a slow and expensive method, so that tax authorities are required to develop efficient ways to tackle this drawback and improve tax style. This issue motivates our proposal. In our analysis we have a tendency to explore the relevancy of the info mining techniques in developing a segmentation model which will contribute to tax style evaluation and therefore the characterization of the segments of potential fraud taxpayers within the Personal Income Tax. Despite the rise within the use of those screening and classification models for police work fraud patterns homeward-bound at audit designing, there are not any studies that focus on the identification of tax advantages within the taxation structure that area unit a lot of doubtless to be utilized by potential fraud taxpayers. to boot, this proposal to phase and characterize potential dishonorable taxpayers also can be applied to differing kinds of taxes.

Data mining techniques are used in this system like regression, classification, clustering.

**Aim of Project**

To detect financial fraud using data mining tools within one decade and communicate the current trends to academic scholars and industry practitioners.

**Objectives of the Project :**

The objectives of the Detecting High Risk Taxpayers Using Data Mining Techniques are:

1. To apply a data mining technique to enhance government tax evasion detection performance.
2. Using a data mining technique, a screening framework is developed to filter possible non-compliant value-added tax (VAT) reports that may be subject to further auditing.
3. To explore how tax administrations could make use of data mining to enhance tax compliance among the taxpayers.
4. To encourage improvement in tax administration.
5. To avoid misrepresenting the financial fact or tax to the government.
6. To have a system that supports the tax administration of cases and tasks.

**Scope of the Project**

Tax evasion is mostly performed by the taxpayers to reduce tax liability and this illegal action is usually performed to misrepresent the financial facts to the government. This system is used to detect high risk taxpayers who avoid paying personal tax. This system uses data mining techniques. The main aim of this analysis is to explore data mining techniques for fraud tax prediction. In data mining techniques, problems like tax fraud detection are usually framed as classification problems, predicting a discrete class label output given a data observation.The main data mining techniques used for tax fraud detection are logistic models, artificial neural networks, the Bayesian network, and decision trees.

**Chapter 2**

**LITERATURE SURVEY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | **Authors** | **Advantages** | **Disadvantages** | **Result** |
| **Detection of High risk taxpayer** | John Wang | This method is used to detect high risk taxpayers in a minimum amount of time. | System output accuracy is quite less to detect the taxpayers. | Data mining technique used in this system. |
| **Data mining** | James G.S. Yang | The power of this system is the combined use of regression techniques, support vector machines and prioritizing the high-income taxpayers. | Time to predict the output is not precised. | Based on Data mining ,We come to know high risk taxpayers. |
| **Tax calculation** | R.K. Tewari | The main feature is the amount of taxable income based on which the purchasing, sales, revenue and profit can be calculated. | Include the misclassification costs. | This system decreases the number of high risk tax payers. |
| **Tax payers** | Yong Wang | The precision of this system is the use of support vector machines and prioritizing the high-income taxpayers. | System takes much time to compile the input fields | Discovering fraud tax payers with potential tax responsibility and Improving compliance of tax diductors |
| **Predictive Analytics For Controlling Tax Evasion** | Sandeep Kumar K | The logistic regression model we built predicts with high accuracy | Include the misclassification costs. | Lift chart is a measure of the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model |
| **Using data mining technique to enhance tax evasion detection performance** | Roung-Shiunn Wu a , C.S. Ou b , Hui-ying Lin b , She-I Chang b , David C. Yen c | Personnel perform their tax evasion screening tasks more efficiently, thereby enhancing the productivity of auditing possible tax evasion cases. | Basic method used to find taxpayers | They were use 3 datasets and the accuracy based on there dataset above 95% |
| **HIGH PERFORMANCE IMPLEMENTATION OF TAX FRAUD DETECTION ALGORITHM** | Mehdi Samee Rad, Asadollah Shahbahrami | Implementation results on some real data show that a performance improvement of 9.2x is achieved using available parallel patterns in .Net framework | System is required more time to run the program | Running the program in serial mode was time consuming. So in this study we tried to rely on new parallelism technology and apply it on parts of procedure to reduce the running time of the program. |
| A Comprehensive Survey of Data Mining-based Fraud Detection Research | CLIFTON PHUA, VINCENT LEE, KATE SMITH & ROSS GAYLER | This field can benefit from other related fields. Specifically, | Detection time is too long for an event-driven system | This survey clearly defines the underlying technical problems and covers more relevant fraud types, methods, and techniques than any of the other survey papers |
|  |  |  |  |  |

**Chapter 3**

**PROBLEM**

**DEFINITION**

**Problem Statement**

Government of our country suffers from lots of economic problems. Because of this high risk taxpayers. This taxpayers represent the wrong financial facts to the government. Sometimes, people purposely show their income low to pay the minimum amount of tax to the government. Government wants proper income tax on time otherwise they won't work for the country. To overcome this drawback, this system is used. This system detects the high risk taxpayers. And notify them the amount of tax after the calculating tax.

**Existing System**

In our Existing system, Based on the analysis of the reviewed articles during this space, it's able to classify tax fraud at a high-level into four major classes, namely, budget fraud, bank fraud, and different connected money fraud. It shows the amount of articles found in every form of money fraud whereas the little items of the chart represent those numbers in percentages. It's evident that budget fraud and bank fraud represent the largest portion; this proportion corresponds to forty one articles out of the sixty five reviewed articles. The projected classification framework will work as a reference in guiding money fraud. Detection analysis through providing assistance to students in distinctive areas that need additional attention. This framework can even give trade professionals associate indexes to pick out the appropriate data mining technique for a particular context of monetary fraud. For instance, firms that suffer from MasterCard fraud, they need the associate possibility of using any of the supervised learning tools (i.e. clustering, neural network, and SVM) and it's suggested to go with the foremost frequently used technique; call tree. As noted, this choice relies on the fraud context and data processing technique frequency however it will be additionally supported performance. Table seven and Chart one higher than highlight the yearly distribution of the sixty five articles across the 10- year amount. The grey highlighted years (2008, 2009, 2010 and 2011) account for quite a few publications in money fraud detection. This high rate of publications reflects a heavy growth in money fraud across industries throughout these years. Specifically, there had been a dramatic increase of the printed papers throughout 2011. This increase perceived to be a natural response to the surge of fraud activities there in year; a thirteen increase of economic fraud in 2011 compared to the previous year.

**Disadvantages of Existing System:**

1. System is not précised.
2. Detecting area is limited.

**Proposed System**

The proposed classification framework will work as a reference in guiding financial fraud detection analysis through providing the help to determine the demanding areas that require more attention. This framework can even provide business professionals an index to pick the acceptable data mining technique for a particular context of financial fraud. for instance, companies that suffer from credit card fraud, they need an choice of employing any of the supervised learning tools (i.e., naïve bayes, decision tree, neural network, and SVM) and it's suggested to travel with the most frequent used technique; decision tree. As noted, this selection is based on the fraud context and data mining technique frequency

**Advantages of Proposed System:**

1. This method is used to detect high risk taxpayers in a minimum amount of time.
2. The power of this system is the combined use of regression techniques, support vector machines and prioritizing the high-income taxpayers.
3. The main feature is the amount of taxable income based on which the purchasing, sales, revenue and profit can be calculated.
4. This system Discovers fraud tax payers with potential tax responsibility and Improving compliance of tax deductors.

**Methodology**

* machine learning classification:-

Following algorithms will be applied on features obtained above.

1. Decision tree :-

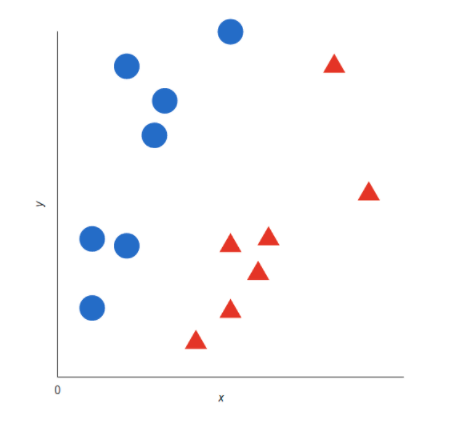
Decision tree build classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy). Leaf node (e.g., Play) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.

1. SVM (support vector machine) :-

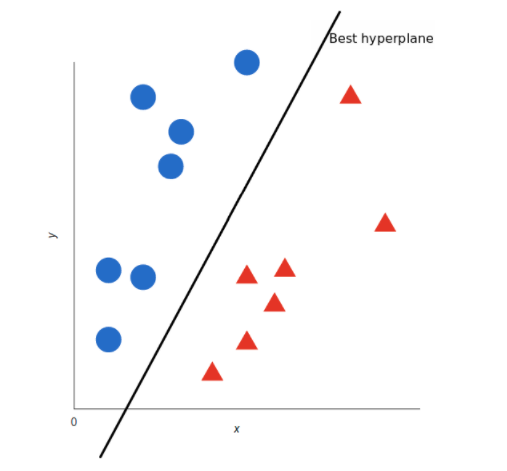
A support vector machine (SVM) may be a supervised machine learning model that uses classification, regression, and outlier’s detection algorithms. once giving an SVM model sets of labelled training data for every class, they’re ready to categorise new text. All of those are common machine learning algorithms.

SVM is associate formula that defines the most effective decision boundary between vectors that belongs to a given category. SVM is applied into each vector that encode into any kind of data which means text need to be remodelled into vector. Vectors are list of numbers that represent a group of coordinates in some space. once SVM decides to draw best decision boundary that divides the space into 2 sub spaces. thus in 2 subspaces one space is for the vectors that belong to the given class and another for the vector which do not belong to it. thus we can find vector representations that encode the maximum amount data from text.

Lets take one example,



Graph 1

The blue circle in the graph 1 are representation of training texts that say the pricing of product and red triangle training text. Then SVM draw best decision boundary line for this graph 

Graph 2

Now this is the algorithm that verify the decision boundary for the class you wish to analyse there’s a good some ways of encoding text in vectors. we only have to get the representation of all the texts you'd wish to classify and check what side of the boundary those representation fall into.

1. Multiple Logistic regression :-

[**Logistic regression**](https://www.statisticssolutions.com/academic-solutions/membership-resources/member-profile/data-analysis-plan-templates/data-analysis-plan-logistic-regression/) is the appropriate regression analysis to conduct when the dependent variable is binary. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

1. **Naive Bayes** :-

Naïve bayes is wide used machine learning classifier and probabilistic algorithmic program. Basic applications of naïve bayes are to filter spam, classify documents etc. The feature feed into the model is independent of each alternative. that is dynamical the worth of any of the other feature utilized in the algorithmic rule. Naïve bayes has important advantage is that we are able to coded up to predict the output real time fast. it's simply scalable and traditional algorithm may be a best option for real world applications that are needed to reply to user as presently as possible.

Let’s take an Example:-

You have a set of reviews and classification

|  |  |  |
| --- | --- | --- |
| Sr. No. | Text | Class |
| 1 | I loved the movie | + |
| 2 | I hated the movie | - |
| 3 | A great movie. Good movie | + |
| 4 | Poor acting | - |
| 5 | Great acting. A good movie | + |

Above table define movie review with sentiment data. In on top of table there's a text column that is input and there are 10 unique words that are: - “I, loved, the, movie, hated, a, great, poor, acting, good”. categories contain the sentiment information that's negative and positive. which define that movie review is negative or positive based on 10 unique words.

Then we've got to convert above data table into features and based on that we tend to get sentiment output. 1st we have to convert it into matrix type and also have to find how many times has that word come back.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | I | Loved | the | movie | hated | a | great | poor | acting | good | Class |
| 1 | 1 | 1 | 1 | 1 |  |  |  |  |  |  | + |
| 2 | 1 |  | 1 | 1 | 1 |  |  |  |  |  | - |
| 3 |  |  |  | 2 |  | 1 | 1 |  |  | 1 | + |
| 4 |  |  |  | 1 |  |  |  | 1 | 1 |  | - |
| 5 |  |  |  |  |  | 1 | 1 |  | 1 | 1 | + |

In above table unique words are comeback that is “I” word is continual in initial and second review then “Loved” word is repeated in exactly initial review and so on. in class column there are total 5 categories that's mixture of positive and negative classes in this there are 3 positive categories and a couple of negative categories. Then we've to count all the positive unique words. Then we've to calculate the likelihood against positive category therefore the probability is 3/5. Then we computing p(I) before that there's a formula that we've to refer that's

n = no. of words in positive class

nk = number of times word occurs in class

| vocabulary | = all unique words

We add 1 in each probability thus the chance, like P(class | text) can never be zero. we are trying to determine if a data row should be classified as negative or positive. due to these we are able to ignore the divisor. therefore we have to calculate the probabilities of every classification and therefore the probabilities of every feature falling into each classification. now we have to place values in our equation

P(I | +) = = 0.0833 P(the | +) = = 0.0833 P(a | +) = = 0.125

In P(I | +) nk 1 that is I is occurred in initial document just once. This method is comparable for negative text also. however vocabulary count is constant in both the cases. Now we have to train our classifier, for that there's one formula that is

Vnb = argmax(summation of all the words occur in sentence)

That is, suppose there's a sentence like “I hated the poor acting” then 1st we've to classify all words that is therein sentence and then we have to get the likelihood to get the class that is positive or negative. thus we grab all distinctive words and then place it in above equation.

If Vj = +; p(+)P(I|+) P(I|+) P(I|+) P(I|+) P(I|+) = 6.03x10-7

If Vj = -; p(-)P(I|-) P(I|-) P(I|-) P(I|-) P(I|-) = 1.22x10-5

So get the probability of having this sentence in positive and negative. For positive classification we get 6.03x10-7 and for negative classification we get 1.22x10-5.now we have to determine whether or not sentence is classed into positive or negative. so, the answer is negative. because of 10-5. In negative values the min negative range is greater than the most negative number so that sentence get classified into negative class. Suppose there's a maximum value that is bigger than -20 then need to take log of that number and then we have to compare both the values.

**Chapter 4**

**HARDWARE & SOFTWARE REQUIREMENT**

**Hardware and Software requirements**

**Hardware:**

1. Processor: Intel Core i3 or more.
2. RAM: 4GB or more.
3. Hard disk: 250 GB or more.

**Software:**

1. Operating System : Windows 10, 7, 8.
2. Python.
3. Anaconda.
4. Spyder, Jupyter notebook, Flask.
5. MYSQL.

**Technologies Used:-**

**Python:**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

**MySQL:**

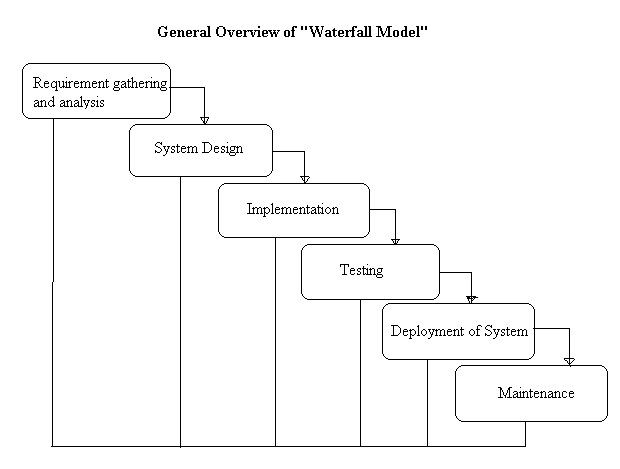
MySQL is well known as world’s most widely used open-source database (back-end). It is most supportive database for PHP as PHP-MySQL is most frequently used open-source scripting database pair. The user-interface which WAMP, LAMP and XAMPP servers provide for MySQL is easiest and reduces our work to a large extent.

**Chapter 5**

**PLANNING AND ESTIMATION**

**Software development Life Cycle**

The entire project spanned for duration of 6 months. In order to effectively design and develop a cost-effective model the Waterfall model was practiced.

****

**Requirement gathering and Analysis phase:**

This phase started at the beginning of our project, we had formed groups and modularized the project. Important points of consideration were

1. Define and visualize all the objectives clearly.

2.Gather requirements and evaluate them

Consider the technical requirements needed and then collect technical specifications of various peripheral components (Hardware) required.

3. Analyze the coding languages needed for the project.

4. Define coding strategies.

5. Analyze future risks / problems.

6. Define strategies to avoid this risks else define alternate solutions to this risks.

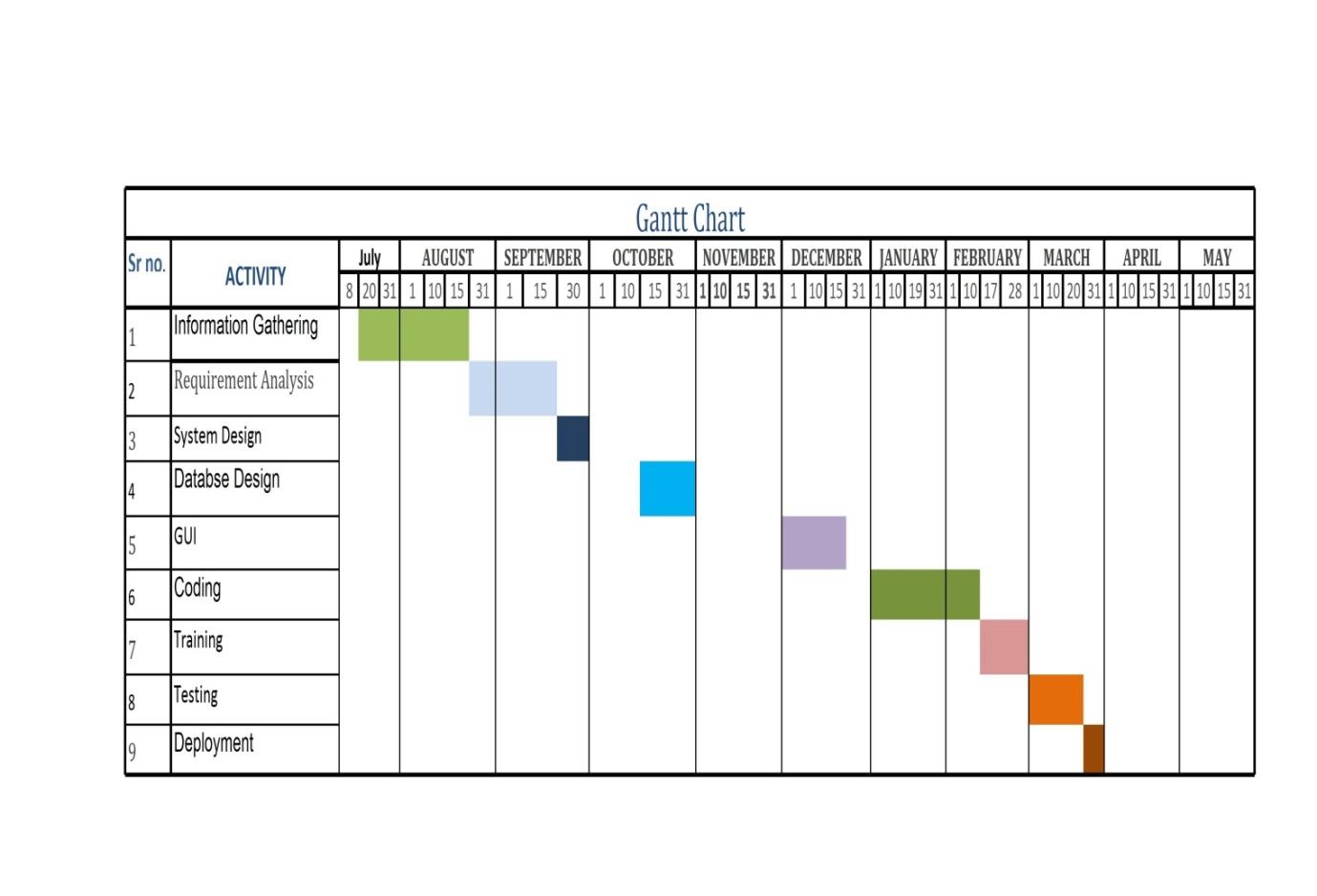
7. Check financial feasibility.

8. Define Gantt charts and assign time span for each phase.

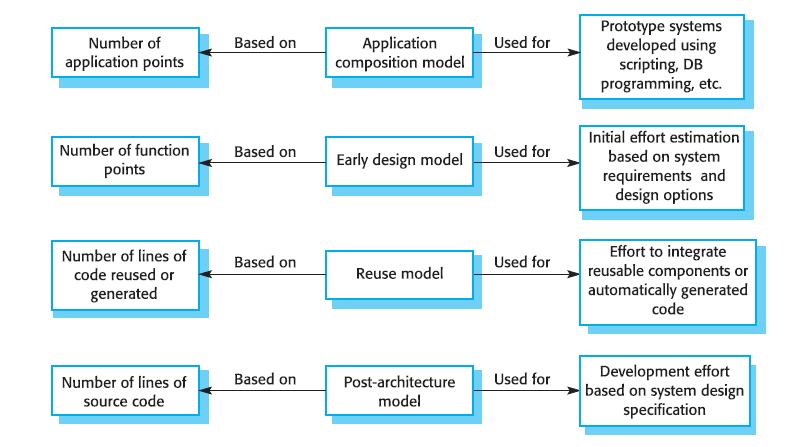
By studying the project extensively we developed a Gantt chart to track and schedule the project. Below is the Gantt chart of our project.

**Timeline**

**Please make changes as per your requirement**



**Cost Estimation**



Cost estimation is done using cocomo model

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cost Drivers | **Ratings** | | | | | |
| Very Low | Low | Nominal | High | Very High | Extra High |
| **Product attributes** |  |  |  |  |  |  |
| Required software reliability | 0.75 | 0.88 | 1.00 | 1.15 | 1.40 |  |
| Size of application database |  | 0.94 | 1.00 | 1.08 | 1.16 |  |
| Complexity of the product | 0.70 | 0.85 | 1.00 | 1.15 | 1.30 | 1.65 |
| **Hardware attributes** |  |  |  |  |  |  |
| Run-time performance constraints |  |  | 1.00 | 1.11 | 1.30 | 1.66 |
| Memory constraints |  |  | 1.00 | 1.06 | 1.21 | 1.56 |
| Volatility of the virtual machine environment |  | 0.87 | 1.00 | 1.15 | 1.30 |  |
| Required turnabout time |  | 0.87 | 1.00 | 1.07 | 1.15 |  |
| **Personnel attributes** |  |  |  |  |  |  |
| Analyst capability | 1.46 | 1.19 | 1.00 | 0.86 | 0.71 |  |
| Applications experience | 1.29 | 1.13 | 1.00 | 0.91 | 0.82 |  |
| Software engineer capability | 1.42 | 1.17 | 1.00 | 0.86 | 0.70 |  |
| Virtual machine experience | 1.21 | 1.10 | 1.00 | 0.90 |  |  |
| Programming language experience | 1.14 | 1.07 | 1.00 | 0.95 |  |  |
| **Project attributes** |  |  |  |  |  |  |
| Use of software tools | 1.24 | 1.10 | 1.00 | 0.91 | 0.82 |  |
| Application of software engineering methods | 1.24 | 1.10 | 1.00 | 0.91 | 0.83 |  |
| Required development schedule | 1.23 | 1.08 | 1.00 | 1.04 | 1.10 |  |

The Intermediate Cocomo formula now takes the form:

**E=*ai*(kloc)*(bi)*.EAF**

Using above calculation we found that the total time period of the project is around 6 months, the per month cost comes out to be Rs.12, 000/- so the total comes to be Rs.72, 000/-

**FEASIBILITY STUDY**

This system is feasible for all healthcare department like lab, hospital and clinic etc. and this system will use without experts in that field anyone will use who have knowledge about using online services which will help to use this system. Any generation people will use this system in pc.

**TECHNICAL FEASIBILITY**

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

* Technical issues raised during the investigation are:
* Does the existing technology sufficient for the suggested one?
* Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

**ECONOMIC FEASIBILITY**

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

* The costs conduct a full system investigation.
* The cost of the hardware and software.
* The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

**BEHAVIORAL FEASIBILITY**

This includes the following questions:

* Is there sufficient support for the users?
* Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

RISK ANALYSISPROCESS

Regardless of the prevention techniques employed, possible threats that could arise inside or outside the organization need to be assessed. Although the exact nature of potential disasters or their resulting consequences are difficult to determine, it is beneficial to perform a comprehensive risk assessment of all threats that can realistically occur to the organization. Regardless of the type of threat, the goals of business recovery planning are to ensure the safety of customers, employees and other personnel during and following a disaster.

The relative probability of a disaster occurring should be determined. Items to consider in determining the probability of a specific disaster should include, but not be limited to: geographic location, topography of the area, proximity to major sources of power, bodies of water and airports, degree of accessibility to facilities within the organization, history of local utility companies in providing uninterrupted services, history of the area’s susceptibility to natural threats, proximity to major highways which transport hazardous waste and combustible products. Potential exposures may be classified as natural, technical, or human threats. Examples include:

* Natural Threats: internal flooding, external flooding, internal fire, external fire, seismic activity, high winds, snow and ice storms, volcanic eruption, tornado, hurricane, epidemic, tidal wave, typhoon.
* **Technical Threats:** power failure/fluctuation, heating, ventilation or air conditioning failure, malfunction or failure of CPU, failure of system software, failure of application software, telecommunications failure, gas leaks, communications failure, nuclear fallout.
* Human Threats: robbery, bomb threats, embezzlement, extortion, burglary, vandalism, terrorism, civil disorder, chemical spill, sabotage, explosion, war, biological contamination, radiation contamination, hazardous waste, vehicle crash, airport proximity, work stoppage (Internal/External), computer crime.

All locations and facilities should be included in the risk analysis. Rather than attempting to determine exact probabilities of each disaster, a general relational rating system of high, medium and low can be used initially to identify the probability of the threat occurring. The risk analysis also should determine the impact of each type of potential threat on various functions or departments within the organization. A Risk Analysis Form, found here(PDF Format), can facilitate the process. The functions or departments will vary by type of organization. The planning process should identify and measure the likelihood of all potential risks and the impact on the organization if that threat occurred.

To do this, each department should be analyzed separately. Although the main computer system may be the single greatest risk, it is not the only important concern. Even in the most automated organizations, some departments may not be computerized or automated at all. In fully automated departments, important records remain outside the system, such as legal files, PC data, software stored on diskettes, or supporting documentation for data entry. The impact can be rated as: 0= No impact or interruption in operations, 1= Noticeable impact, interruption in operations for up to 8 hours, 2= Damage to equipment and/or facilities, interruption in operations for 8 - 48 hours, 3= Major damage to the equipment and/or facilities, interruption in operations for more than 48 hours. All main office and/or computer center functions must be relocated. Certain assumptions may be necessary to uniformly apply ratings to each potential threat.

Following are typical assumptions that can be used during the risk assessment process:

1. Although impact ratings could range between 1 and 3 for any facility given a specific set of circumstances, ratings applied should reflect anticipated, likely or expected impact on each area.

2. Each potential threat should be assumed to be “localized” to the facility being rated.

3. Although one potential threat could lead to another potential threat (e.g., a hurricane could spawn tornados), no domino effect should be assumed.

4. If the result of the threat would not warrant movement to an alternate site(s), the impact should be rated no higher than a “2.”

5. The risk assessment should be performed by facility. To measure the potential risks, a weighted point rating system can be used.

**Functional requirement**

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define *what* a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.

Functional requirements are supported by non-functional requirements (also known as *quality requirements*), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). Generally, functional requirements are expressed in the form "system must do <requirement>", while non-functional requirements are "system shall be <requirement>". The plan for implementing *functional* requirements is detailed in the system *design*. The plan for implementing *non-functional* requirements is detailed in the system *architecture*.

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the *application architecture* of a system, while non-functional requirements drive the *technical architecture* of a system.

**Non-functional requirement**

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing *functional* requirements is detailed in the system *design*. The plan for implementing *non-functional* requirements is detailed in the system *architecture*.

In general, functional requirements define what a system is supposed to *do* whereas non-functional requirements define how a system is supposed to *be*. Functional requirements are usually in the form of "system shall do <requirement>", while non-functional requirements are "system shall be <requirement>".

Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements".

**Chapter 6**

**TESTING**

**Testing**

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

#### ***Black box testing***

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

* **Specification-based testing**:

Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.Specification-based testing is necessary, but it is insufficient to guard against certain risks.

* **Advantages and disadvantages**:

The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code *must* have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers do not. *But,* on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. As a result, there are situations when (1) a tester writes many test cases to check something that could have been tested by only one test case, and/or (2) some parts of the back-end are not tested at all.

Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

#### *White box testing*

White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.

**Types of white box testing**

The following types of white box testing exist:

* API testing (application programming interface) - Testing of the application using Public and Private apis
* Code coverage - creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)
* Fault injection methods - improving the coverage of a test by introducing faults to test code paths
* Mutation testing methods
* Static testing - White box testing includes all static testing

**Code completeness evaluation**

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

Two common forms of code coverage are:

* *Function coverage*, which reports on functions executed
* *Statement coverage*, which reports on the number of lines executed to complete the test

They both return code coverage metric, measured as a percentage.

**Integration testing**

Integration testing is any type of software testing, which seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be localized more quickly and fixed.

### **Acceptance testing**

Acceptance testing can mean one of two things:

1. A smoke test is used as an acceptance test prior to introducing a new build to the main testing process, i.e. Before integration or regression.
2. Acceptance testing performed by the customer, often in their lab environment on their own HW, is known as user acceptance testing (UAT).

**Chapter 7**

**Design & Implementation**

System flowchart:

A **flowchart** is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution to a given problem. Process operations are represented in these boxes, and arrows; rather, they are implied by the sequencing of operations. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

**Start and end symbols**

-Represented as circles, ovals or rounded (fillet) rectangles, usually containing the word “Start” or “End”, or another phrase signaling the start or end of a process, such as “submit inquiry” or “receive product”.

**Arrows**

Showing “[flow of control](http://en.wikipedia.org/wiki/Control_flow)” An arrow coming from one symbol and ending at another symbol represents that control passes to the symbol the arrow points to. The line for the arrow can be solid or dashed. The meaning of the arrow with dashed line may differ from one flowchart to another and can be defined in the legend.

**Generic processing steps**

Represented as [rectangles](http://en.wikipedia.org/wiki/Rectangles) Examples: “Add 1 to X”; “replace identified part”; “save changes” or similar.

**Subroutines**

Represented as rectangles with double-struck vertical edges; these are used to show complex processing steps which may be detailed in a separate flowchart. Example: process-files. One subroutine may have multiple distinct entry points or exit flows (see [co routine](http://en.wikipedia.org/wiki/Coroutine)); if so, these are shown as labeled ‘wells’ in the rectangle, and control arrows connect to these ‘wells’.

**Input/output**

Represented as a [parallelogram](http://en.wikipedia.org/wiki/Parallelogram) Examples: Get X from the user; display X **Prepare conditional** Represented as a [hexagon](http://en.wikipedia.org/wiki/Hexagon) Shows operations which have no effect other than preparing a value for a subsequent conditional or decision step (see below).

**Conditional or decision**

Represented as a diamond ([rhombus](http://en.wikipedia.org/wiki/Rhombus)) showing where a decision is necessary, commonly a Yes/No question or True/False test. The conditional symbol is peculiar in that it has two arrows coming out of it, usually from the bottom point and right point, one corresponding to Yes or True, and one corresponding to No or False. (The arrows should always be labeled.) More than two arrows can be used, but this is normally a clear indicator that a complex decision is being taken, in which case it may need to be broken-down further or replaced with the “pre-defined process” symbol.

**Junction symbol**

Generally represented with a black blob, showing where multiple control flows converge in a single exit flow. A junction symbol will have more than one arrow coming into it, but only one going out. In simple cases, one may simply have an arrow point to another arrow instead. These are useful to represent an [iterative](http://en.wikipedia.org/wiki/Iteration) process (what in Computer Science is called a [loop](http://en.wikipedia.org/wiki/Control_flow#Loops)). A loop may, for example, consist of a connector where control first enters, processing steps, a conditional with one arrow exiting the loop, and one going back to the connector. For additional clarity, wherever two lines accidentally cross in the drawing, one of them may be drawn with a small semicircle over the other, showing that no junction is intended.

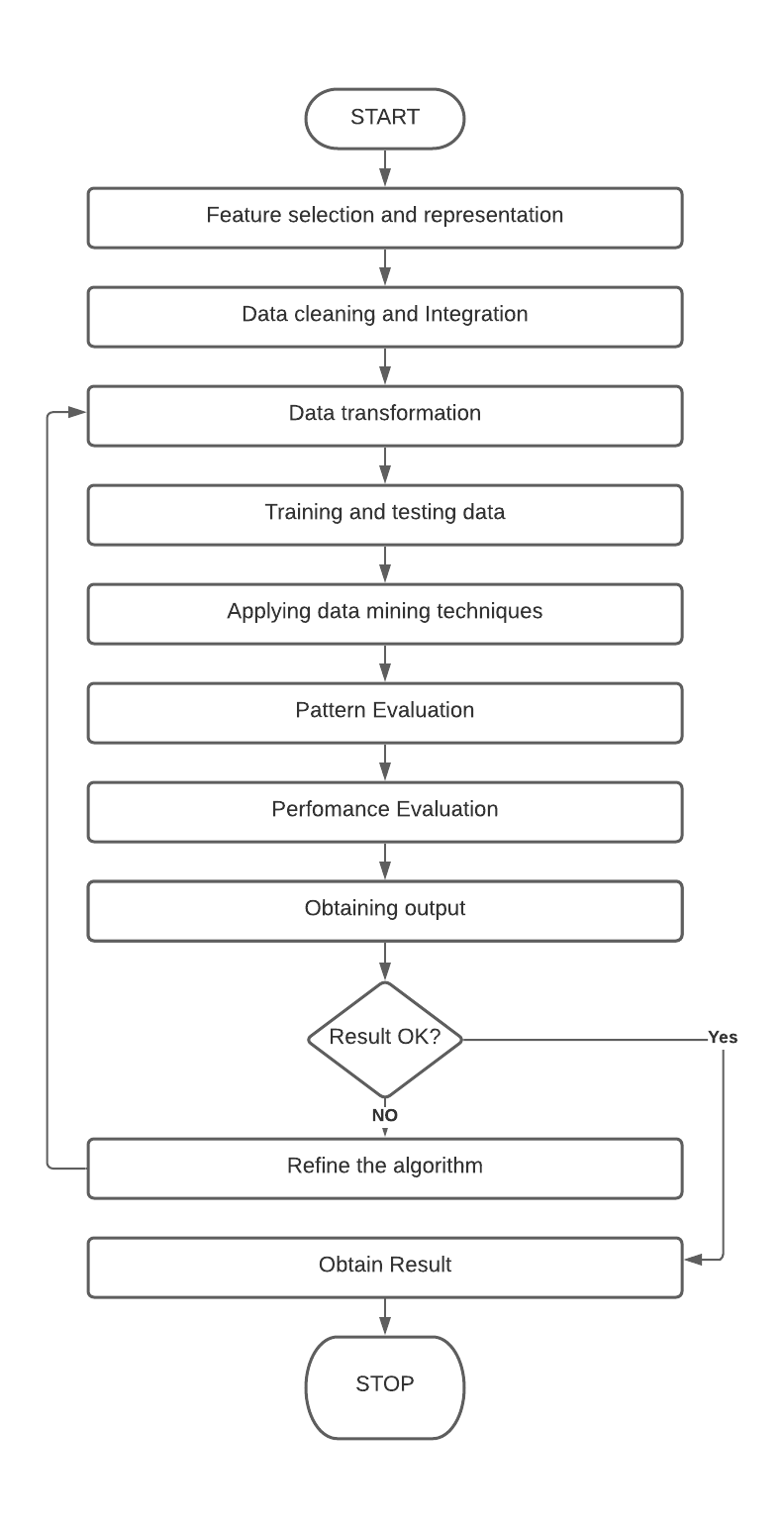
**Labeled connectors**

Represented by an identifying label inside a circle. Labeled connectors are used in complex or multi-sheet diagrams to substitute for arrows. For each label, the “outflow” connector must always be unique, but there may be any number of “inflow” connectors. In this case, a junction in control flow is implied.

**Concurrency symbol**

Represented by a double transverse line with any number of entry and exit arrows These symbols are used whenever two or more control flows must operate simultaneously. The exit flows are activated concurrently when all of the entry flows have reached the concurrency symbol. A concurrency symbol with a single entry flow is a *fork*; one with a single exit flow is a *join*. It is important to remember to keep these connections logical in order. All processes should flow from top to bottom and left to right.

**Flow chart**



**SYSTEM IMPLEMENTATION**

Implementation includes all those activities that take place to convert from the old system to the new. The old system consists of manual operations, which is operated in a very different manner from the proposed new system. A proper implementation is essential to provide a reliable system to meet the requirements of the organizations. An improper installation may affect the success of the computerized system.

**IMPLEMENTATION METHODS:**

There are several methods for handling the implementation and the consequent conversion from the old to the new computerized system.

The most secure method for conversion from the old system to the new system is to run the old and new system in parallel. In this approach, a person may operate in the manual older processing system as well as start operating the new computerized system. This method offers high security, because even if there is a flaw in the computerized system, we can depend upon the manual system. However, the cost for maintaining two systems in parallel is very high. This outweighs its benefits.

Another commonly method is a direct cut over from the existing manual system to the computerized system. The change may be within a week or within a day. There are no parallel activities. However, there is no remedy in case of a problem. This strategy requires careful planning.

A working version of the system can also be implemented in one part of the organization and the personnel will be piloting the system and changes can be made as and when required. But this method is less preferable due to the loss of entirety of the system.

**IMPLEMENTATION PLAN:**

The implementation plan includes a description of all the activities that must occur to implement the new system and to put it into operation. It identifies the personnel responsible for the activities and prepares a time chart for implementing the system. The implementation plan consists of the following steps.

* List all files required for implementation.
* Identify all data required to build new files during the implementation.
* List all new documents and procedures that go into the new system.

The implementation plan should anticipate possible problems and must be able to deal with them. The usual problems may be missing documents; mixed data formats between current and files, errors in data translation, missing data etc.

**DFD**

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then exploded to show more detail of the system being modeled.

**Symbols:**

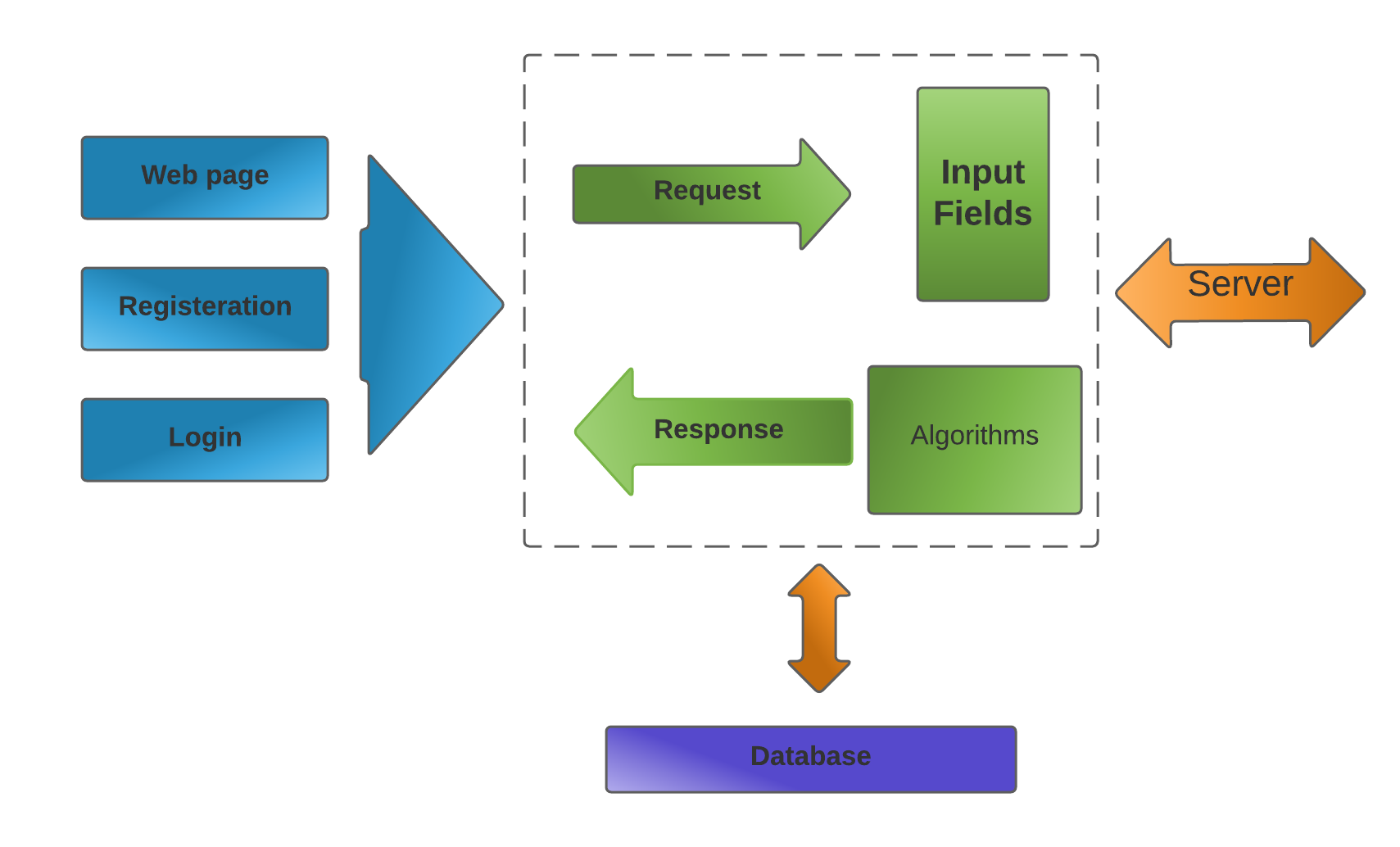
**The four components of a data flow diagram (DFD) are:**

*\_* External Entities/Terminators are outside of the system being modeled. Terminators represent where information comes from and where it goes. In designing a system, we have no idea about what these terminators do or how they do it.

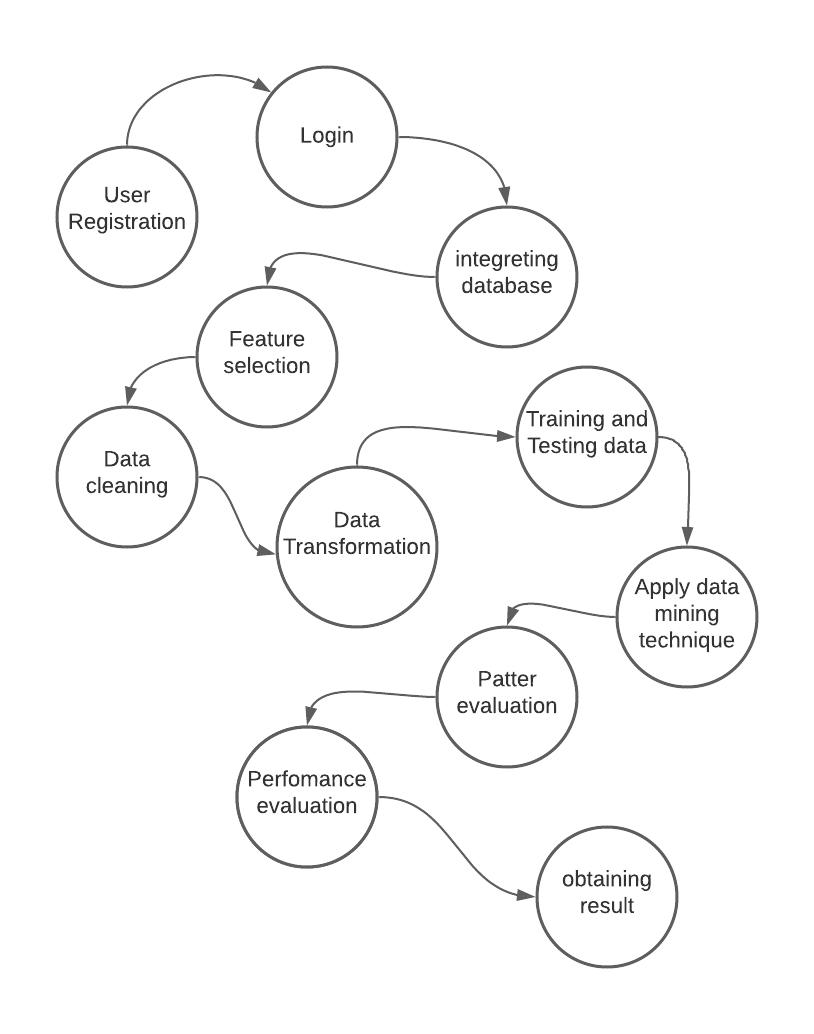
*\_* Processes modify the inputs in the process of generating the outputs

*\_* Data Stores represent a place in the process where data comes to rest. A DFD does not say anything about the relative timing of the processes, so a data store might be a place to accumulate data over a year for the annual accounting process.

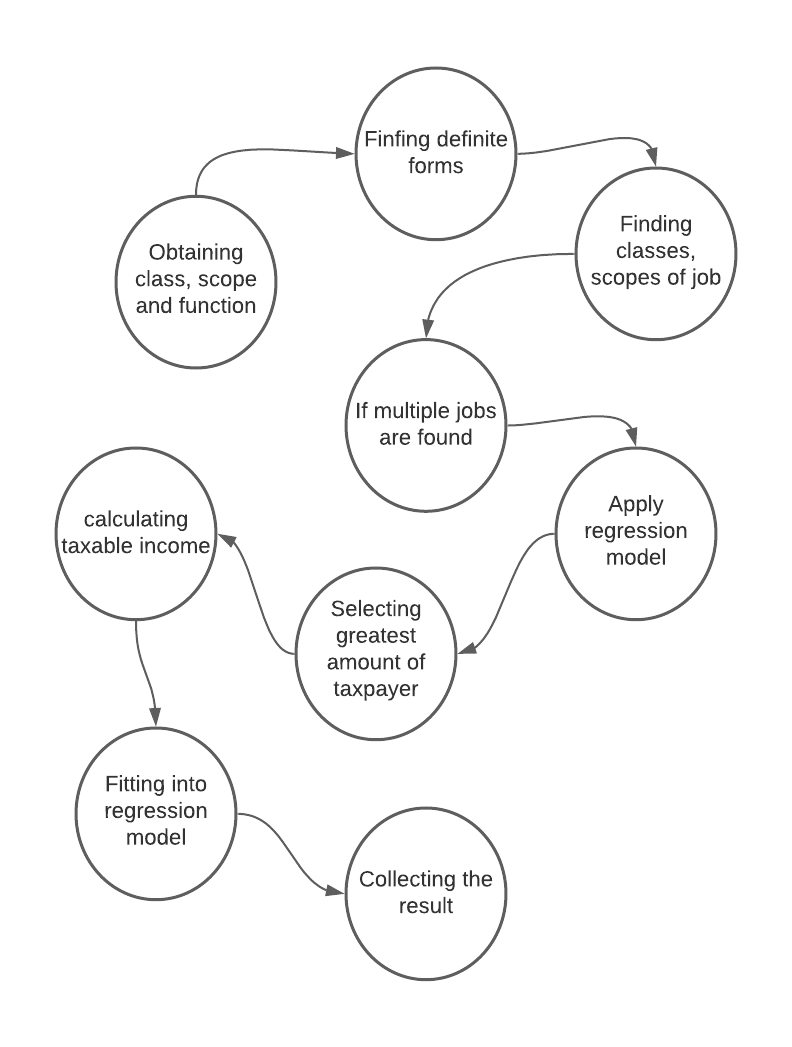
**Architecture Diagram:**



**DFD level 1**

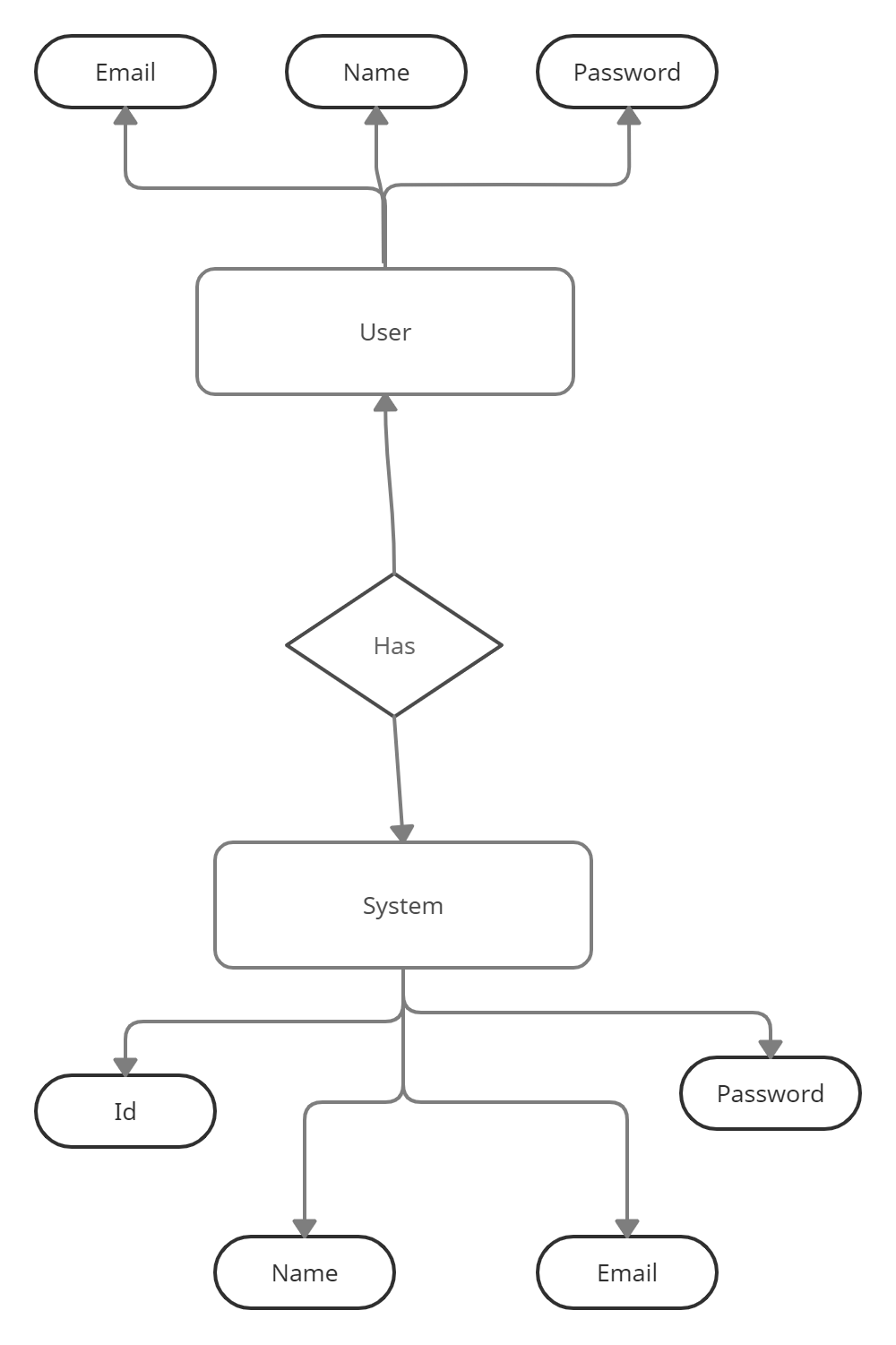


**DFD level 2:**

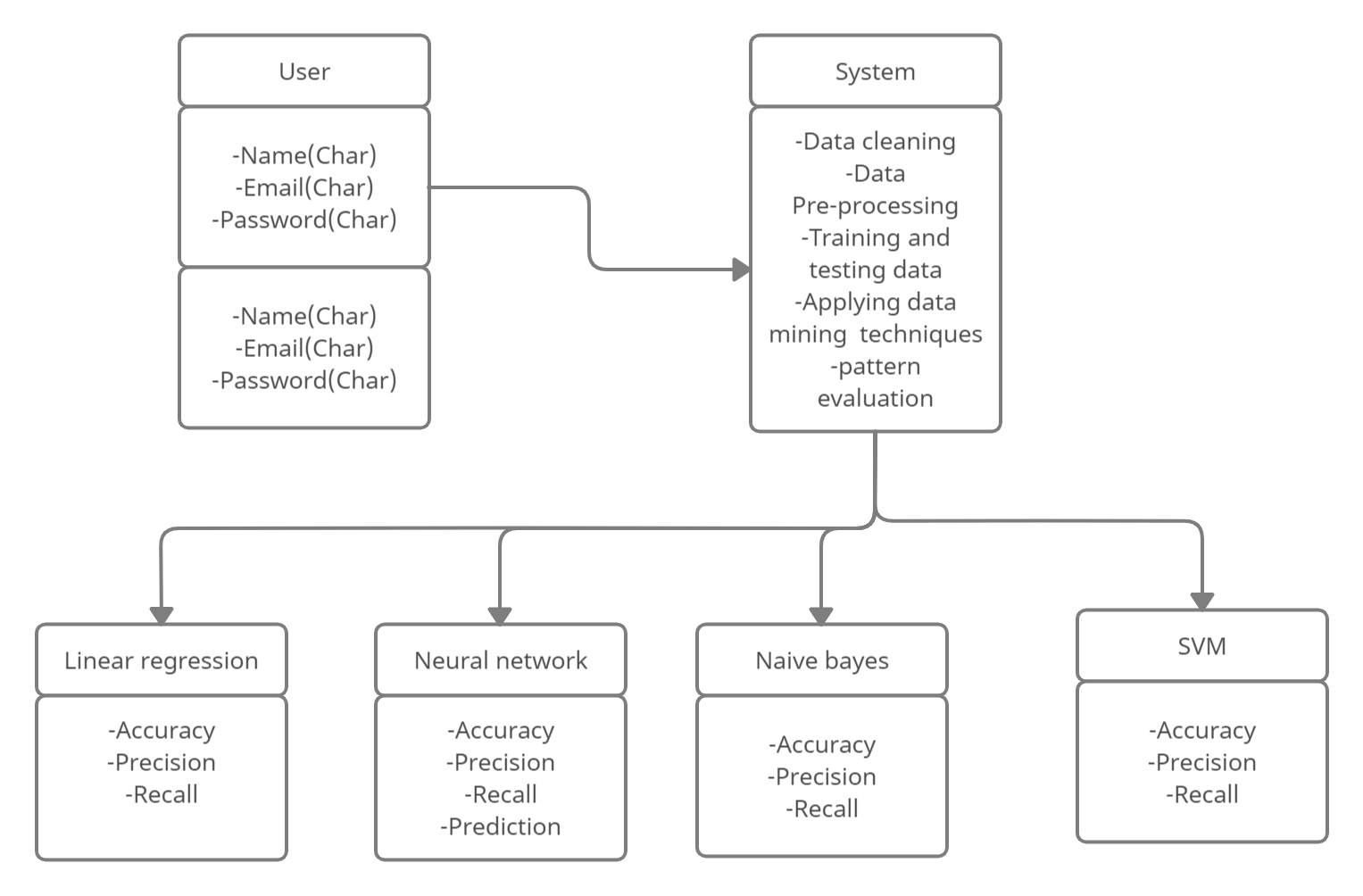




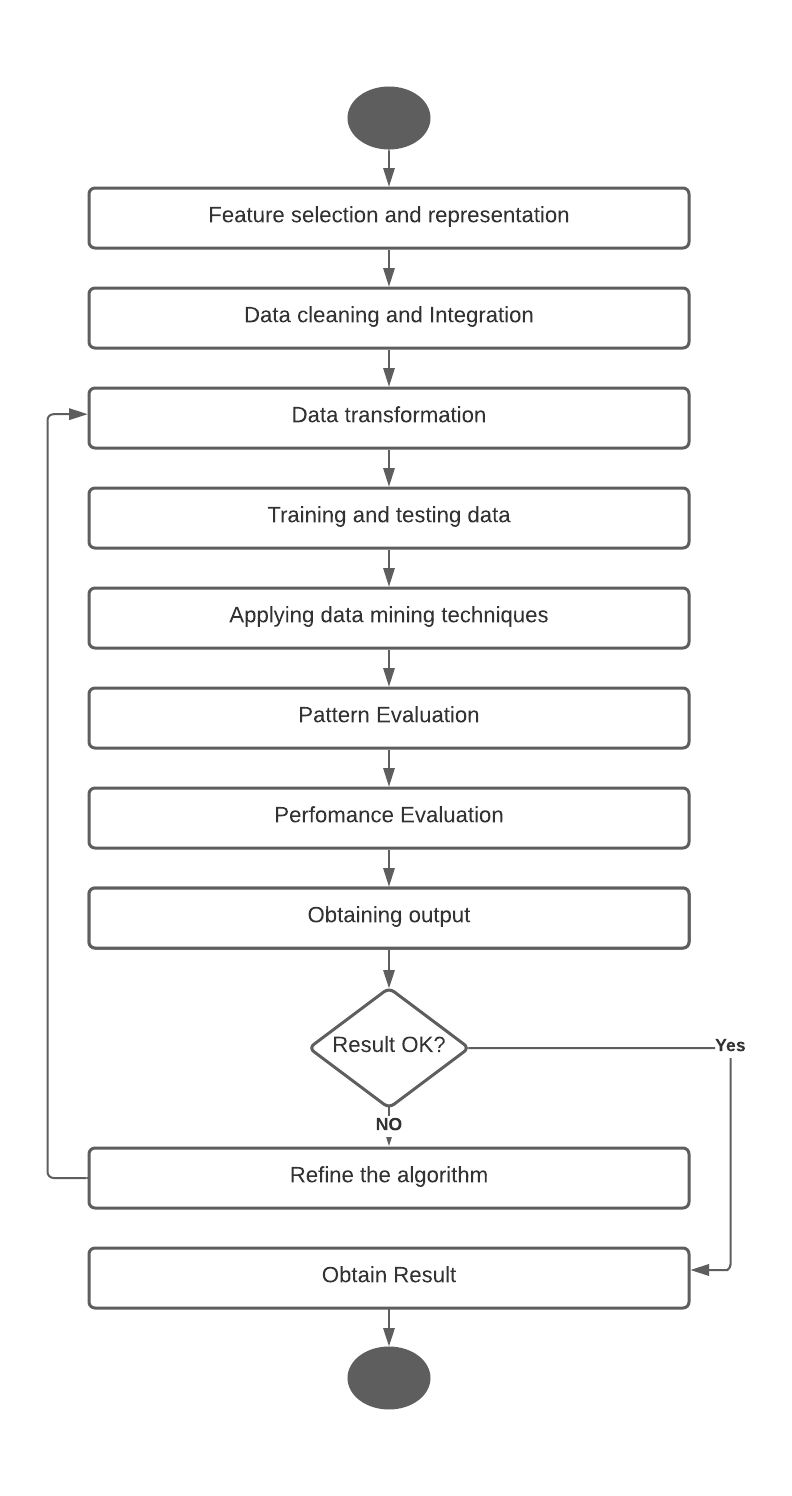
**E-R Diagram:**



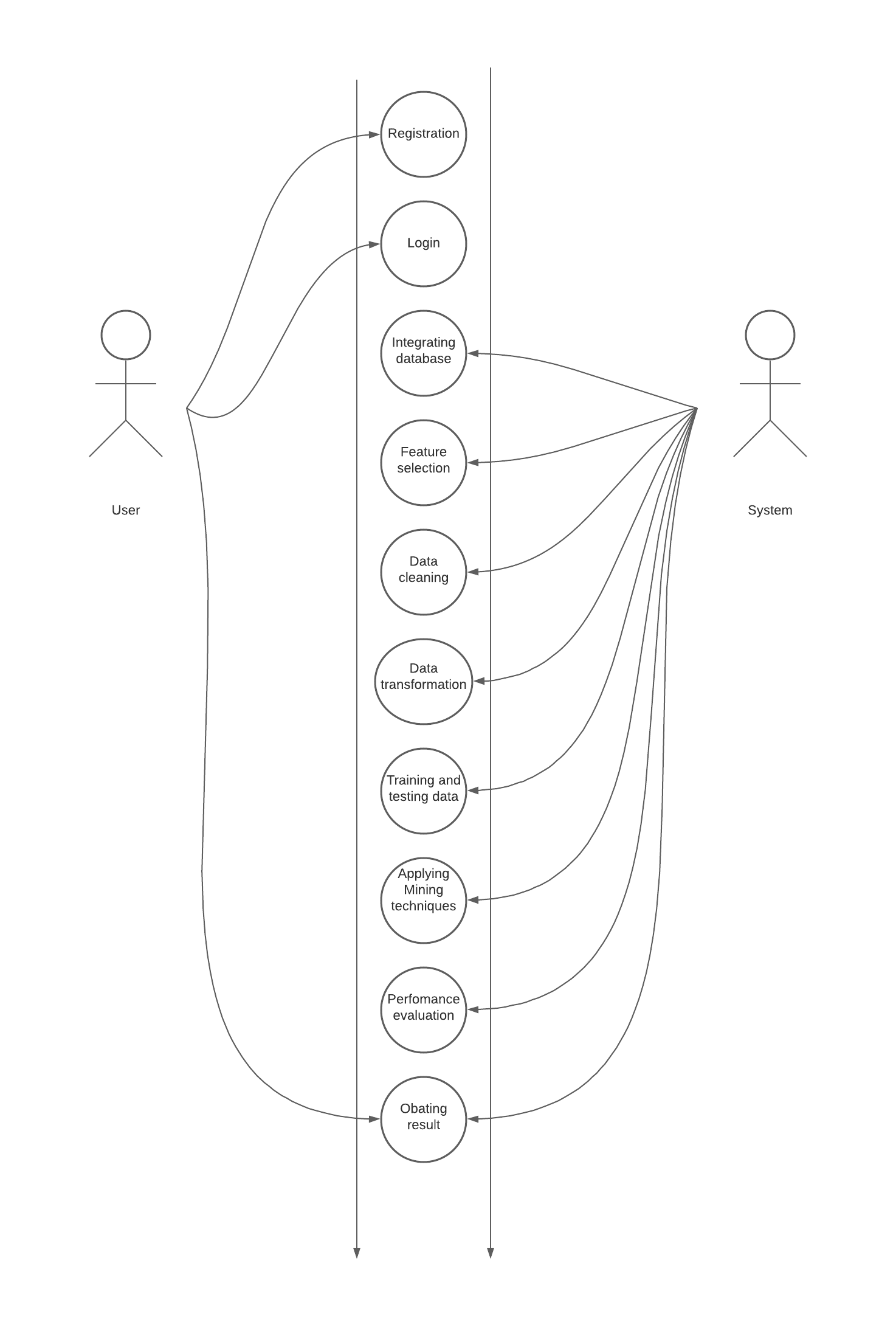
**Class Diagram:**

****

**Activity Diagram :**



**Use Case Diagram :-**



**Chapter 8**

**Advantages**

**Advantages:**

1. This method is used to detect high risk taxpayers in a minimum amount of time.
2. The power of this system is the combined use of regression techniques, support vector machines and prioritizing the high-income taxpayers.
3. The main feature is the amount of taxable income based on which the purchasing, sales, revenue and profit can be calculated.
4. This system Discovers fraud tax payers with potential tax responsibility and Improving compliance of tax deductors.

**Chapter 09**

**FUTURE MODIFICATIONS**

**&**

**CONCLUSION**

**Future Modification**

Tax evasion is mostly performed by the taxpayers to reduce tax liability and this illegal action is usually performed to misrepresent the financial facts to the government. This system is used to detect high risk taxpayers who avoid paying personal tax. This system uses data mining techniques. The main aim of this analysis is to explore data mining techniques for fraud tax prediction. In data mining techniques, problems like tax fraud detection are usually framed as classification problems, predicting a discrete class label output given a data observation. The main data mining techniques used for tax fraud detection are logistic models, artificial neural networks, the Bayesian network, and decision trees.

**Conclusion**

Various strategies are enforced to extract the high risk taxpayers however in apply the colorful taxpayers’ formula given the most effective result compared to the mean variance, job factor, quantity based and modification based methods. The facility of this technique is that the combined use of regression techniques, support vector machine and prioritizing the high-income taxpayers.

**Chapter 10**

**BIBILIOGRAPHY**

**References**

[1] P. Ravisankar, V. Ravi, G.R. Rao, I. Bose, "Detection of financial statement fraud and feature selection using data mining technique," Decision Support Systems vol.50, no. 2, pp. 491-500, (2011).

[2] E.W.T. Ngai, Y. Hu, Y.H. Wong, Y. Chen, X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," Decision Support Systems vol.50, no. 3, pp. 559-569, (2011).

[3] P.C. González, J.D. Velásquez, "Characterization and detection of taxpayers with false invoices using data mining techniques," Expert Systems with Applications vol.40, no. 5, pp. 1427-1436, (2013).

[4] Central Michigan University, “Computational Data Mining Techniques in Automotive Insurance Fraud Detection”; Data Science Journal 2012.

[5]. B. Jyothi, Y. M. Latha, and V. S. K. Reddy, “Medical image retrieval using multiple features,” Research India Publications, vol. 3, pp. 387–396, 2010. Available: searchdl.org/public/book\_series/AETS/7/546.pdf

[6] Z. Q. Song, H. Cui, and Y. A. Hu, “Research and development of support vector machine theory,” Journal of Naval Aeronautical Engineering Institute, vol. 23, no. 2, pp.143-148, 2008.

[ 7] Nazia Fathima S.M (2013) Classification of blood type by microscopic color images, International Journal of Machine Learning and Computing. Vol. 3, No. 4 Available: www.ijmlc.org/papers/342-L472.pdf

[ 8] Priyadharshini, Ramya, Kalaiyarasi, Kalpana Devi., SuthathiraVanitha.

A NOVEL APPROACH IN IDENTIFICATION OF BLOOD GROUP USING LASER TECHNOLOGY.

International Journal of Research in Engineering and Technology Volume: 03 Special Issue: 11 Available: www.ijret.org

[ 9] Mäenpää, Topi The local binary pattern approach to texture analysis-Extension and application

Available:http://urn.fi/urn:isbn:9514270762

**Chapter 11**

**SCREENSHOTS**

**Chapter 12**

**SOURCE CODE**