**ST. XAVIER'S COLLEGE, AUTONOMOUS**

**KOLKATA**

**30, MOTHER TERESA SARANI,**

**KOLKATA 700016**

**Group Formation in Collaborative Learning**

ARKIT SEN - **2-72-14-0013**

**Under Guidance of: Prof. Anal Acharya**

**Date: 19th May 2016**

**Submitted to the Department of Computer Science in partial fulfilment of the**

**requirements for the degree of M.Sc.**



***1***

**CERTIFICATE OF AUTHENTICATED WORK**

This is to certify that the project report entitled submitted to Department of Computer Science , ST. XAVIER'S COLLEGE [AUTONOMOUS], KOLKATA, in partial fulfilment of the requirement for the award of the degree of MASTER OF SCIENCE (M.SC.) is an original work carried out by

**Mr. Arkit Sen**,………………………...Registration no. : **A01-1112-0105-14**

under my guidance. The matter embodied in this project is authentic and is genuine work done by the student and has not been submitted whether to this College or to any other Institute for the fulfilment of the requirement of any

|  |  |
| --- | --- |
| course of study. |  |
|  |  |
|  |  |
|  |  |
| ………………………………… | ...…………………………………. |
| Signature of the Student: | Signature of the Professor |
| Date: **19TH** **May 2016** | Date: **19TH** **May 2016** |
|  | **Prof. Anal Acharya** |
|  | Name, Designation and Address |
| **Arkit Sen** | of the Professor |

Name and Address of the student



***2***

**ABSTRACT:**

A class consists of a variety of students, some may score quite well in exam some may not, but this should not be considered as constrains in their process of learning and gaining knowledge. It is observed that students of similar academic merits, group together and thus forming homogeneous clusters. To let students help each other for their own development and growth, heterogeneous groups should be created. In this project groups of students are formed with dissimilar academic merits. From the given set of data of small clusters with similar attributes, heterogeneous groups are created.

**ACKNOWLEDGEMENT:**

The very idea of this project was devised by our Project Guide respected Professor Anal Acharya, who introduced us to this wonderful concept of group formation with the help of a clustering algorithm and something which can actually be implemented in real life, benefiting many students in the long run. The novel approach which is followed and is reflected in this project could have been materialized because of his patience and guidance from the beginning, which helped us to motivate, nurture ourselves better. Thus we collectively thank him for his guidance and the endeavor that has been put into this Project with us.



***4***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | **TABLE OF CONTENTS** |  |
| **1.** | INTRODUCTION | | | 09 |
|  | 1.1. | Background | | 09 |
|  | 1.2. | Objectives | |  |
|  | 1.3. | Purpose, Scope, and Applicability | | 10 |
|  | 1.4. | Category of the project | | 11 |
|  | 1.5. | Clustering | | 11 |
|  | 1.6. | Requirements for clustering | | 12 |
|  | 1.7. | Overview of Basic Clustering Methods: | | 13 |
|  |  | 1.7.1. | Hierarchical methods: | 14 |
|  |  | 1.7.2. | Density-based methods: | 15 |
|  |  | 1.7.3. | Grid-based methods: | 15 |
|  |  | 1.7.4. | Partitioning methods | 17 |
|  |  | 1.7.4.1.K-Means: A Centroid Based Technique | | 18 |
|  |  | 1.7.4.2.How does the K-means algorithm work? | | 18 |
|  |  | 1.7.4.3.Activity Diagram: | | 22 |
| 2. | SURVEY OF TECHNOLOGIES | | | 24 |
| 3. | REQUIREMENTS AND ANALYSIS | | | 27 |
|  | 3.1. | Problem definition | | 28 |
|  | 3.2. | Requirements specification | | 29 |
|  | 3.3. | Planning and scheduling | | 29 |
|  | 3.4. | Software and hardware requirements | | 30 |
|  |  | 3.4.1. | The r environment | 30 |
|  | 3.5. | Preliminary product description | | 31 |
|  | 3.6. | Gnat chart | | 32 |
|  | 3.7. | Pert chart | | 32 |
|  | 3.8. | Conceptual models | | 33 |
|  |  | 3.8.1. | Module 1 | 33 |
|  |  | 3.8.2. | Module 2 | 34 |
|  |  | 3.8.3. | Module 3 | 34 |
|  |  | 3.8.4. | Module 4 | 35 |
| 4. | SYSTEM DESIGN | | | 37 |
|  | 4.1. | Schema Design | | 39 |
|  | 4.2. | Module 1 | | 40 |
|  | 4.3. | Module 2 | | 42 |
|  | 4.4. | Module 3 | | 46 |
|  | 4.5. | Module 4 | | 49 |
|  | 4.6. | User Interface Design | | 52 |
|  | 4.7. | Security issues | | 55 |
|  | 4.8. | Test case design | | 55 |



***5***

|  |  |  |
| --- | --- | --- |
| 5. IMPLEMENTATION AND TESTING | | 56 |
| 5.1. | Coding Details | 88 |
| 5.2. | Code Efficiency | 88 |
| 5.3. | Testing Approach | 88 |
| 5.4. | Unit Testing | 88 |
| 5.5. | Integrated Testing | 88 |
| 5.6. | Modifications and Improvements | 89 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6. | RESULTS AND DISCUSSION | |  | 90 |
|  | 6.1. | Test Reports |  | 90 |
|  | 6.2. | Test Output |  | 90 |
|  | 6.3. | User Data Set 1 |  | 94 |
|  | 6.4. | User Data Set 2 |  | 98 |
|  | 6.5. | Discussion |  | 103 |
|  | 6.6. | User Documentation |  | 106 |
|  |  | 6.6.1. | Module 1 | 106 |
|  |  | 6.6.2. | Module 2 | 107 |
|  |  | 6.6.3. | Module 3 | 108 |
|  |  | 6.6.4. | Module 4 | 108 |
| 7. | CONCLUSIONS | |  | 109 |
|  | 7.1. | Conclusion |  | 109 |
|  | 7.2. | Limitations of the System |  | 110 |
|  | 7.3. | Future Scope of the Project |  | 110 |
| 8. | REFERENCES | |  | 111 |

APPENDIX I



***6***

2. CHAPTER 2: SURVEY OF TECHNOLOGIES

2.1. Comparison of features of various algorithm

3. CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1. Plan of project

3.2. Gnat chart

3.3. Pert chart

3.4. Conceptual model

3.4.1. Module 1

3.4.2. Module 2

3.4.3. Module 3

3.4.4. Module 4

4. CAHPTER 4: SYSTEM DESIGN

4.1. Schema diagram

4.2. DFD for the Module 01

4.3. Flow chart for Module 1

4.4. DFD of the Module 02

4.5. Flowchart: module 02

4.6. A formal flowchart and algorithm of K means.

4.7. DFD for the MODULE 03

4.8. Flow Chart module 03

4.9. DFD of the final module 4.

4.10.

Flowchart module 04

4.11.

The Basic Front End 1 design

4.12.

Insertion into the Database.

4.13.

Deletion into the Database

4.14.

Updating the data into the database

**TABLE OF FIGURES**

**1. CHAPTER 1: INTRODUCTION**

**1.1. Agglomerative and Divisive hierarchical clustering.**

**14**

**1.2. Dendrogram Representation for hierarchical clustering**

**14**

**1.3. Density based Clustering Example.**

**15**

**1.4. Grid based Clustering Example**

**16**

**1.5. Finding spherical-shaped clusters in small- to medium-size databases 17**

**1.6. Clustering using Partition Method (K-means).**

**18**

**1.7. The Clustered groups**

**21**

26

29

32

32

33

34

34

35

38

39

40

42

43

45

47

47

50

50

52

52

53

53

****

***7***

|  |  |  |
| --- | --- | --- |
| 5. CAHPTER 5: IMPLEMENTATION AND TESTING | |  |
| 5.1. | Front End Tool | 56 |
| 5.2. | Inserting record | 57 |
| 5.3. | Tool after successful insertion | 58 |
| 5.4. | Deletion of record | 60 |
| 6. CHAPTER 6: RESULTS AND DISCUSSION | |  |
| 6.1. | The Equalization process done through the Front End 2 | 93 |
| 6.2. | Table containing Data | 94 |
| 6.3. | Description of the Table STUDENT | 95 |
| 6.4. | The insertion of Data in the DB via the Front End 1. | 95 |
| 6.5. | The running of the script in the R programming language | 96 |
| 6.6. | The Unequalled Cluster formed by the R. | 96 |
| 6.7. | The Equalization Process through the Front End 2, Module 03. | 97 |
| 6.8. | The end result with equaled clusters. | 98 |
| 6.9. | Table contain the set of data, entered into the DB. | 99 |

|  |  |  |
| --- | --- | --- |
| 6.10. | Insertion of data into DB, Module 01. | 100 |
| 6.11. | Processing of data done in R, Module 2 | 100 |
| 6.12. | Clusters formed but unequalized | 101 |
| 6.13. | The Front End 2 accepting the cluster details to Equalize | 102 |
| 6.14. | Bar graphs of USER DATA SET 2, X vs. Y | 106 |
| 6.15. | The front end 1, simply enter Roll no. In roll no field and the two | |
| scores in the respective fields. | | 107 |
| 6.16. | The Front End 2 design the Equalization phase | 108 |
| 6.17. | Likewise MS. Excel chart for a comparison | 108 |



***8***

**CHAPTER 1**

1. **INTRODUCTION**

**1.1. BACKGROUND**

A group by definition means some people who come together on a free and voluntary basis, with a spirit of co-operation to work together for mutual social and economic benefit. A group is needed for

1. To organize and guide an action,
2. For training members in necessary skills,
3. To provide a channel for information sharing.[1]

Collaborative learning is a situation in which two or more people learn or attempt to learn something together. Unlike individual learning, people engaged in collaborative learning capitalize on one another’s resources and skills (asking one another for information, evaluating one another’s ideas, monitoring one another’s work, etc.).[1] Put differently, [collaborative](https://en.wikipedia.org/wiki/Collaboration) learning refers to [methodologies and environments](https://en.wikipedia.org/wiki/Collaborative_method) in which learners engage in a common task where each individual depends on and is accountable to each other. Alternatively, collaborative learning occurs when individuals are actively engaged in a community in which learning takes place through explicit or implicit collaborative efforts. [2]

**1.2. OBJECTIVES**

Our Objective is Formation of groups/teams to implement Collaborative Learning among students. Aim is to make the team heterogeneous. The unfairness of forming a group with only weak students is obvious, but groups with only strong students are equally undesirable. In heterogeneous groups, the weaker students gain from seeing how the better students approach the problems



***9***

and the stronger students gain a deeper understanding of the subject by teaching it to the others. [2]

Cooperative learning is generally defined as a teaching arrangement in which small, heterogeneous groups of students work together to achieve a common goal. Students encourage and support each other, assume responsibility for their own and each other's learning, employ group related social skills, and evaluate the group's progress. The basic elements are positive interdependence, equal opportunities, and individual accountability. Human beings are social creatures by nature and cooperation has been used throughout history in all aspects of our lives. Therefore, it follows that cooperative learning groups in schools would be used as a logical teaching method. [3]

**1.3.** **PURPOSE AND SCOPE**

*PURPOSE:*

Collaboration

* Mutual engagement of participants
* A coordinated effort to solve the problem
* Continuous shared conception of the problem.

Cooperation

* Division of labour
* Individual responsibility for sections
* Coordination when assembling partial results. [4]

The method of implementing this process of study is to ensure that all students who may not be at the same level academically can cooperate with each other and learn from mutual sharing of knowledge.



***10***

This project implements this very technique with the help of formation of groups and clustering the data and then analysing them. [4]

***SCOPE:***

The Project works on the sample data that is being noted beforehand and works on formation of heterogeneous groups consisting of pupils in small numbers. It then applies certain algorithms on the collected data and takes out the requisite result from it.

**1.4. CATEGORY OF THE PROJECT**

The project belongs to the category of Artificial Intelligence (AI) and Computational Intelligence.

Artificial intelligence (AI) is the intelligence exhibited by machines or software. It is also the name of the academic field of study which studies how to create computers and computer software that are capable of intelligent behavior. Major AI researchers and textbooks define this field as "the study and design of intelligent agents", in which an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success.[6] John McCarthy, who coined the term in 1955, defines it as "the science and engineering of making intelligent machines" [6]

Computational Intelligence is a set of nature-inspired computational methodologies and approaches to address complex real-world problems to which mathematical or traditional modelling can be useless for a few reasons: the processes might be too complex for mathematical reasoning

**1.5.** **CLUSTERING**

Cluster Analysis or simple clustering is the process of partitioning a set of data objects into subsets. Each subset is a cluster such that objects in a cluster are similar to one another, yet dissimilar to objects on other clusters. Clustering analyzes data objects without consulting class labels. In many cases, class-labeled data may not exist at the beginning.

Basic Clustering Methods:



***11***

1. Partitioning Methods.
2. Hierarchical Methods.
3. Density Based Methods.
4. Grid Based Methods.

Here we have used a specific kind of Partitioning Method of clustering known as “**K-Means**” which is a centroid based technique. An objective function is used to assess the partitioning quality so that similar objects within a cluster are similar to one another but dissimilar to objects in other clusters. The objects functions aims for high intra cluster similarity and low inter cluster similarity.

**1.6.** **REQUIREMENTS FOR CLUSTERING**

Typical Requirements of clustering in data mining:

1. Scalability: Clustering on only a sample of given large data may lead to biased results. Therefore, highly scalable clustering algorithms are needed.[5]
2. Ability to deal with different types of attributes: Many algorithms are designed to cluster numeric data. Applications may require clustering other data types, such as binary, nominal and ordinary data. Clustering data for complex data types such as graphs, sequences, images and documents.[5]
3. Discovery of clusters with arbitrary shape: Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures. Such algorithms find spherical clusters with similar size and density. However, a cluster could be of any shape. We may want to use clustering to find the frontier of a running forest fire, which is often not spherical. It is important to detect clusters of arbitrary shape.[5]



***12***

1. Requirements for domain knowledge to determine input parameters: Many clustering algorithms require users to provide domain knowledge in the form of input parameters such as desired number of clusters. Clustering maybe sensitive to such parameters.[5]
2. Ability to deal with noisy data: Clustering algorithms can be sensitive to noise and may produce poor-quality clusters. Therefore, we need clustering methods that are robust to noise.[5]
3. Incremental clustering and insensitivity to input order: Clustering may also be sensitive to input data order. That is given a set of data objects, clustering algorithms may return dramatically different clustering depending on the order in which the objects are presented.[5]
4. Capability of clustering high-dimensional data: The most clustering algorithms are good at handling low-dimensional data such as data sets involving only two or three dimensions. Finding clusters of data objects in high dimensional space is challenging, considering such data can be very sparse and highly skewed.[4]
5. Interpretability and usability: Users want clustering results to be interpretable, comprehensive and usable. Clustering needs to be tied in with specific semantic interpretations and applications.[4]

**1.7. Overview of Basic Clustering Methods:**

There are many clustering algorithm in the literature. In general the major fundamental clustering methods can be classified into following categories, which are discussed below:



***13***

**1.7.1.** **Hierarchical methods:**

A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either *agglomerative* or *divisive*, based on how the hierarchical decomposition isformed. The *agglomerative approach*, also called the *bottom-up* approach, starts with each object forming a separate group. It successively merges the objects or groups close to one another, until all the groups are merged into one (the topmost level of the hierarchy), or a termination condition holds.

The *divisive approach*, also called the *top-down* approach, starts with all the objects in the same cluster. In each successive iteration, a cluster is split into smaller clusters, until eventually each object is in one cluster, or a termination condition holds. [10][11]

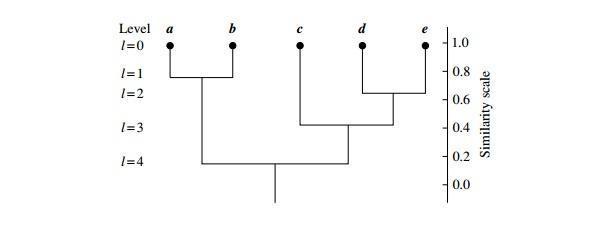


Fig: Agglomerative and Divisive hierarchical clustering. [10].

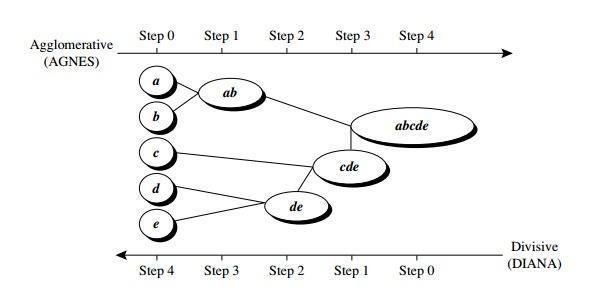


Fig: Dendrogram Representation for hierarchical clustering. [10].



***14***

**1.7.2.** **Density-based methods:**

Most partitioning methods cluster objects based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty in discovering clusters of arbitrary shapes. Other clustering methods have been developed based on the notion of density. Their general idea is to continue growing a given cluster as long as the density (number of objects or Data points) in the “neighborhood” exceeds some threshold. For example, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points. Such a method can be used

to filter out noise or outliers and discover clusters of arbitrary shape. [10][11]

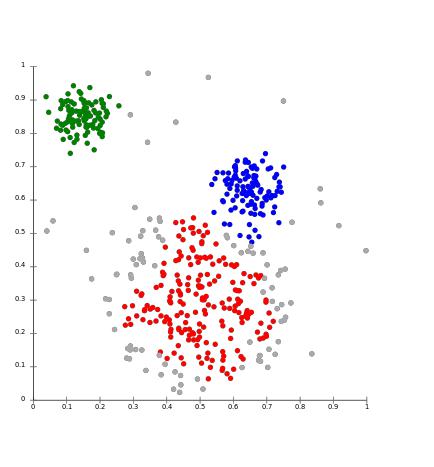


Fig:

Density based Clustering Example. [12].

**1.7.3.** **Grid-based methods:**

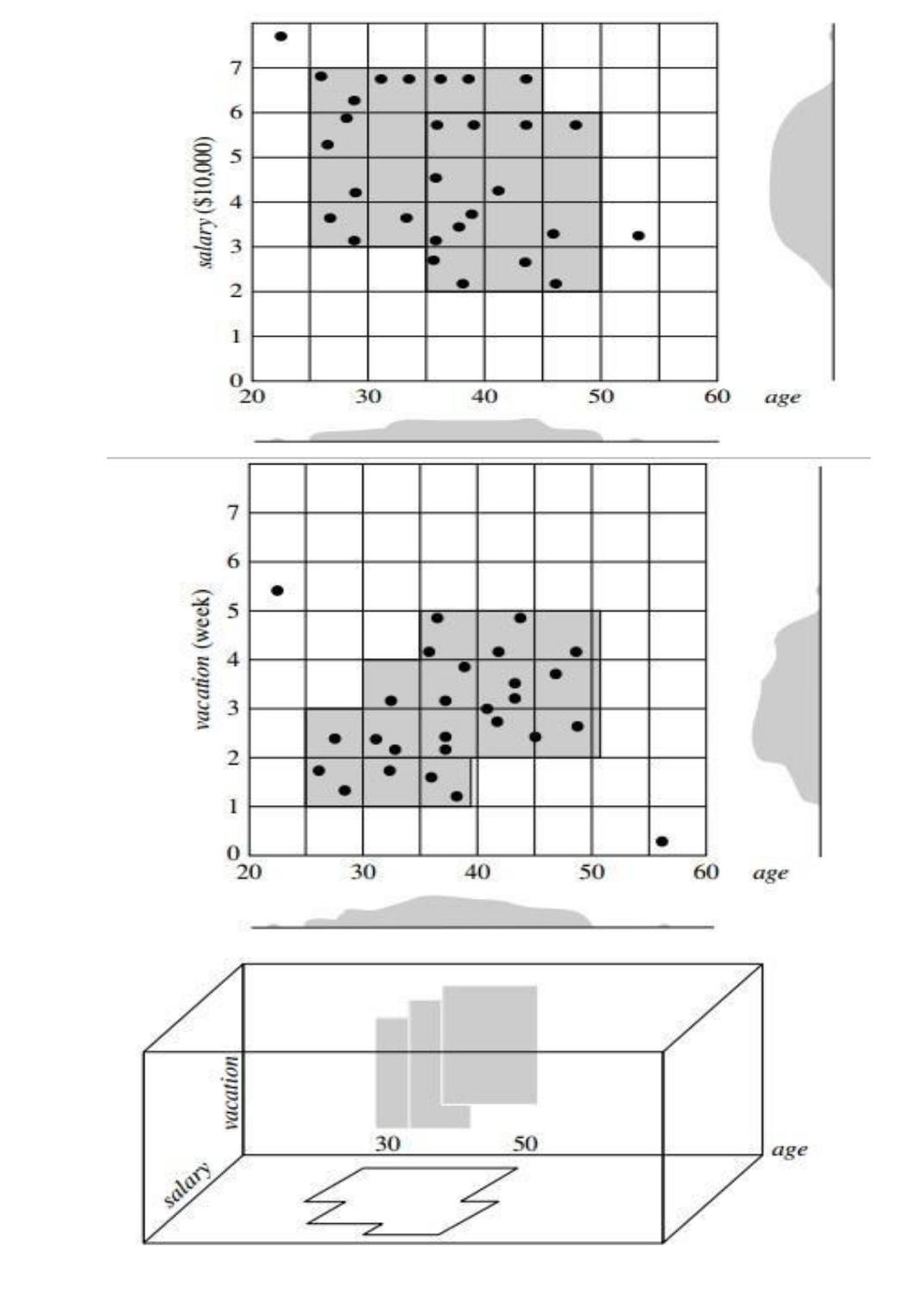
Grid-based methods quantize the object space into a finite number of cells that form a grid structure. All the clustering operations are performed on the grid structure (i.e., on the quantized space).



***15***

The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space. Using grids is often an efficient approach to many spatial data mining problems, including clustering. Therefore, grid-based methods can be integrated with other clustering methods such as density-based methods and hierarchical methods.

[10][11]



|  |  |
| --- | --- |
| Fig: Grid based Clustering Example. | [1 0 ]. |



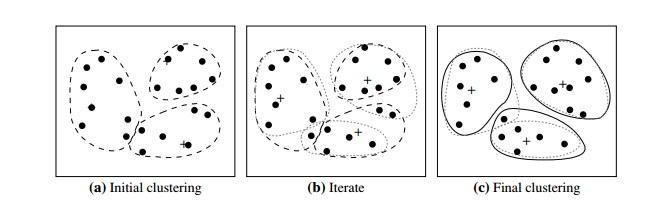
***16***

**1.7.4.** **Partitioning methods:**

Given a set of n objects, a partitioning method constructs k partitions of the data, where each partition represents a cluster and k ≤ n. That is, it divides the data into k groups such that each group must contain at least one object. In other words, partitioning methods conduct one-level partitioning on data sets. The basic partitioning methods typically adopt exclusive cluster separation. That is, each object must belong to exactly one group. [10]

Most partitioning methods are distance-based. Given k, the number of partitions to construct, a partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. The general criterion of a good partitioning is that objects in the same cluster are “close” or related to each other, whereas objects in different clusters are “far apart” or very different. There are various kinds of other criteria for judging the quality of partitions. Traditional partitioning methods can be extended for subspace clustering, rather than searching the full data space. This is useful when there are many attributes and the data are sparse.[10][11]

Achieving global optimality in partitioning-based clustering is often computationally prohibitive, potentially requiring an exhaustive enumeration of all the possible partitions. These heuristic clustering methods work well for



finding spherical-shaped clusters in small- to medium-size databases. [10]



***17***

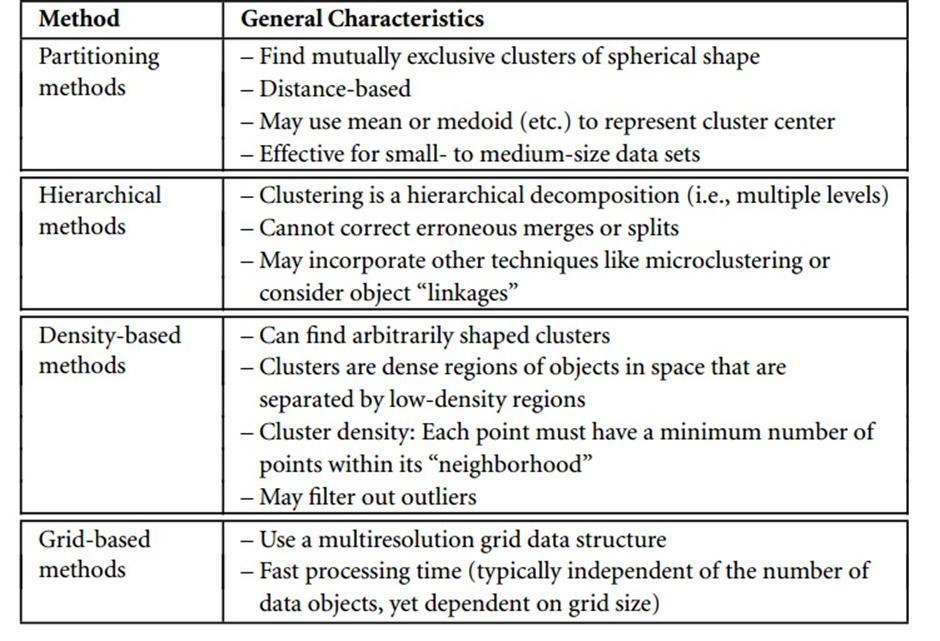


Fig: Clustering using Partition Method (K-means). [10]

Fig: Overview of Clustering Method discussed above. [10]

**1.7.3.1. K-Means: A Centroid Based Technique**

Suppose a given data set, D, contains n objects in Euclidean space. Partitioning methods distribute the objects in D into k clusters C1, C2, C3, .... , Ck that is Ci ⊂

D and Ci ∩ Cj for 1 ≤ i, j ≤ k.

An objective function is used to assess the portioning quality so that similar objects within a cluster are similar to one another but dissimilar to objects on other clusters. The objective function aims for high intracluster similarity and how intercluster similarity.

**1.7.3.2. How does the K-means algorithm work?**

The K-means algorithm defines the centroid of a cluster as the mean value of the points within the cluster. Steps are as follows-



***18***

**Step 1:** It randomly selects the K of the objects in D, each of which initially

represents a cluster mean or center.

**Step 2:** For each of the remaining objects, an object is assigned to the clusterto which it is most similar, based on the Euclidean distance between the object and the cluster mean.

**Step 3:** The K-means algorithm then iteratively improves the within clustervariation. For each cluster, it computes the new mean using the objects assigned to the cluster in the previous iteration. All the objects are then reassigned using the updated means as the new cluster centres.

**Step 4:** The iteration continues till the assignment is stable, that is, that is theclusters formed in the current round are the same as those formed in the previous round. [10]

**Algorithm**

1. Arbitrarily choose k objects to the cluster to which the object is the most similar.
2. Repeat until no change in cluster
   1. Assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster.
   2. Update the cluster means, that is, calculate the mean value of the objects for each cluster.[10]

**K-means Partitioning-**

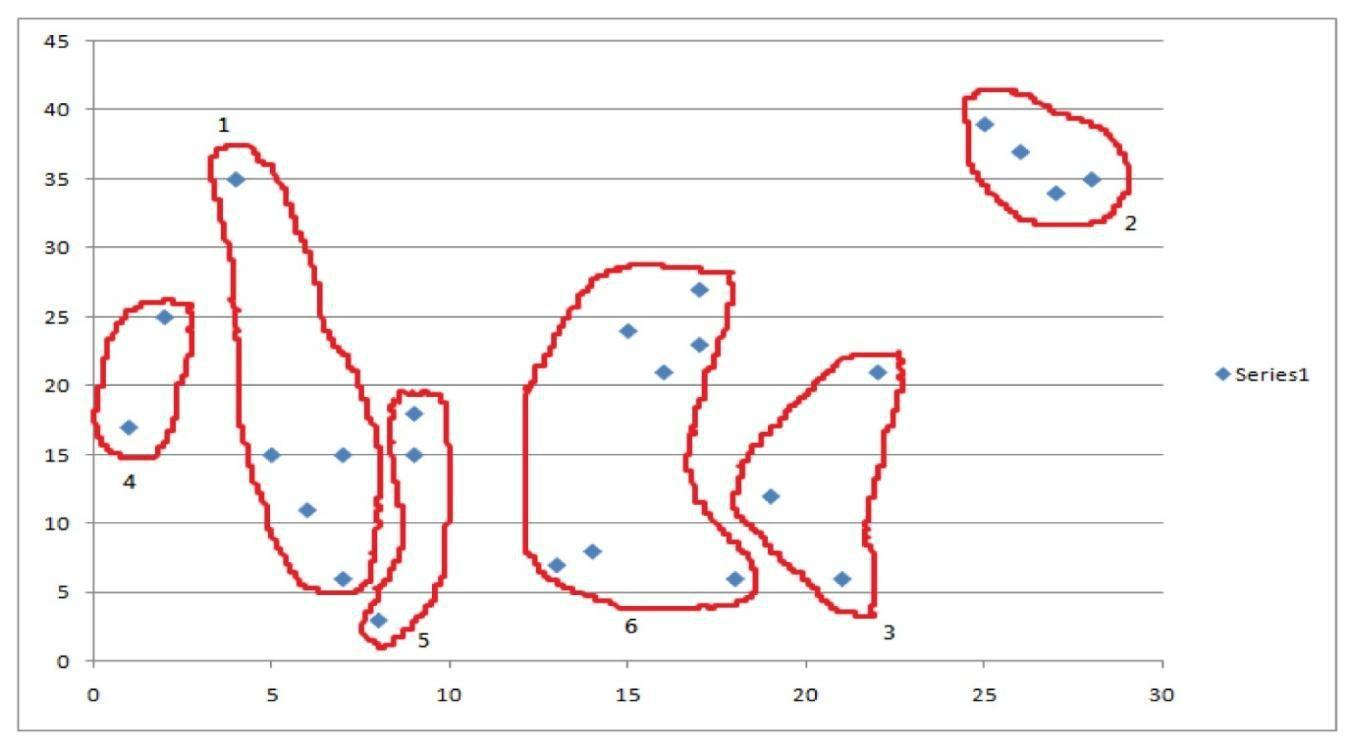
Consider a set of objects located in a 2-D space, as depicted in the following figure-

|  |  |  |  |
| --- | --- | --- | --- |
| Student id | A Score | B Score | Cluster |
|  |  |  |  |
| 1 | 7 | 15 | 1 |
|  |  |  |  |
| 2 | 27 | 34 | 2 |
|  |  |  |  |
| 3 | 18 | 6 | 6 |
|  |  |  |  |
| 4 | 15 | 24 | 6 |
|  |  |  |  |
| 5 | 13 | 7 | 6 |
|  |  |  |  |
| 6 | 17 | 23 | 6 |
|  |  |  |  |
| 7 | 21 | 6 | 3 |
|  |  |  |  |



***19***

|  |  |  |  |
| --- | --- | --- | --- |
| 8 | 5 | 15 | 1 |
|  |  |  |  |
| 9 | 6 | 11 | 1 |
|  |  |  |  |
| 10 | 14 | 8 | 6 |
|  |  |  |  |
| 11 | 8 | 3 | 5 |
|  |  |  |  |
| 12 | 2 | 25 | 4 |
|  |  |  |  |
| 13 | 17 | 27 | 6 |
|  |  |  |  |
| 14 | 7 | 6 | 1 |
|  |  |  |  |
| 15 | 28 | 35 | 2 |
|  |  |  |  |
| 16 | 26 | 37 | 2 |
|  |  |  |  |
| 17 | 9 | 15 | 5 |
|  |  |  |  |
| 18 | 16 | 21 | 6 |
|  |  |  |  |
| 19 | 25 | 39 | 2 |
|  |  |  |  |
| 20 | 19 | 12 | 3 |
|  |  |  |  |
| 21 | 4 | 35 | 1 |
|  |  |  |  |
| 22 | 22 | 21 | 3 |
|  |  |  |  |
| 23 | 9 | 18 | 5 |
|  |  |  |  |
| 24 | 1 | 17 | 4 |
|  |  |  |  |



According to the algorithm we arbitrarily choose 6 objects as the initial cluster centres, which are marked as “+”. Each object is assigned to the cluster based on the cluster centre to which it is the nearest.



***20***

Next, the cluster centres are updated. That is, the mean value of each cluster is recalculated based on the current objects in the cluster. Using the new cluster centres, the objects are redistributed to the cluster based on which cluster centre is the nearest.

This process continues leading to figure given above. The process of iteratively reassigning objects to clusters to improve the partitioning is referred to as “iterative relocation”. Eventually, no reassignment of the objects in any cluster occurs and so the process terminates. The resulting clusters are returned by the clustering process.

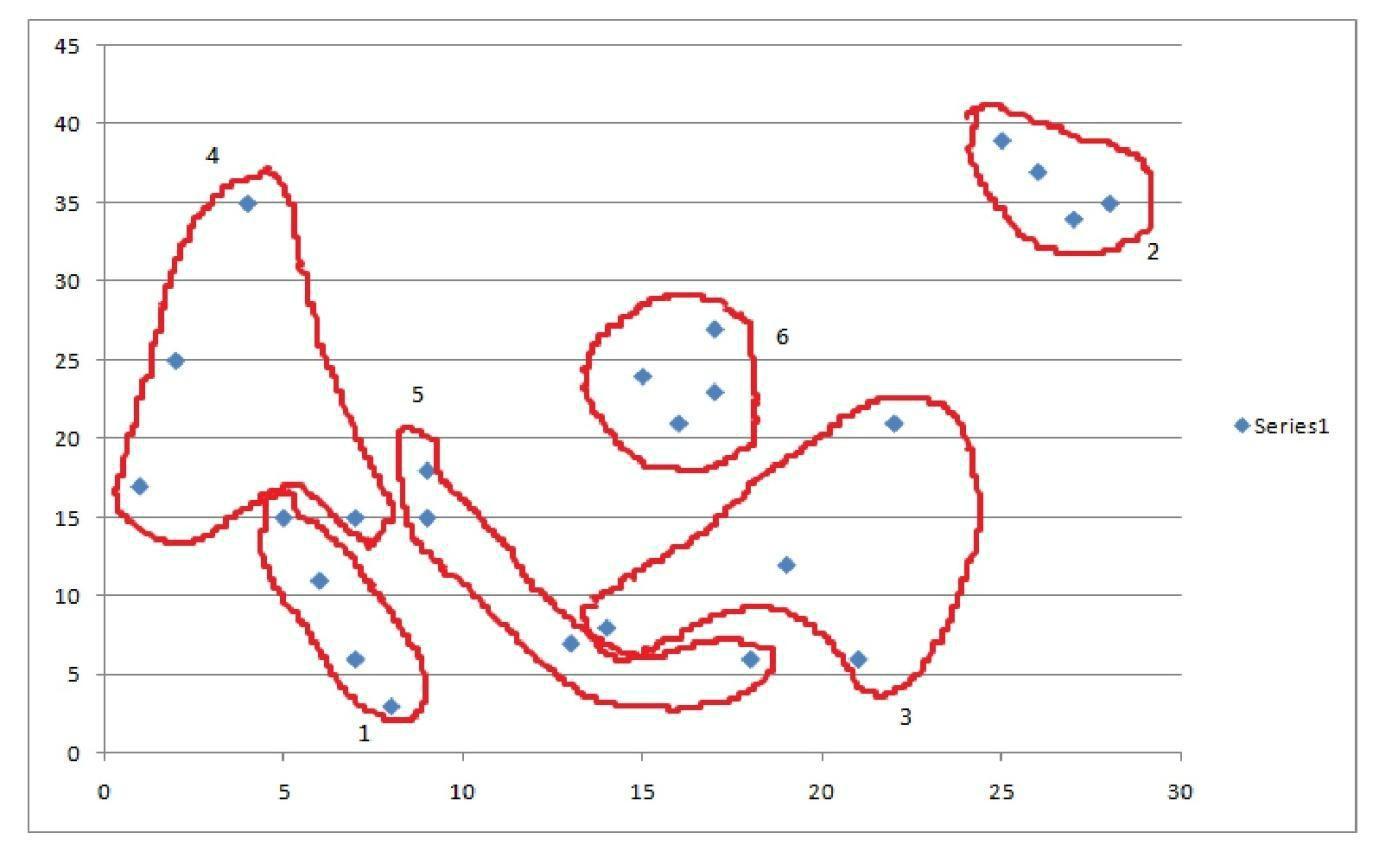


FIG: The Clustered groups

**Working Details**

We are assuming n number of homogeneous clusters that have been assigned so far and each cluster has at least one object assigned to it. The following steps are taken to reassign these objects into groups of each number of objects.

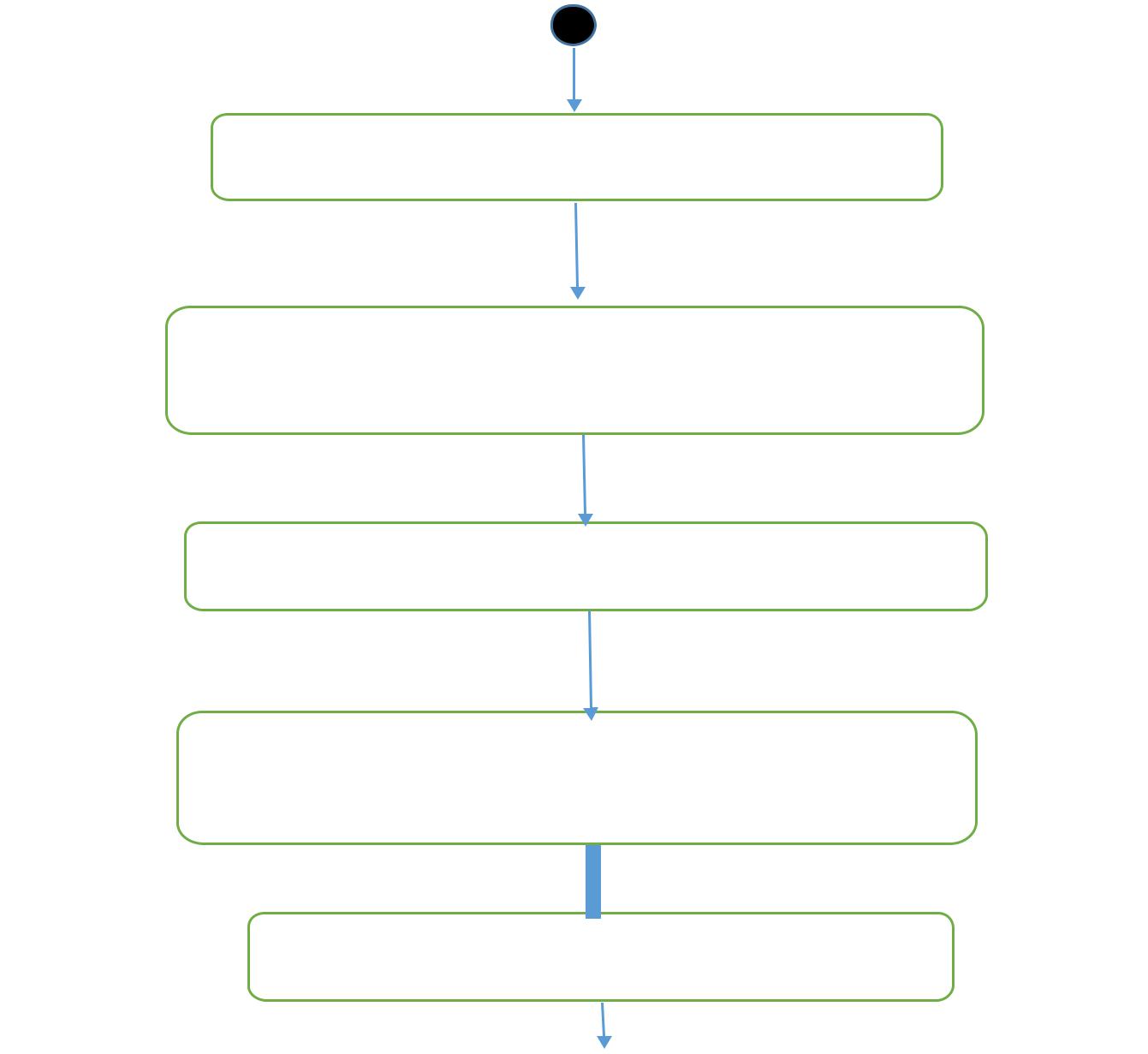
We are assuming n number of homogeneous clusters that have been assigned so far and each cluster has at least one object assigned to it. The following steps are taken to reassign these objects into groups of each number of objects.



***21***

1. Compare clusters Ci and Ci+1 and evaluate
   1. If the number of objects in Ci < Ci+1, do
      1. Find the object in Ci+1 having the maximum distance and transfer it to Ci
      2. Continue this process until Ci has the required number of objects.
   2. If the number of objects in Ci+1 < Ci, do
      1. Find the object in Ci having the maximum distance and transfer it to Ci+1
      2. Continue this process until Ci+1 has the required number of objects.
2. Repeat the above steps until all the cluster have equal number of objects or there is no more clusters to adjust.

**1.7.3.3. Activity Diagram:**



Problem: Group formation in Collaborative

Learning.

A Class consisting of a group of students has academically meritorious students as well as

students who are not so.

Firstly students of alike kinds clubbed together in small homogeneous groups.

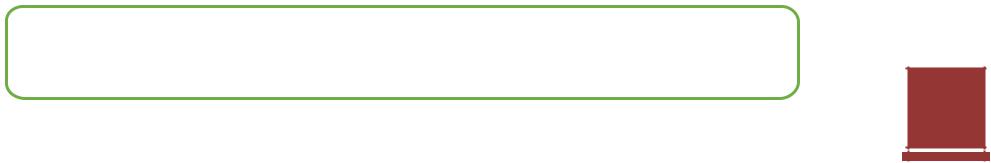
Thus in order to make groups with all kinds of students with various attributes, further

rearrangement is done.

N numbers of homogeneous clusters with

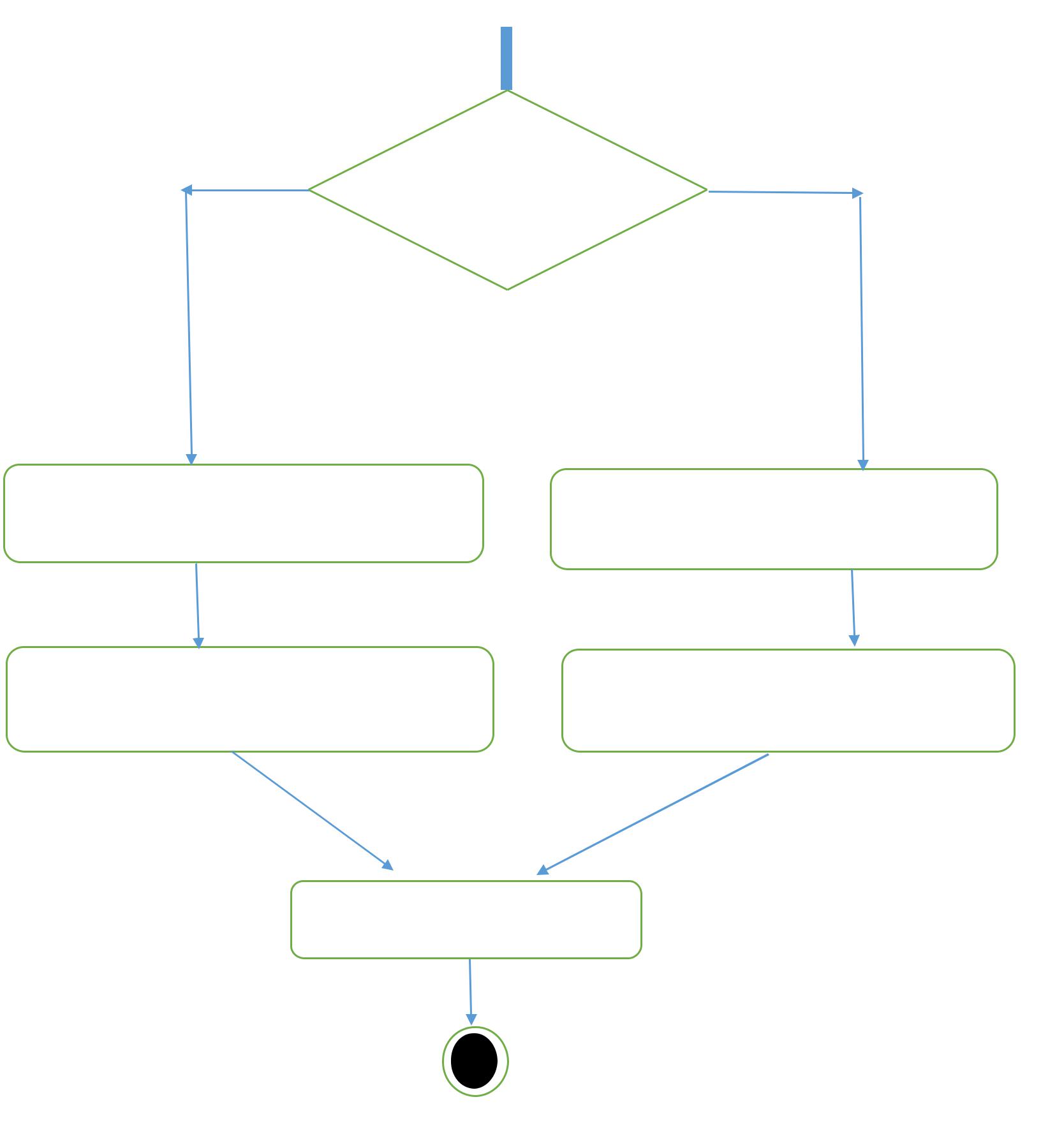


at least 1 object in them is formed.



Consider Cluster Ci and Ci+1. 0<=i<=N

***22***



|  |  |  |  |
| --- | --- | --- | --- |
| yes | If no. of | No |  |
|  |  |
|  | objects in C | i < |  |
|  | C i+1 |  |  |

Find the objects having highest

distance in C i+1 and transfer in C i.

Continue this process until the C i equal to desired no. of objects.

|  |  |  |
| --- | --- | --- |
| Find the objects h | aving highest |  |
| distance in C i and transfer in C | | i+1 . |

|  |  |
| --- | --- |
| Continue this process until the C | i+1 |

equal to desired no. of objects.

Repeat this until all

clusters are processed.



***23***

**CHAPTER 2**

**SURVEY OF TECHNOLOGIES**

Dewiyanti, Brand-Gruwel, S.Jochems & Broers (2007) carried out a project on *Computers in Human Behavior* which emphasized on collaborative learning in asynchronous computer-supported collaborative learning environments. In the project a study was carried out to gain response from distance students on their experiences with collaborative learning in synchronous computer supported collaborative learning (CSCL) environments. The study was undertaken among students of Open University of the Netherlands who were working in groups of four to eleven persons. The finding revealed positive experiences and students being satisfied with collaborative learning. The findings suggested group cohesion influences students’ satisfaction with collaborative learning.

Crystal Daugherty in her paper on *The Benefits of Collaborative* *Learning in the Elementary Classroom* under St. CatherineUniversity designed to see the possible benefits of collaborative learning in an elementary classroom. This investigation was conducted at a charter school with two fourth grade classes. Data collection methods included: a pre and post student attitude survey, individual student and peer evaluation, work completion tracking, and behavior tracking. The results showed a positive effect on student learning. Student attitude reflected in the surveys showed improvement. This also was seen in the amount and quality of work completed throughout the study. Off task behaviors were also lowered. The study suggested to continue with group roles in collaborative learning and educating students on their importance.



***24***

Michael J. Sipusic, Robert L. Pannoni, Randall B. Smith, John Dutra, James F. Gibbons, and William R. Sutherland in their paper on Virtual Collaborative Learning defined Tutored Video Instruction (TVI) which is a collaborative learning methodology in which a small group of students studies a videotape of a lecture. A fully virtual version of TVI called Distributed Tutored Video Instruction (DTVI) were considered, in which each student has a networked computer with audio microphone-headset and video camera to support communication within the group. In the report, he compared survey questionnaires, observations of student interactions, and grade outcomes for students. Course grade outcomes for TVI and DTVI were indistinguishable, and these collaborative conditions proved better than lecture. The idea of the project is to use K-Means algorithm as a process of clustering groups. K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results is a partitioning of the data space into groups.

The advantages of using K- Means are

1. If variables are huge, then K-Means most of the times performs computationally faster than hierarchical clustering for small values of K.
2. K-Means produce tighter clusters than hierarchical clustering

These are reasons why we use K-Means over hierarchical methods, density methods, grid based methods as our clustering method. Although there are disadvantages of K-Means like



***25***

difficulty to predict the K-Value and it does not work well with clusters of different sizes and different density. These we can overcome by having initial boundary constraints on the K-values or using optimization algorithms to minimize the cross-validation error. Methods like DBSCAN, OPTICS which are density based methods cannot cluster data sets well with large differences in densities. Methods such as BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) which is Hierarchical Clustering method suffers from high time complexity and needs to store all data in memory. Hierarchical Methods have a time complexity of O(n2 log n).

CLUSTERING METHODS

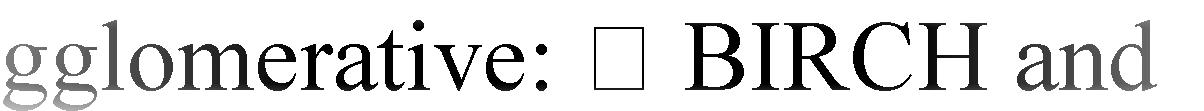
Partitioning clustering

K- Means

K-Medoids : 

and 

Hierarchical clustering

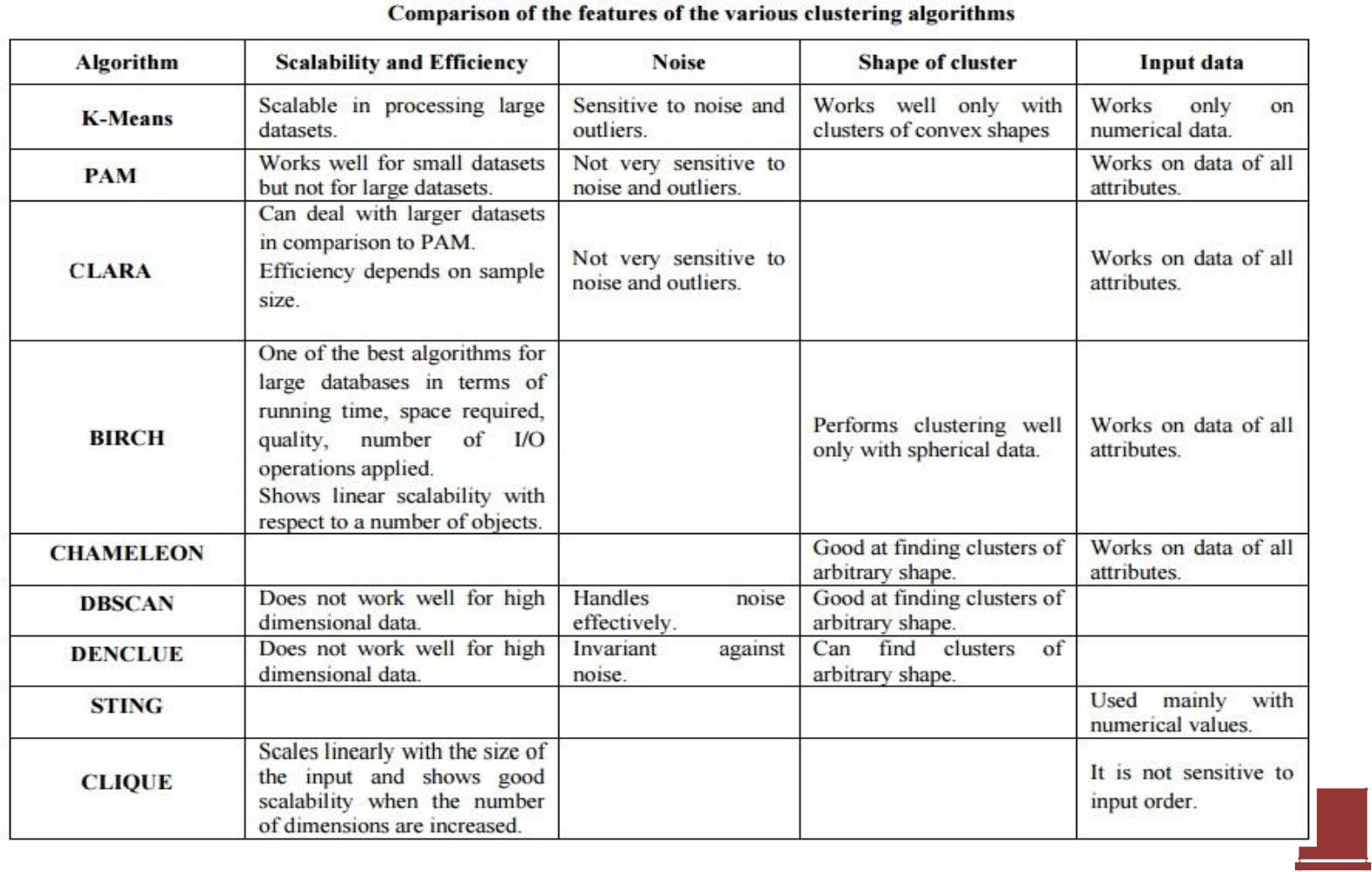


A

Divisive

Density-based clustering: DBSCAN and DENCLUE

Grid-based clustering: STING



CLIQUE

***26***

**CHAPTER 3**

**3.1. REQUIREMENTS AND ANALYSIS**

**3.1.1. Problem Definition:**

Formation of groups/teams is an important aspect of implementing Collaborative Learning among students. The followings are recommended in Group formation - Formation of teams of 3-4 students for most tasks. Making the team Heterogeneous in its ability level. The unfairness of forming a group with only weak students is obvious, but groups with only strong students are equally undesirable. In heterogeneous groups, the weaker students gain from seeing how the better students approach the problems and the stronger students gain a deeper understanding of the subject by teaching it to the others.

Clustering allows to use multiple attributes to identify similar groups in an unsupervised fashion. The clustering is the process of partitioning set of data objects into subsets. Each subset is a cluster such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters.

Thus the process of clustering groups can be identified; K-Means algorithm is one of the Basic Clustering Methods (Partitioning Method) which uses 'Iterative Relocation Technique' to do so. Here this has been used to get the desired output till now.

The clustering process in the project work has been divided into sub problems. Heterogeneous groups are created initially. Then the work of developing an application type frontend had been undertaken where the user can easily input the student details which gets stored into the database. Post the JDBC connectivity, the final phase has been to extract the data from database and apply suitable clustering method on it, thereby achieving proper equi spaced heterogeneous groups of data.



***27***

**3.2. REQUIREMENT SPECIFICATIONS:**

The data so collected at the beginning was of a group of students and the corresponding values against them which were used to implement the heterogeneous group formation. A number of tables was created in which different sets were kept as per to their A and B values (mean calculation).

The implementation of this Group formation problem is being done in the Java Platform at the back end and the logic is implemented using Linked List data structure. In the program the clustering algorithms implemented using the Cluster Analysis technique and Partitioning Algorithm (K means). To do this we needed the Oracle Java JRE and JVM, available in the Oracle website. The development was undertaken by maintaining the OOP concept from the beginning and thus Java turned out quite handy on implementing so.

In this project the requirement of the hardware components are not rigorous, but can be implemented with bare minimum computers and laptops. The hardware components should be sufficient enough to support the software (i.e.: Java (SDK 1.6 or above).

The major coding portion for the implementation of this project requires the help of the Java programming language (SDK 1.6 or above) and MS Access (MS Office 2007 or above).

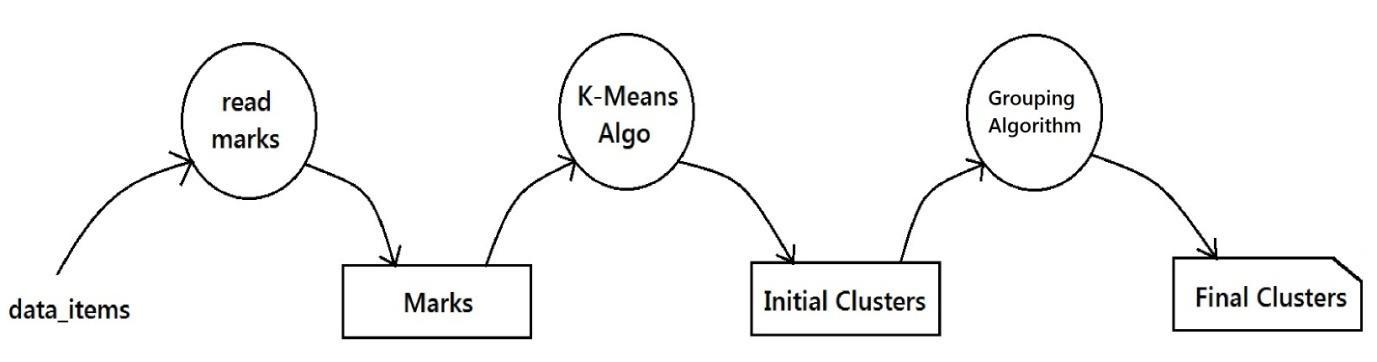


***28***

**3.3. PLANNING AND SCHEDULING:**

The planning process started after studying about Collaborative and Cooperative Learning texts and the problem was selected in the due course. Then the Brief description, Problem Statements were formed and development was being done at the same time under the guidance of the mentor. The front end design and the further developments in this projects is expected to be done in the near future.

**3.3.1. PLAN OF THE PROJECT:**



**3.4. SOFTWARE AND HARDWARE REQUIREMENTS:**

The Software and the Hardware Requirements are quite minimal in this case as only Java programming software was needed available as a freeware over the internet, that is only Java Development Kit and Java Virtual Machine were needed. Apart from that we needed ORACLE 10g or 11g. JDBC connectivity was required to move data from java to database and vice versa. Apart from these for the last part we needed RStudio for the clustering the student data.

*WHAT IS R?*

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language



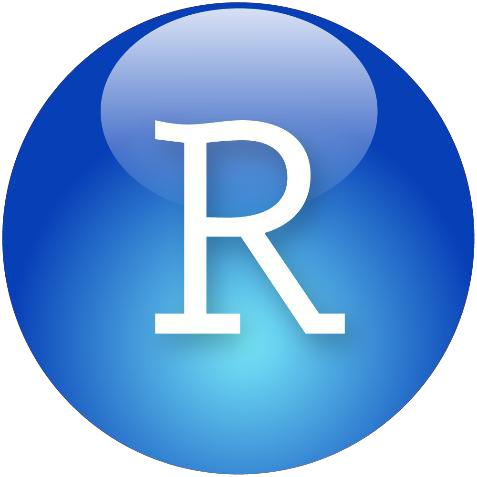
***29***

and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues’ provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, …) and graphical techniques, and is highly extensible. One of R’s strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. The user retails the full control here.

**3.4.1. THE R ENVIRONMENT**

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

* An effective data handling and storage facility.
* A suite of operators for calculations on arrays, in particular matrices.
* A large, coherent, integrated collection of intermediate tools for data analysis.
* Graphical facilities for data analysis and display either on-screen or on hardcopy.
* A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.



The term “environment” is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

Many users think of R as a statistics system. We prefer to think of it of an environment within



***30***

which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy.

For Hardware requirements only a set of Desktops and Laptops with descent configuration are ample for implementing this project.

**3.5. Preliminary Product Description:**

The Software so far developed can generate the small and different heterogeneous groups which stand as the basis of this project and that is done with the help of Java Programming language implemented by using Linked List Data structures. Different table formation is also implement for forming the groups.



***31***

**3.6. GNATT CHART**



Project Specification



Project Requirements Analysis



Project Planning



Project Scheduling



Project Design



Coding Module 1



Testing Module 1



Coding Module 2



Coding Module 3



Testing Module 2



Testing Module 3



Coding Module 4



Testing Module 4



System Testing



Validation



Implementation



Documentation

Aug

Sep

Oct

Nov

Dec

Jan

Feb

Mar

Apr

‘15

‘15

‘15

‘15

‘15

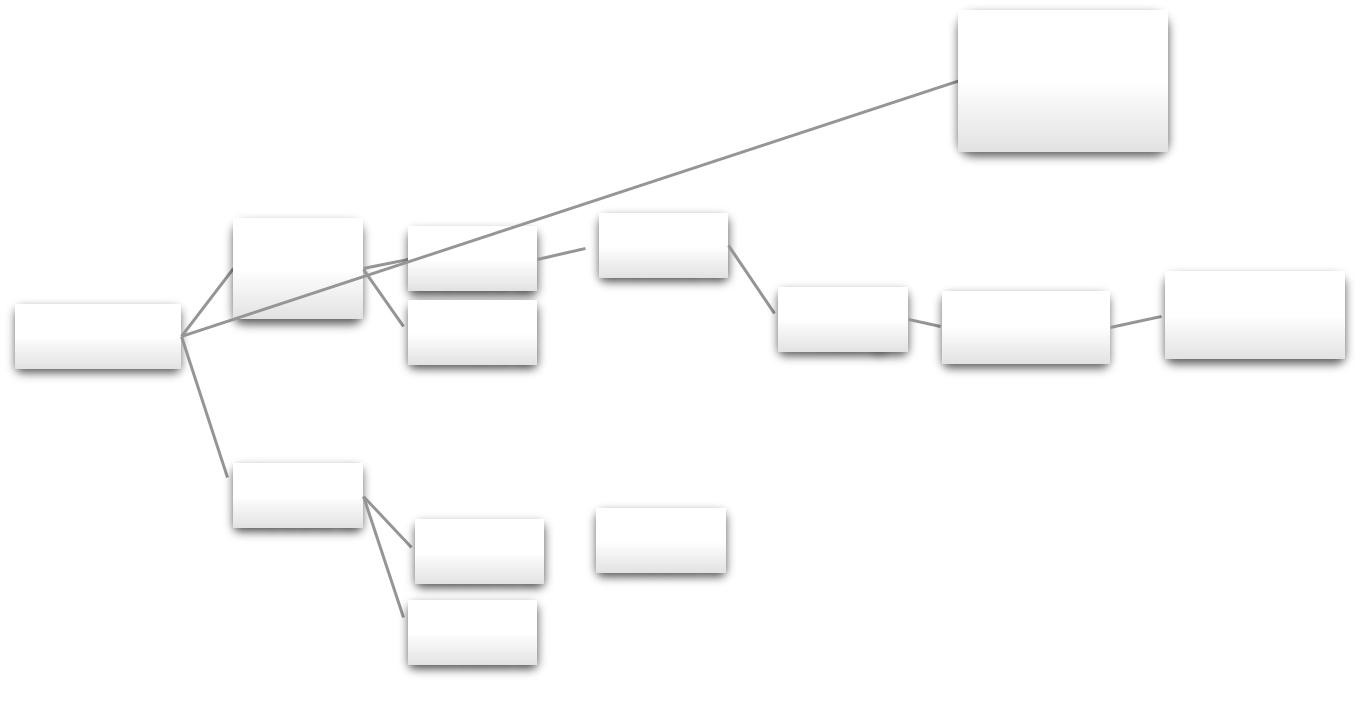
‘16

‘16

‘16

‘16

**3.7.PERTCHART**:



Documentation

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Coding | Coding | Coding |  |  |  |
|  |  |  | Module 4 |  |  |  |
|  |  |  | Module 2 |  |  |  |
|  |  |  | Module 1 |  |  |  |
|  |  |  | System |  | Implementati |  |
|  |  |  |  |  |  |
|  |  |  |  | Coding | Validation |  |
| Planning |  | Scheduling |  | Testing | on |  |
|  |  | Module 3 |  |
|  |  |  |  |  |  |  |  |



Testing

Module 1 Testing Testing

Module 2  Module 4

Testing

Module 3

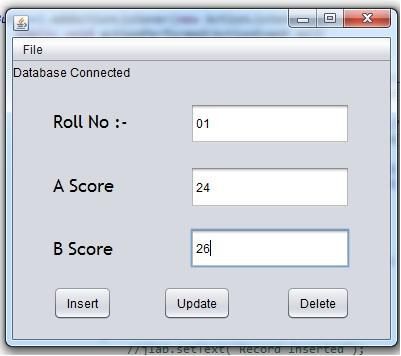


***32***

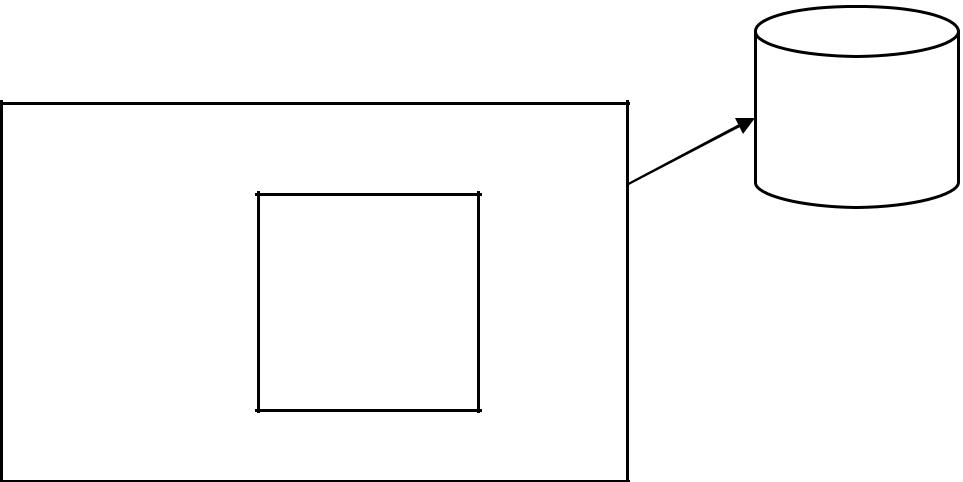
**3.8. Conceptual Model:**

The Problem so far solved gives us an idea about the implementation phase starting from the collection of data at the beginning and then channelizing them step by steps giving rise to the required groups at the end.

**3.8.1. MODULE 1:- (Data Acceptance)**



*INITIAL DATA ACCEPTANCE*



JAVA

Front End 1

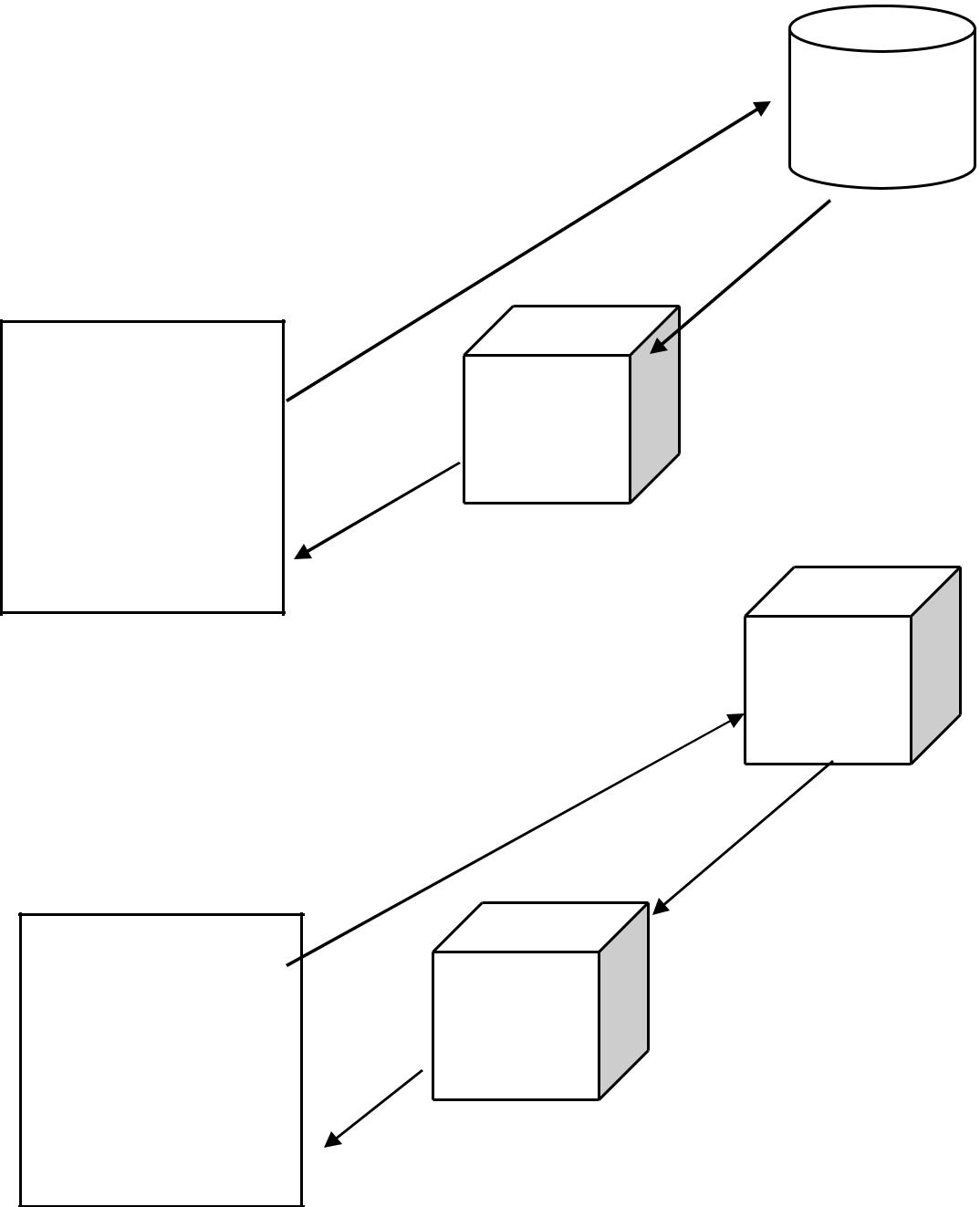
DATABASE



***33***

**3.8.2. MODULE 2:- (Clustering)**

The phase 2 or sub problem 2 mainly deals with the clustering purpose. The RStudio requests for data from the database which is then retrieved into the R Engine and displayed in the RStudio. R functions and commands are run to form unequalized clusters.



Request DATABASE

Retrieve

RSTUDIO

R ENGINE

**3.8.3. MODULE 3:- (Equalization)**

Request R ENGINE

Retrieve

FRONT END 2

Displaying Equalized

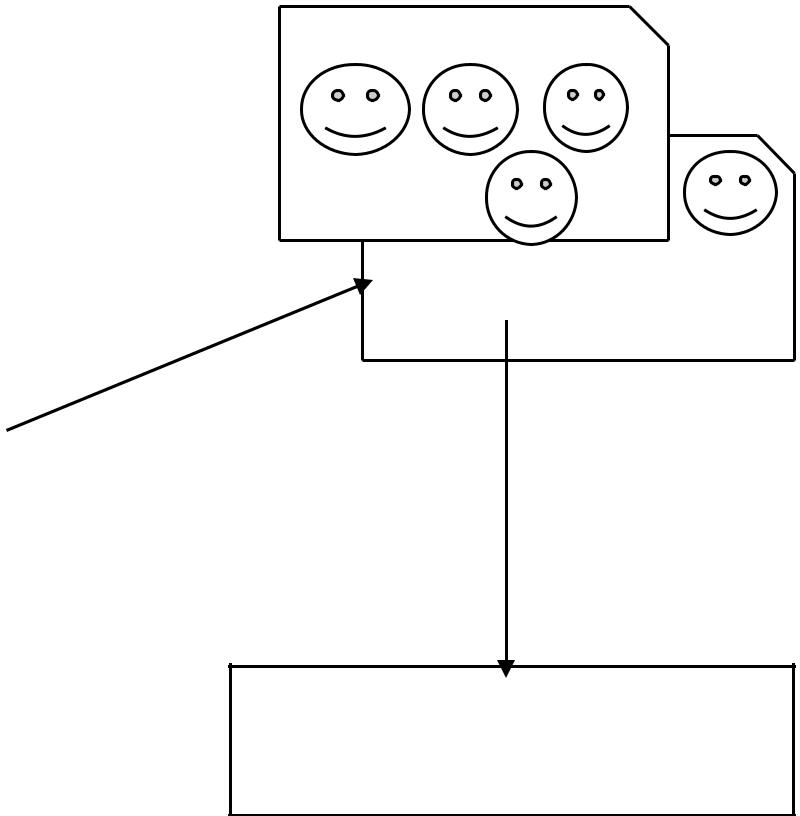
Clusters Fig: Clustering Process

In this phase a second front end is designed which internally retrieves the data from the R Engine which was in turn retrieved and stored from the database. A clustering process or a method is applied on this set of data and the resulting equalized clusters are displayed in the front end.

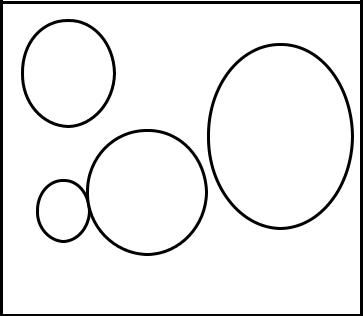


***34***

**3.8.4. MODULE 4:- (Implementation)**



Multiple Heterogeneous Groups



CLUSTERED GROUP

Implementation

Effectiveness of the System

The clustered groups are implemented in real life scenario and surveys are conducted to measure the effectiveness of the system.



***35***

**CHAPTER 4**

**SYSTEM DESIGN:**

The project concentrates on the fact that how the Groups can be formed among students in a class containing pupils of different academic capabilities. A student can be good in learning new things at the first time when he/ she is made to understand certain concepts. On the other hand another student can have difficulties in learning new concepts as easily as it was in the case of the previous student. In the similar fashion one pupil can be talented in extracurricular activities whereas another is not, the same is reflected in case of the academics. Thus to ensure an overall development and growth of all the students present in the class it is necessary to look into their progress in the academic as well as in other co-academic fields leading to overall personality development.

The differences of such capabilities in students can be observed in a class consisting of normal pupils itself. This project has aimed to find out a solution to such a problem, on doing that small groups among students are planned to be formed. The group formation and there after the Equalization ensure that no similar kinds of student are placed in a group, rather they are formed keeping in mind the principle of heterogeneity. Then the students seated in such groups have abilities unlike to each other, such that one can be a quick learner academically whereas another can be a quick learner in other extra/co-curricular activities or sports. Thus they can help each other where they find their group-



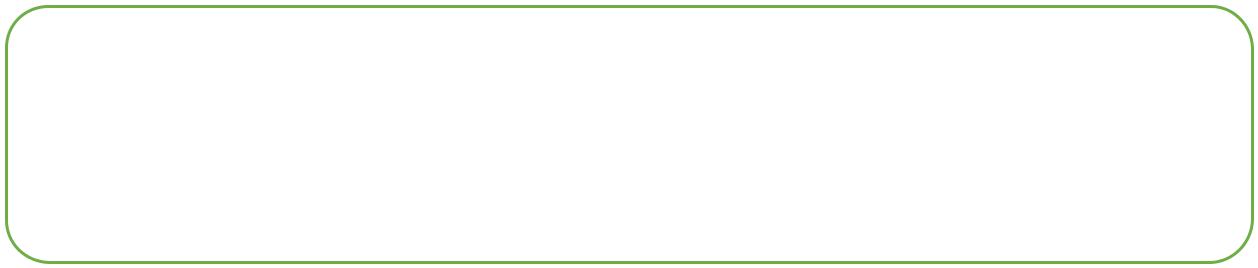
***36***

mate lack in efforts and can be helped. Therefore giving them a scope for overall development and in their upbringing and personality building.

This project has divided the whole process into three subsections:

1. The collection of the initial data and storing them in the data base.
2. Retrieving data from the database and applying K-means algorithm in forming clusters.
3. The process of Equalization.
4. Collection of the final data set to check the effective ness of the result.

h



**MODEULE 01**: Initial Data Collection and storing In DB



**MODULE 02**: Retrieving Data from DB and applying K- Means

algorithm to form Clusters.



**MODULE 03**: The Equalization process to have equal number of

students in each clusters.



**MODULE 04**: The final phase to implement and check the effective

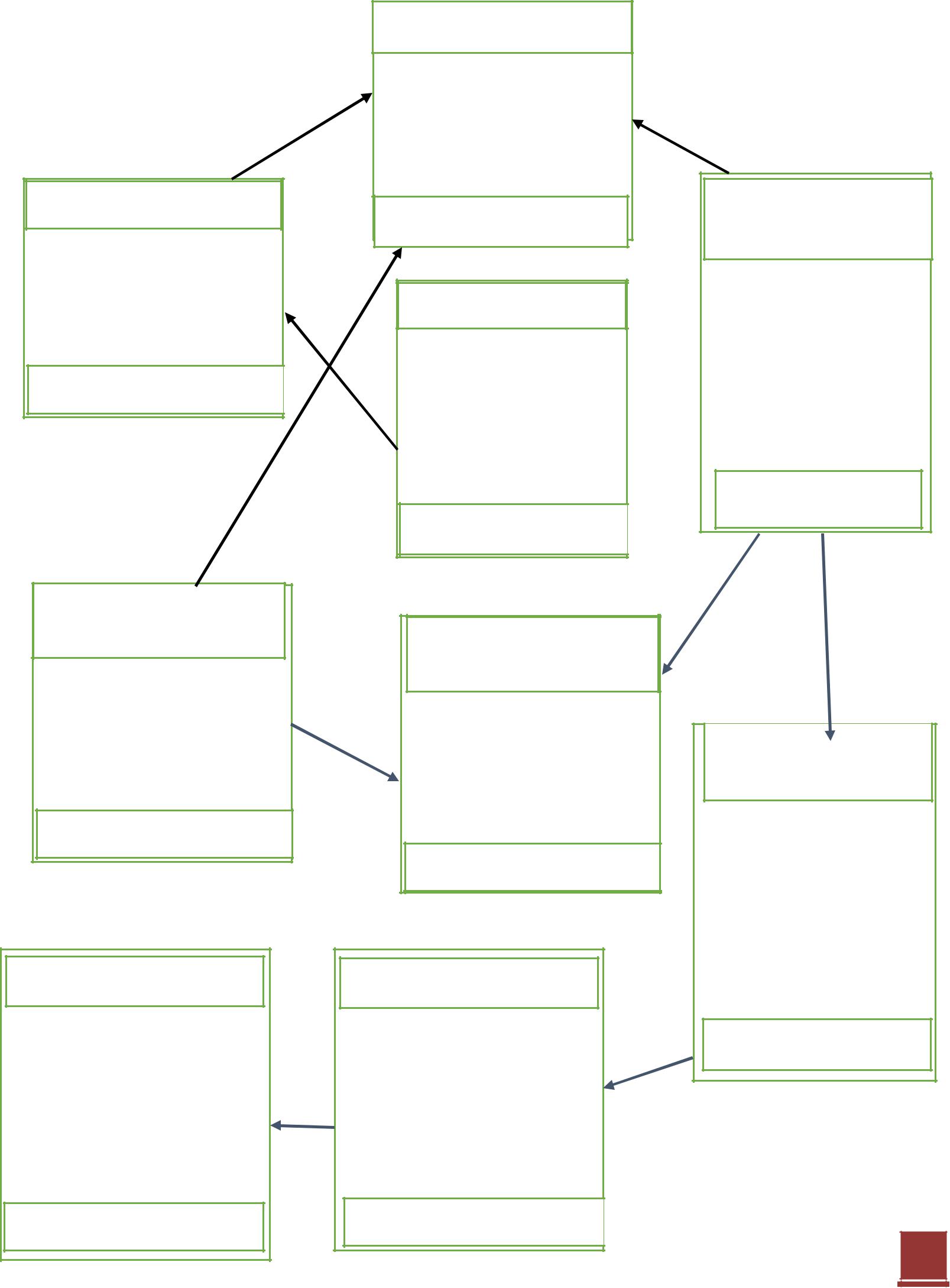
ness of the whole method.

Chart: Module division in the Project.



***37***

**4.1. SCHEMA DIAGRAM:**



**Oracle Data Base**

**Java Code**

* Storing data in temporary variable
* Connection with

DB

**MODULE 01**

* Table Student in

DB

* Storing Data in DB

**MODULE 01**

**Initial Data**

* Front End Tool l
* Accepting the Data from the User/ Programmer

**MODULE 01**

**Equalizations in**

**Java**

* Front End Tool 2
* Cluster information from Module 2, for Equalization

**MODULE 03**

1. **Programming Language**

* Establishment of connection with the

DB

* Retrieving data from DB

**MODULE 02**

**Implementation**

* Implementation of the concept in real in class.

 Checking for the effectiveness of the System

**MODULE 4**

1. **Programming Language**

* Applying K Means Algorithm for Clustering
* Plotting Data in graph

**MODULE 02**

**Implementation**

* Arrangement of groups among students in class.
* Letting them Learn of their own.

**MODULE 4**

**Equalizations in**

**Java**

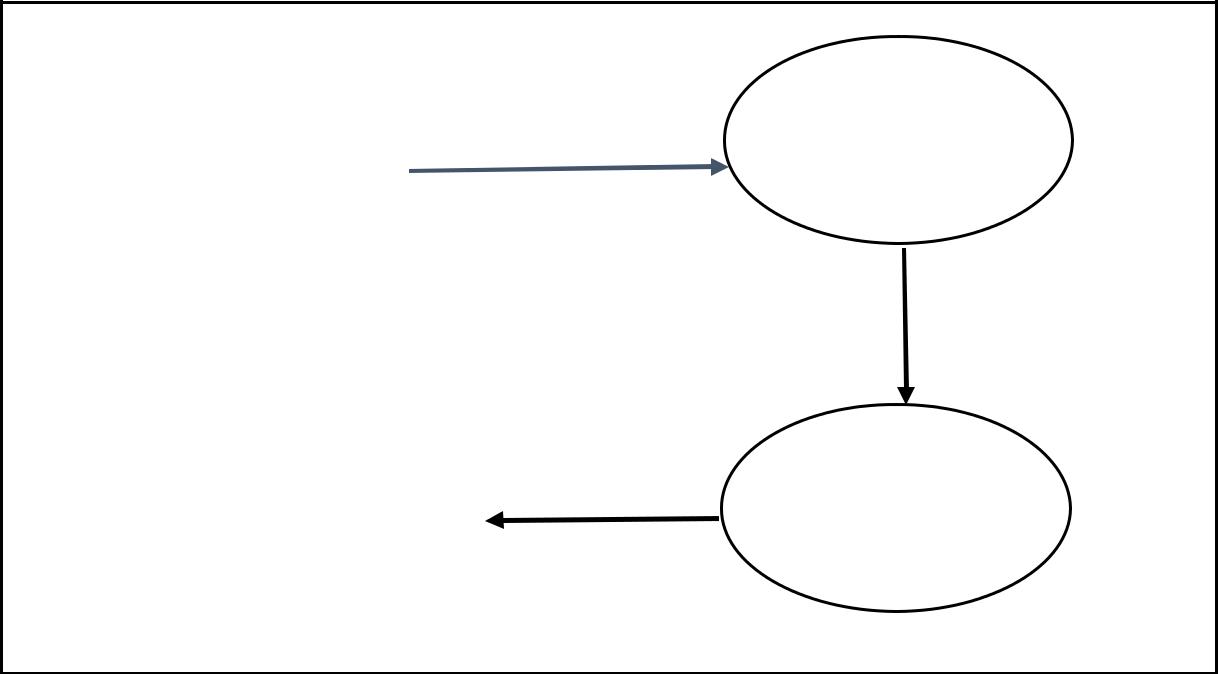
* Running the Equalization Algorithm
* New Clusters with equal number of subjects

**MODULE 03**

***38***

**4.2.MODULE 1**

The Module 1 or the first step of this project concentrates on the fact that a preliminary data is collected from the class containing students on basis of certain performances reflected upon the marks they scored. On basis of the marks they are feed to the system. The Front End 1 easily allows user to enter the values of the two subjects used for group formation. The data so entered is stored in a Data Base table at the Back end, from where the data is referred from time again.



|  |  |  |  |
| --- | --- | --- | --- |
|  | Prepares for the test | Take Initial |  |
| Students in Class | Test |  |
|  |  |  |  |

Marks obtained

|  |  |  |
| --- | --- | --- |
| Data Stored in DB | Access to DB | Enter through |
|  |  | the Front End |
|  |  | 1 |

Fig: DFD for the Module 01

The DFD clearly shows the various process of the phase that has to undergo in this module. The following is the Flowchart for the phase 1: shows the process starts and the students are asked to take a test which generally is designed to record their performance in different subjects. Here for simplicity 2 subjects are only considered which are recorded and are entered into the system through the front end and is ultimately stored in the Data base. The Data base table was



***39***

already created with ROLL\_NO of the students as the Primary key and

corresponding to that there are other 2 fields namely the A\_SCORE and the

B\_SCORE.

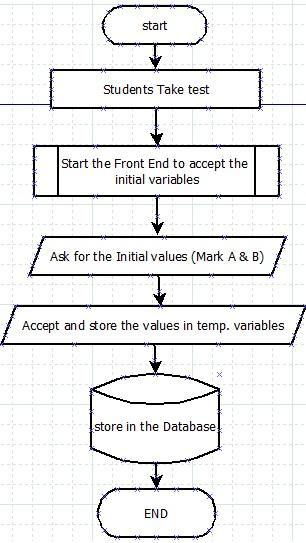


Fig: Flow chart for Module 1

Algorithm:

1. Start
2. Students take test and the marks of the two subject are taken as inputs via the Front End 1.
3. Values are stored temporary in local variable and Data Structure.
4. The values are then written on to the Data Base at the back end.
5. Successful Database Update, Insert, Delete can be done in Table.
6. Stop.



***40***

**4.3.MODULE 02:**

In the first module it was observed that the data was collected via a test and was stored in the Data Base. Now in this module we will observe how the data is retrieved from the data base and is utilized in R programming language. The R is a programming language and software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. Polls, surveys of data miners, and studies of scholarly literature databases show that R's popularity has increased substantially in recent years. R and its libraries implement a wide variety of statistical and graphical techniques, including linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, and others. R is easily extensible through functions and extensions, and the R community is noted for its active contributions in terms of packages. Many of R's standard functions are written in R itself, which makes it easy for users to follow the algorithmic choices made.

Once the connection with the Data base is created the table from the database is retrieved and from the retrieved table the data is prepared for the analysis of the clustering for the further use. In this stage it has to be kept in mind that clustering is the process by which data with similar characteristics are grouped together, or data with certain similarities in their attributes are clubbed to form groups with similar nature. Thus the same concept is implemented in this case the data so collected in form of the marks of two subjects after the test was taken by the students are placed in fields of two columns namely A\_SCORE and B\_SCORE, the two field are utilized for the clustering processing. The two fields are treated



***41***

as the two coordinated of the graph on which the marks are plotted. The plotted marks are then grouped together. This grouping is termed as clustering. There are number of clustering algorithms are available and they are done so with different characteristics of the data. Some are hierarchical, partitioning, density based etc etc.

Here we have used the K Means algorithm which is a Partitioning algorithm, where a random Cluster Centre is generated for the measuring the mean distance (Euclidean Distance) between the cluster subject and the cluster center. The closest the subject is to a Cluster Centre, the higher the probability for it to get included into that cluster. Thus by this way the K Means clustering algorithm works. And this similar principle is also applied in the R programming language which has all this pre-defined as a function in its libraries.

The following DFD diagram clearly shows hoe this clustering is performed and more over the overall task performed by this Module in this section. The first thing is done is to ensure connection with the database and the R programming language and after that the processing is performed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Connection | Database | Access |  |
| Connecting | |  |  |  |
| with DB | | Existing Table |  |  |
|  |  |  |  |
|  | Output |  |  |  |
| Clusters (Un |  | K Means | Retrieve |  |
| equalized) |  | KM Algo | Table for |  |
|  |  |  | Processing |  |
|  |  |  |  |  |

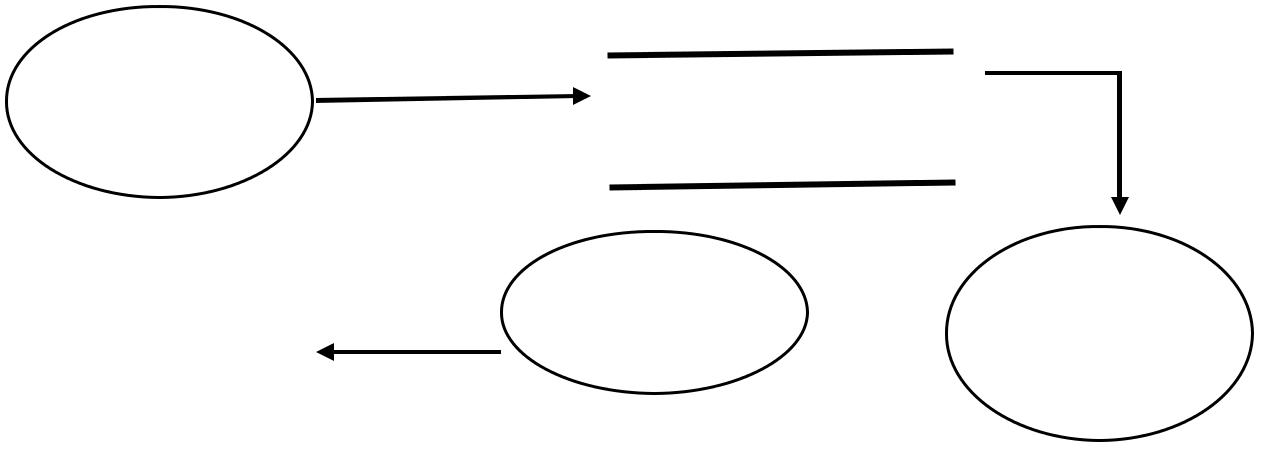


Fig: DFD of the Module 02



***42***

Flowchart:

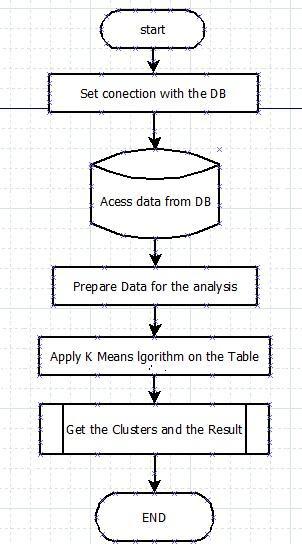


Fig: The Flowchart of the Module 2

Algorithm:

1. Start
2. Set connection with the Database and the R programming.
3. Access the data table so created and values entered in module 1.
4. Retrieve data from the Table and prepare the data for the next phase that is the analysis.



***43***

1. Apply the following K Means Algorithm on the data and get the un equalized clusters:
   1. Inputs:
      1. K: the number of clusters.
      2. D: a data set containing n objects.
   2. Outputs:
      1. A set of K clusters.
2. Methods:
   1. Arbitrarily choose K objects from D as the initial cluster centers;
   2. Repeat
      1. (re) assign each object to the cluster to which the objects is the most similar, based on the mean value of the objects in the cluster.
      2. Update the cluster means, that is calculate the mean value of the objects for the each cluster.
      3. Until no change.
   3. End.
3. Stop.

According to the algorithm, we arbitrarily choose objects as the initial cluster centers, where clusters are marked. Each object is assigned to a cluster based on the cluster center to which it is nearest. Such a is formed and ultimately give rise to the final cluster structures. Next, the clusters centers are updated. That is, the mean value of each cluster is recalculated based on the current objects in the cluster. Using the new cluster centers, the objects are redistributed to the clusters based on which the cluster centers are nearer. This process iterates, leading to



***44***

iteratively reassigning objects to clusters to improve the partitioning which is also referred to as Iterative Relocation. Eventually no reassignment of the objects to any cluster occurs and so the process terminates. The resulting clusters are returned to the clustering process.

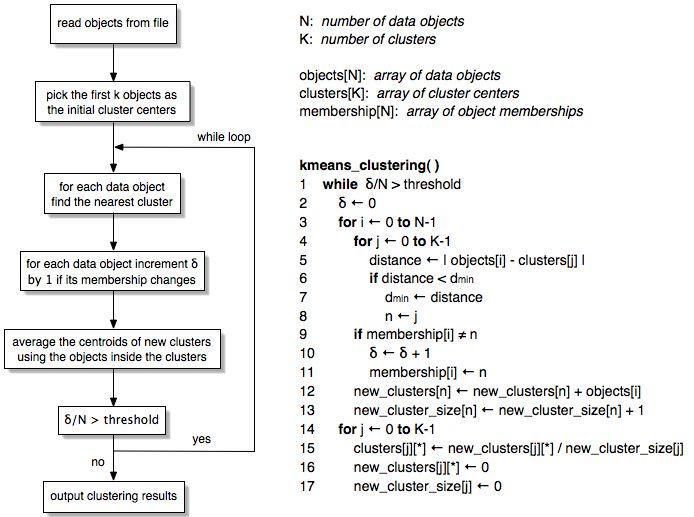


Fig: A formal flowchart and algorithm of K means.

**4.4.MODULE 03:**

The third module is the one which deals with the equalization part. The Module 02 calculates the different clusters and allocates different objects which are dealt with to different clusters in accordance to the distance from the cluster center. The iterative relocation process allows the K Means algorithm to change the value of the cluster center constantly and it keeps on updating the cluster center



***45***

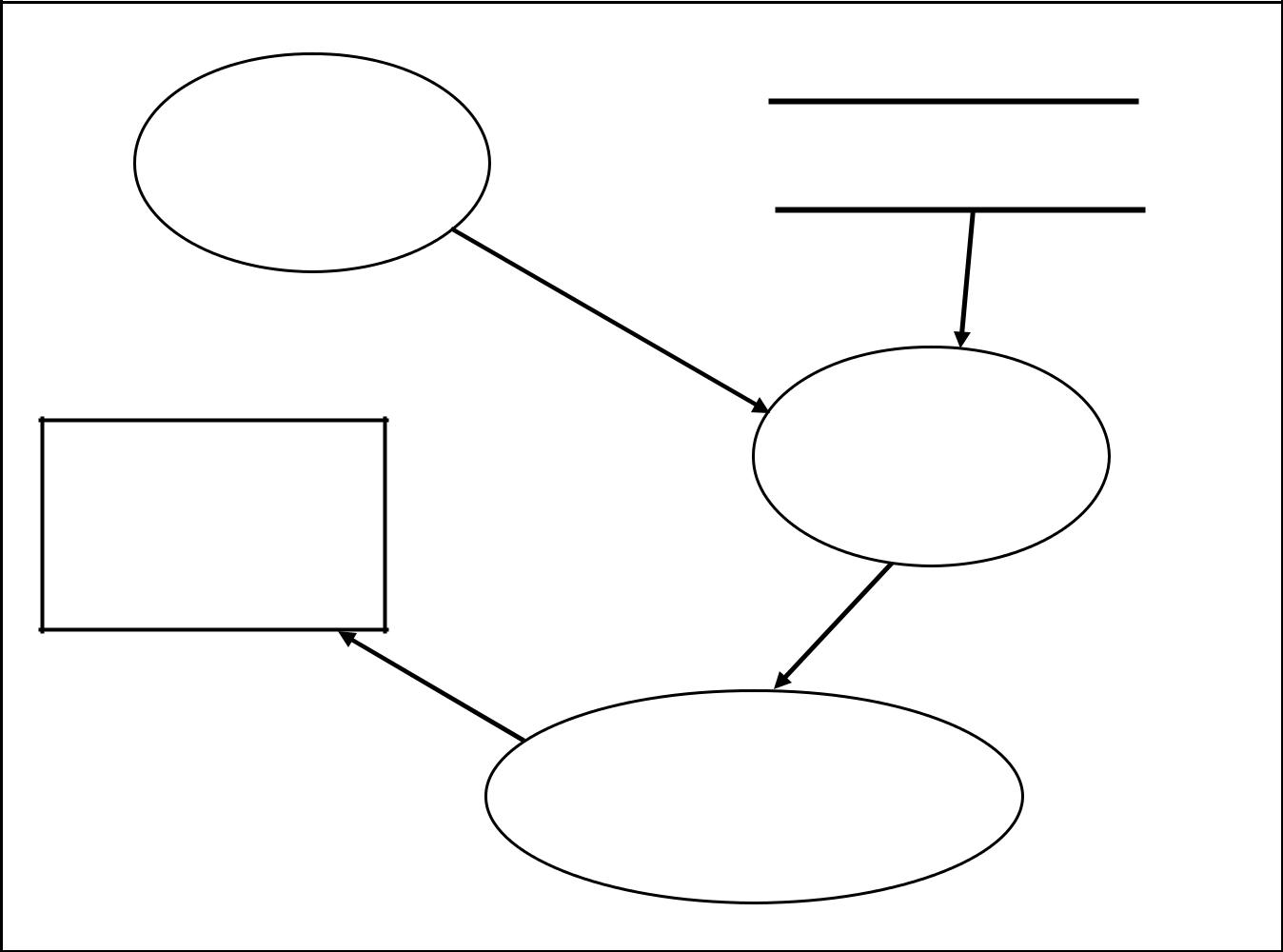
as the iteration process continues. The iteration process ensures that the new distances are calculated with respect to all the objects in the cluster and also with objects which are present in that cluster vicinity but belongs to another cluster. If the distance is less than its own cluster center then is is absorbed into another cluster to which it is nearer.

This process leads to successful allocation of clusters to every objects present in the set, but fails to contain equal number of objects in each of the clusters. Thus to ensure all the cluster have equal number of objects in them, the Equalization process comes into the forefront. The Module 3 of the Equalization phase ensures all the clusters objects are stored in definite data structures and are equally distributed among them. The Front End 2 helps to input the cluster objects and as per to their cluster numbers, with respective A\_SCORE and B\_SCORE cluster center averages. Then they are calculated as per to the distance and are redistributed at the end giving rise to the final set of clusters with equal number of objects present in them.

It is important to keep in mind that the number of initial objects so entered are to be equally divided by the total number of clusters present. If unequal objects are supplied to the cluster numbers then it may not give the required result as per to the need of the program or the system. This process is designed to ensure that there is equal mixture of students in the class which belong to different levels of adaptabilities as per to their knowledge and their ability to learn. The more diversity can be introduced in a class the more holistic the growth of the children will be, not only in area of academics but also in personality development, leading to imparting a complete education to them.



***46***



Front End 2:

cluster details Details from DB of objects

Accessing DB

Inserting data

Storing the

data in linked

Final Equalized lists.

Clusters

Equalize

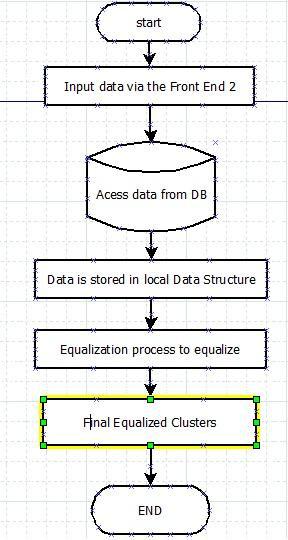
Final Output.

Equalization Process to

have clusters with equal

number of objects.

Fig: DFD for the MODULE 03



Flow Chart:



***47***

**ALGORITHM:**

1. Start
2. Insert the data in the Front End 2.
3. Insert cluster centers of the clusters.
4. Insert the objects in respect to the clusters
5. Insert the A\_SCORE, B\_SCORE center averages.
6. Access the Data Base to get the information on the objects and the respective A\_SCORE and B\_SCORE.
7. Store the data obtained in the located Data Structure, here a linked list.
8. Start the Equalization process to get the equalized clusters, with the equal number of objects in all the clusters.
9. The resultant Equalized clusters so obtained are finally implemented in real life.
10. The next implementation phase take account of the result and implements so.
11. Stop.

**4.5.MODULE 04:**

The last or the end module ensures that the implementation of the whole program is done in an effective manner and the result is checked for the final output. The Equalized clusters so formed in the Module 03 gives the details of the objects (here students) who are to be placed in which clusters, and in accordance to that information, the setting of the students are done in the separate groups. Once the groups are formed with the required objects in them then the process of learning starts. As this is an educational project the whole aspect of this project is to ensure the growth and learning of the students be in



***48***

an appropriate manner, which means the learning both from their teachers and from their peers. It has to be kept in mind that all students in a class are not equivalent in terms of their ability to learn new techniques or to implement them in real fields as quick as some of their friends may. Whereas in a similar fashion a student who may excel in academics but fails to generate interest in sports or other curricular activities. Thus a heterogeneous mixture of students with different talents exists in a class. To ensure proper nurturing of their talents it is quite important that they co-exist in the class and help each other to grow in all sphere including academics, sports and other curricular activities.

Thus in this phase they are actually made to sit in groups which consists friends from different talent pool and it definitely inspires them to learn from their friends and know things which are unknown to them.

Therefore in the last phase of the project the implementation make sure whether the whole process of the collaborative learning in groups can be implemented in real life or not. This is the phase where the students sited in groups are made to take another exam but at this time they are to answer the questions in discussion with their friends. Then the result of the group is collected and checked for the effectiveness of the system. The process can only be called a success if the data so obtained after that states that there has been a significant improvement in their performance, if so then this process can be implemented in real life easily. Here we have obtained the data which is collected from a class full of students and their previous result is kept in the record and the final record performances are tallied to ensure that they have learnt something new from their peers which being taught in groups.

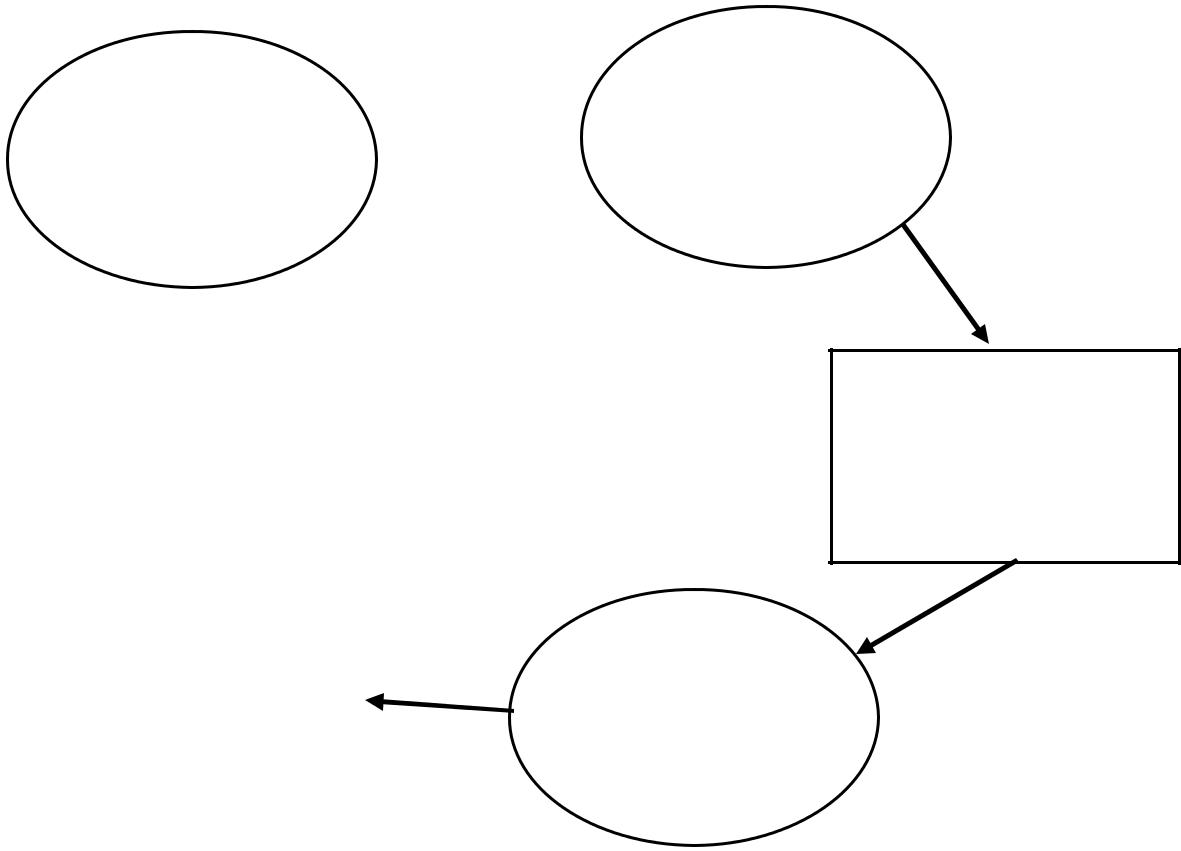


***49***

Let the

students be

Groups



Students take

test in groups.

sited in groups

Marks

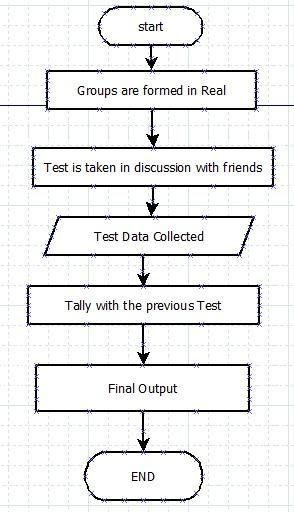
Test data marks

collected

|  |  |  |
| --- | --- | --- |
|  | Result | comparison |
| Final Output |  | Marks tallied |
|  |  |  |

Fig: DFD of the final module 4.

Flowchart:



***50***

**ALGORITHM:**

1. Start.
2. Get the equalized clusters from the Module 03.
3. Make the objects (here students) be arranged in groups as per the mentioned groups.
4. Let the students take test, with discussion among group mates.
5. Collect the scores from the test.
6. Compare the present test data with the previous result.
7. Final result.
8. Stop.

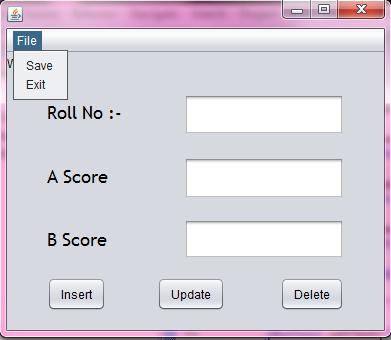
**4.6.** **User Interface Design:**

The user interface over here is been designed with the help of Net Beans, and then the Front End 1 code is written with the help Eclipse Luna, and all the coding has been done in Java. The initial marks of the data of the two subjects are inserted through the Front End 1 to the back end Database. The first front end has the facilities of insert, update and delete the values stored in the backend at the database. The following screen shots of the front end taken at different stages of inserting the data is described here. The functions done by the first front end is as follows:

1. Insert.
2. Delete.
3. Update.



***51***

Fig: The Basic Front

End 1 design.

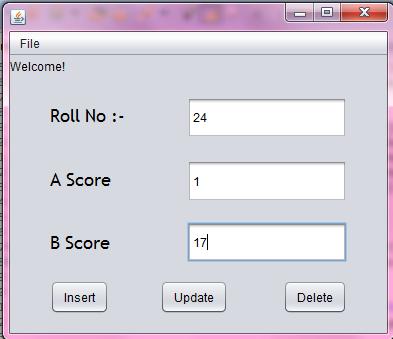


Fig: Insertion into the Database.



***52***

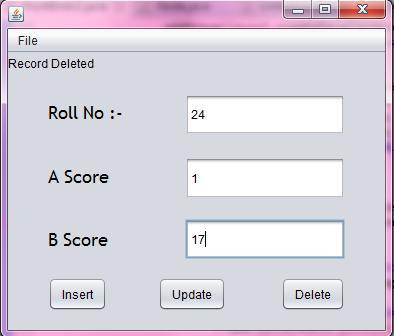


Fig: Deletion into the Database.

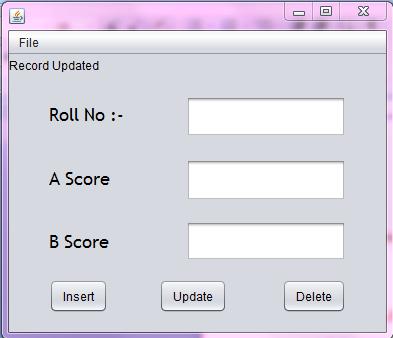


Fig: Updating the data into the database



***53***

The Second Front End 2 is the one which is used to equalize the cluster groups

and is also designed in similar fashion like the Front End 1, developed with the

help of Eclipse Luna and coded in Java.

**4.7.SECURITY ISSUES:**

The working of the project is done in such a manner such that it can been openly used by students, teachers or people related in the field of academia. Thus as it has been ensured that all eager students, scholars, teachers can have access to data, store data in database or initiate the process of clustering to obtain un equalized groups and then run the Equalization process to get the Equalized groups at the End.

The access to the Data Base (Oracle 10g XE) is entered through the username and password (here: system, system respectively).

**4.8.TEST CASE DESIGN:**

To implement the Test Case in this project, we intended to implement the whole process practically in real life scenario. Thus to experience the whole process the best way would have been to ensure the implementation is performed in a class full of students. We did just the same in the similar fashion, the test subjects in this case are being the students from the Final year Post graduate students of Department of Computer Science, St. Xavier’s College Kolkata. The total numbers of students who took part in the survey were 30, who were initially provided with an “X” coded paper and were allowed to fill in for the questionnaires. The paper was divided in Section A consisting of 5 general Knowledge and Section B consisting of 5 Mathematical Ability questions. Thus group A and B reflected as A\_SCORE and B\_SCORE in the data clustering method.



***54***

The Second phase of the Implementation process is the one which takes place after the clustering is formed and the Equalization is implemented, thus giving rise to the equalized groups of students consisting in equal numbers in every groups. This is done in the paper marked as “Y”, again divided in two section A consisting of General Knowledge and section B consisting of the Mathematical ability and Problem solving skills. Thus this is the User set Data 1 and 2 explained later, the following is the Test Case Data, which consists of the Student id and the A\_score and B\_score.

Test Case Data set 1:

|  |  |  |
| --- | --- | --- |
| serial |  |  |
| no. | A value | B value |
| 1 | 7 | 15 |
| 2 | 27 | 34 |
| 3 | 18 | 6 |
| 4 | 15 | 24 |
| 5 | 13 | 7 |
| 6 | 17 | 23 |
| 7 | 21 | 6 |
| 8 | 5 | 15 |
| 9 | 6 | 11 |
| 10 | 14 | 8 |
| 11 | 8 | 3 |
| 12 | 2 | 25 |
| 13 | 17 | 27 |
| 14 | 7 | 6 |
| 15 | 28 | 35 |
| 16 | 26 | 37 |
| 17 | 9 | 15 |
| 18 | 16 | 21 |
| 19 | 25 | 39 |
| 20 | 19 | 12 |
| 21 | 4 | 35 |
| 22 | 22 | 21 |
| 23 | 9 | 18 |
| 24 | 1 | 17 |



***55***

**CHAPTER 05**

**IMPLEMENTATION AND TESTING:**

**5.1. Coding Details**

The project has been divided into 4 modules.

1. Front End Tool- This Front End tool connects the user to the central database (Oracle Database). The user can input, update and delete the two scores of each student from this module in the database. While inserting records in the database the roll number is automatically incremented for the user if he/she requires that. Manual change to the

‘Roll No.’ field is also allowed.

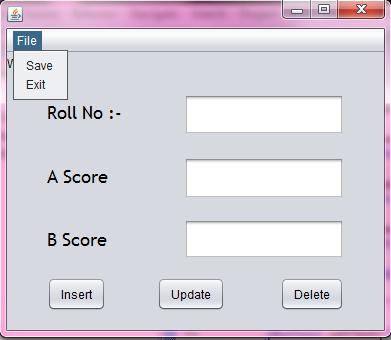


Figure – Front End Tool



***56***

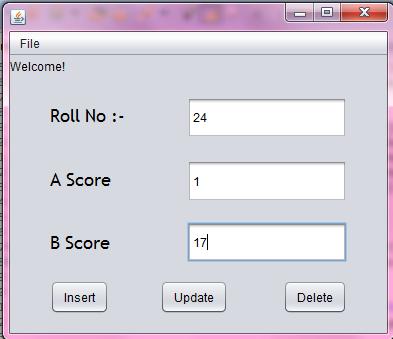


Figure –Inserting record

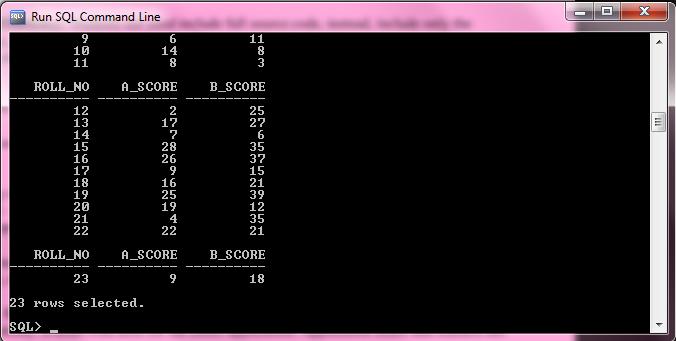


Figure –Database before insertion



***57***

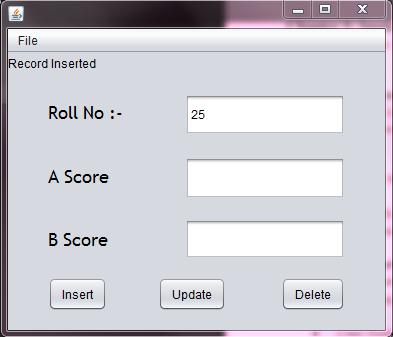


Figure –Tool after successful insertion

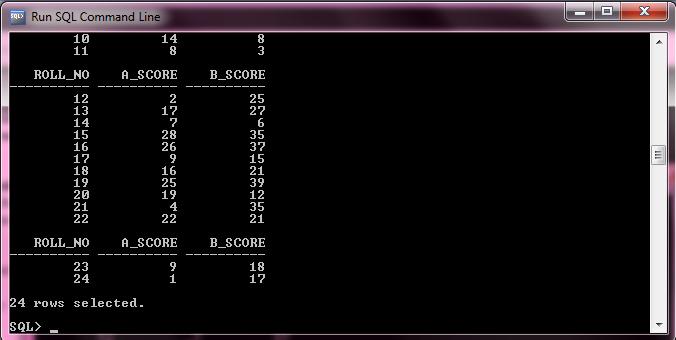


Figure –Database after insertion

*jButton1*.addActionListener(**new**ActionListener(){**public void** actionPerformed(ActionEvent ae){

**try**

{



***58***

//Connection to Oracle Database

Class.*forName*("oracle.jdbc.driver.OracleDriver");

Connection

con=DriverManager.*getConnection*("jdbc:oracle:thin:@127.0.0.1:1521

:XE","system","system");

String strqry="insert into StudentData values(?,?,?)"; PreparedStatement pst=con.prepareStatement(strqry);

pst.setInt(1,Integer.*parseInt*((*jTextField1*.getText())));

pst.setInt(2,Integer.*parseInt*((*jTextField2*.getText()))); pst.setInt(3,Integer.*parseInt*((*jTextField3*.getText())));

**int** res=pst.executeUpdate();

**if**(res>0)

{

**int** n=Integer.*parseInt*(*jTextField1*.getText());

String s="";

s+=(++n);

*jTextField1*.setText(s);

*jTextField2*.setText("");

*jTextField3*.setText("");

*jTextField2*.requestFocus();

*jLabel4*.setText("Record Inserted");

}

**else**

*jLabel4*.setText("Record Not Inserted");

}



***59***

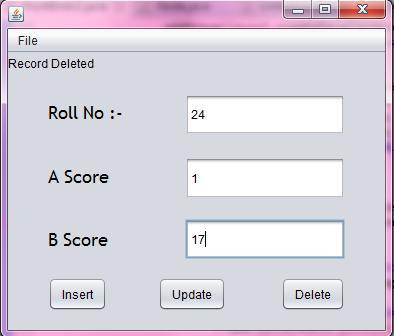


Figure –Deletion of record

*jButton3*.addActionListener(**new**ActionListener(){

**public void** actionPerformed(ActionEvent

ae){

**try**

{

Class.*forName*("oracle.jdbc.driver.OracleDriver"); Connection

con=DriverManager.*getConnection*("jdbc:oracle:thin:@127.0.

0.1:1521:XE","system","system");

String strqry="delete from StudentData where Roll\_No=?";

PreparedStatement pst=con.prepareStatement(strqry); pst.setInt(1,Integer.*parseInt*((*jTextField1*.getText() )));

**int** res=pst.executeUpdate();

**if**(res>0)

{

*jTextField1*.setText("");

*jTextField2*.setText("");

*jTextField3*.setText("");

*jLabel4*.setText("Record Deleted");



***60***

}

**else**

*jLabel4*.setText("Record Not Deleted");

}

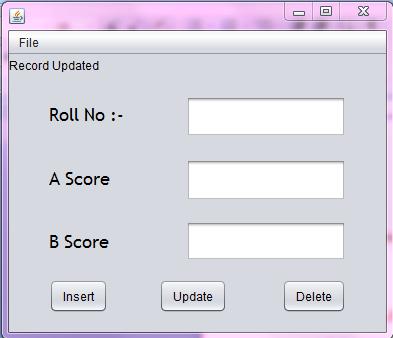


Figure **–** Updating Record

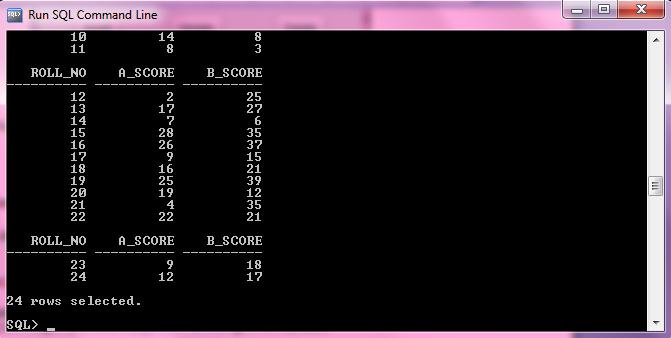


Figure –Database Updated



***61***

*jButton2*.addActionListener(**new**ActionListener(){

**public void** actionPerformed(ActionEvent ae){

**try**

{

Class.*forName*("oracle.jdbc.driver.OracleDriver");

Connection

con=DriverManager.*getConnection*("jdbc:oracle:thin:@127.0.0.1:1521:XE","s ystem","system");

String strqry="update StudentData set A\_Score=?, B\_Score=?

where Roll\_No=?";

PreparedStatement pst=con.prepareStatement(strqry); pst.setInt(3,Integer.*parseInt*((*jTextField1*.getText()))); pst.setInt(1,Integer.*parseInt*((*jTextField2*.getText()))); pst.setInt(2,Integer.*parseInt*((*jTextField3*.getText())));

**int** res=pst.executeUpdate();

**if**(res>0)

{

*jTextField1*.setText("");

*jTextField2*.setText("");

*jTextField3*.setText("");

*jLabel4*.setText("Record Updated");

}

**else**

*jLabel4*.setText("Record Not Updated");

}

**The over all code is as follows:**

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.\*;

/\*\*

\*

* @author Sagnik Raychowdhuri \*/

@SuppressWarnings("serial")



***62***

public class FrontEndv2 extends javax.swing.JFrame {

/\*\*

* Creates new form FrontEndv2 \*/

public FrontEndv2() {

initComponents();

}

/\*\*

* This method is called from within the constructor to initialize the form.
* WARNING: Do NOT modify this code. The content of this method is always
* regenerated by the Form Editor.

\*/

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jMenuBar1 = new javax.swing.JMenuBar();

jMenu1 = new javax.swing.JMenu();

jMenuItem1 = new javax.swing.JMenuItem();

jMenuItem2 = new javax.swing.JMenuItem();

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

jLabel1.setFont(new java.awt.Font("Trebuchet MS", 0, 18)); // NOI18N jLabel1.setText("Roll No :-");

jLabel2.setFont(new java.awt.Font("Trebuchet MS", 0, 18)); // NOI18N jLabel2.setText("A Score");

jLabel3.setFont(new java.awt.Font("Trebuchet MS", 0, 18)); // NOI18N jLabel3.setText("B Score");

jLabel4.setText("Database Connected");

jButton1.setText("Insert");



***63***

jButton2.setText("Update");

jButton3.setText("Delete");

jMenu1.setText("File");

jMenuItem1.setText("Save");

jMenu1.add(jMenuItem1);

jMenuItem2.setText("Exit");

jMenu1.add(jMenuItem2);

jMenuBar1.add(jMenu1);

setJMenuBar(jMenuBar1);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(layout.createSequentialGroup()

.addGap(40, 40, 40)

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

.addGroup(layout.createSequentialGroup()

.addComponent(jButton1)

.addGap(51, 51, 51)

.addComponent(jButton2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton3))

.addGroup(layout.createSequentialGroup()

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

.addComponent(jLabel3,

javax.swing.GroupLayout.DEFAULT\_SIZE,

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

.addComponent(jLabel2,

javax.swing.GroupLayout.DEFAULT\_SIZE,

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

.addComponent(jLabel1,

javax.swing.GroupLayout.DEFAULT\_SIZE, 119, Short.MAX\_VALUE))

.addGap(18, 18, 18)



***64***

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L EADING)

.addComponent(jTextField1,

javax.swing.GroupLayout.PREFERRED\_SIZE, 160, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField2,

javax.swing.GroupLayout.PREFERRED\_SIZE, 160, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField3,

javax.swing.GroupLayout.PREFERRED\_SIZE, 160, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addContainerGap(40, Short.MAX\_VALUE))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel4)

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

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel4,

javax.swing.GroupLayout.PREFERRED\_SIZE, 22, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B ASELINE)

.addComponent(jLabel1,

javax.swing.GroupLayout.PREFERRED\_SIZE, 41, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField1,

javax.swing.GroupLayout.PREFERRED\_SIZE, 41, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(20, 20, 20)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B ASELINE)

.addComponent(jLabel2,

javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField2,

javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(20, 20, 20)

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

.addComponent(jLabel3,

javax.swing.GroupLayout.DEFAULT\_SIZE, 40, Short.MAX\_VALUE)



***65***

.addComponent(jTextField3))

.addGap(18, 18, 18)

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

.addComponent(jButton2,

javax.swing.GroupLayout.DEFAULT\_SIZE, 34, Short.MAX\_VALUE)

.addComponent(jButton1,

javax.swing.GroupLayout.DEFAULT\_SIZE,

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

.addComponent(jButton3,

javax.swing.GroupLayout.DEFAULT\_SIZE,

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

.addContainerGap(19, Short.MAX\_VALUE))

);

jButton1.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent ae){

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection

con=DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:XE","s

ystem","system");

String strqry="insert into

StudentData values(?,?,?)";

PreparedStatement

pst=con.prepareStatement(strqry);

pst.setInt(1,Integer.parseInt((jTextField1.getText())));

pst.setInt(2,Integer.parseInt((jTextField2.getText())));

pst.setInt(3,Integer.parseInt((jTextField3.getText())));

int res=pst.executeUpdate();

if(res>0)

{

int

n=Integer.parseInt(jTextField1.getText());

String s="";

s+=(++n);

jTextField1.setText(s);

jTextField2.setText("");

jTextField3.setText("");

jTextField2.requestFocus();



***66***

jLabel4.setText("Record

Inserted");

}

else

jLabel4.setText("Record

Not Inserted");

}

catch(ClassNotFoundException es)

{

es.printStackTrace();

}

catch(SQLException ess)

{

ess.printStackTrace();

}

}

});

jButton2.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent ae){

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection

con=DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:XE","s

ystem","system");

String strqry="update

StudentData set A\_Score=?, B\_Score=? where Roll\_No=?"; PreparedStatement

pst=con.prepareStatement(strqry);

pst.setInt(3,Integer.parseInt((jTextField1.getText())));

pst.setInt(1,Integer.parseInt((jTextField2.getText())));

pst.setInt(2,Integer.parseInt((jTextField3.getText())));

int res=pst.executeUpdate();

if(res>0)

{

jTextField1.setText("");

jTextField2.setText("");

jTextField3.setText("");

jLabel4.setText("Record

Updated");

}



***67***

else

jLabel4.setText("Record

Not Updated");

}

catch(ClassNotFoundException es)

{

es.printStackTrace();

}

catch(SQLException ess)

{

ess.printStackTrace();

}

}

});

jButton3.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent ae){

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection

con=DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:XE","s

ystem","system");

String strqry="delete from

StudentData where Roll\_No=?";

PreparedStatement

pst=con.prepareStatement(strqry);

pst.setInt(1,Integer.parseInt((jTextField1.getText())));

int res=pst.executeUpdate();

if(res>0)

{

jTextField1.setText("");

jTextField2.setText("");

jTextField3.setText("");

jLabel4.setText("Record

Deleted");

}

else

jLabel4.setText("Record

Not Deleted");

}

catch(ClassNotFoundException es)

{



***68***

es.printStackTrace();

}

catch(SQLException ess)

{

ess.printStackTrace();

}

}

});

jMenuItem1.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent ae){

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection

con=DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:XE","s

ystem","system");

String strqry="commit";

PreparedStatement

pst=con.prepareStatement(strqry);

int res=pst.executeUpdate();

if(res>0)

jLabel4.setText("Database Saved");

}

catch(ClassNotFoundException es)

{

es.printStackTrace();

}

catch(SQLException ess)

{

ess.printStackTrace();

}

}

});

jMenuItem2.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent ae){

System.exit(0);

}

});



***69***

pack();

}// </editor-fold>

public static void main(String args[]) {

try {

for (javax.swing.UIManager.LookAndFeelInfo info :

javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) { javax.swing.UIManager.setLookAndFeel(info.getClassName()); break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(FrontEndv2.class.getName()).log(java.util.

logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(FrontEndv2.class.getName()).log(java.util.

logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(FrontEndv2.class.getName()).log(java.util.

logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(FrontEndv2.class.getName()).log(java.util.

logging.Level.SEVERE, null, ex);

}

/\* Create and display the form \*/ java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new FrontEndv2().setVisible(true);

}

});

}

* Variables declaration - do not modify private static javax.swing.JButton jButton1; private static javax.swing.JButton jButton2; private static javax.swing.JButton jButton3; private static javax.swing.JLabel jLabel1; private static javax.swing.JLabel jLabel2; private static javax.swing.JLabel jLabel3; private static javax.swing.JLabel jLabel4; private static javax.swing.JMenu jMenu1;



***70***

private static javax.swing.JMenuBar jMenuBar1;

private static javax.swing.JMenuItem jMenuItem1;

private static javax.swing.JMenuItem jMenuItem2;

private static javax.swing.JTextField jTextField1;

private static javax.swing.JTextField jTextField2;

private static javax.swing.JTextField jTextField3;

// End of variables declaration

}

1. **‘R’ Module-**

library("RODBC")

* myconn2<- odbcConnect("XE",uid="system",pwd="system")
* SDATA<-sqlQuery(myconn2,"select A\_Score,B\_Score from Student")
* result=kmeans(SDATA,6)
* plot(SDATA[c("A\_SCORE","B\_SCORE")],col=result$cluster)
* result

One of R's core features is the ability to work with a variety of tools and data sources. Server side R is very often used to connect to Microsoft SQL Server as well as other popular relational databases. We here at RStudio have personally used RStudio Server this way for many years and can attest to its utility.



***71***

Querying a database from a server running R requires three things: (1) Network security that allows you to communicate between the machines; (2) Drivers installed on the R server; and (3) Configurations that allow you to connect from R.

* Load RODBC package library(RODBC)
* Create a connection to the database called "channel"

channel <- odbcConnect("DATABASE", uid="USERNAME", pwd="PASSWORD", believeNRows=FALSE)

* Check that connection is working (Optional) odbcGetInfo(channel)
* Find out what tables are available (Optional) Tables <- sqlTables(channel, schema="SCHEMA")
* Query the database and put the results into the data frame "dataframe" dataframe <- sqlQuery(channel, "

SELECT \* FROM SCHEMA.DATATABLE")

* 1. Equalization module- This module now takes account of the initial clusters formed by the R module. The cluster centres and cluster number



***72***

of each student is inserted into the program that optimize the clusters

and allots equal number of students to each cluster.

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import java.sql.PreparedStatement;**

**import java.sql.ResultSet;**

**import java.sql.SQLException;**

**/\***

* **To change this license header, choose License Headers in Project Properties.**
* **To change this template file, choose Tools | Templates**
* **and open the template in the editor.**

**\*/**

**/\*\***

**\***

* **@author Sagnik Raychowdhuri \*/**

**@SuppressWarnings("serial")**

**public class FrontEnd2 extends javax.swing.JFrame {**

**/\*\***

* **Creates new form FrontEnd2 \*/**

**public FrontEnd2() {**

**initComponents();**

**}**

**/\*\***

* **This method is called from within the constructor to initialize the form.**
* **WARNING: Do NOT modify this code. The content of this method is always**
* **regenerated by the Form Editor.**

**\*/**

* **<editor-fold defaultstate="collapsed" desc="Generated Code"> private void initComponents() {**

**jLabel1 = new javax.swing.JLabel();**

**jLabel2 = new javax.swing.JLabel();**

**jLabel3 = new javax.swing.JLabel();**

**jLabel4 = new javax.swing.JLabel();**

**jLabel5 = new javax.swing.JLabel();**



***73***

**jTextField1 = new javax.swing.JTextField();**

**jTextField2 = new javax.swing.JTextField();**

**jButton1 = new javax.swing.JButton();**

**jTextField3 = new javax.swing.JTextField();**

**jTextField4 = new javax.swing.JTextField();**

**jLabel6 = new javax.swing.JLabel();**

**jTextField5 = new javax.swing.JTextField();**

**jButton2 = new javax.swing.JButton();**

**jButton3 = new javax.swing.JButton();**

**setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CL OSE);**

**jLabel1.setText("No. of Students");**

**jLabel2.setText("No. of Cluster");**

**jButton1.setText("Submit");**

**jButton1.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent evt)**

**{**

**no=Integer.parseInt(jTextField1.getText());**

**clstr=Integer.parseInt(jTextField2.getText());**

**jLabel3.setText("A Score of 1");**

**jLabel4.setText("B Score of 1");**

**jButton2.setText("Submit");**

**jTextField3.requestFocus();**

**cluster=new double[2][clstr+1];**

**obj=new LinkedList[clstr+1];**

**i=1; k=1;**

**}**

**});**

**jButton2.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent evt)**

**{**

**if(i<clstr)**

**{**

**cluster[0][i]=Double.parseDouble(jTextField3.getText());**

**cluster[1][i++]=Double.parseDouble(jTextField4.getText());**

**String s1="A Score of "+i;**

**String s2="B Score of "+i;**

**jLabel3.setText(s1);**

**jTextField3.setText("");**

**jLabel4.setText(s2);**

**jTextField4.setText("");**

**jTextField3.requestFocus();**

**for(int j=1;j<=clstr;j++)**

**{**



***74***

**obj[j]=new LinkedList();**

**}**

**}**

**if(i==clstr)**

**{**

**cluster[0][i]=Double.parseDouble(jTextField3.getText());**

**cluster[1][i]=Double.parseDouble(jTextField4.getText());**

**for(int i=1;i<=clstr;i++)**

**{**

**System.out.println(cluster[0][i]+" "+cluster[1][i]);**

**}**

**jLabel5.setText("Enter Cluster No. of Student");**

**String s=""+k;**

**jLabel6.setText(s);**

**jButton3.setText("Submit");**

**jTextField5.requestFocus();**

**}**

**}**

**});**

**jButton3.addActionListener(new java.awt.event.ActionListener() { public void actionPerformed(java.awt.event.ActionEvent evt)**

**{**

**try**

**{**

**if(k<=no)**

**{**

**Class.forName("oracle.jdbc.driver.OracleDriver"); Connection**

**con=DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:X E","system","system");**

**String strqry="select \*** **from**

**StudentData where Roll\_No=?";**

**int as=0,bs=0;**

**pst=con.prepareStatement(strqry);**

**pst.setInt(1,Integer.parseInt(jLabel6.getText()));**

**res=pst.executeQuery();**

**while(res.next())**

**{**

**as=res.getInt(2);**

**bs=res.getInt(3);**

**}**



***75***

**int**

**clusterno=Integer.parseInt(jTextField5.getText());**

**double**

**dist=Math.sqrt(Math.pow(cluster[0][clusterno]-as,**

**2)+Math.pow(cluster[1][clusterno]-bs, 2));**

**obj[clusterno].add(k++,dist);**

**String s=""+k;**

**jLabel6.setText(s);**

**jTextField5.setText("");**

**jTextField5.requestFocus();**

**}**

**if(k>no)**

**{**

**for(int i=1;i<=clstr;i++)**

**{**

**System.out.print("\nCluster "+i+"- ");**

**obj[i].display();**

**}**

**for(int i=1;i<clstr;i++)**

**for(int**

**j=i+1;j<=clstr;j++)**

**{**

**double temp;**

**LinkedList tmp;**

**if(cluster[0][i]>cluster[0][j])**

**{**

**temp=cluster[0][i];**

**cluster[0][i]=cluster[0][j];**

**cluster[0][j]=temp;**

**temp=cluster[1][i];**

**cluster[1][i]=cluster[1][j];**

**cluster[1][j]=temp;**

**tmp=obj[i];**

**obj[i]=obj[j];**

**obj[j]=tmp;**



***76***

**}**

**}**

**for(int i=1;i<clstr;i++)**

**{**

**if(obj[i].count()>=(no/clstr))**

**{**

**while(obj[i].count()!=(no/clstr))**

**{**

**int**

**data=obj[i].max();**

**Node**

**ptr=obj[i].remove(data);**

**obj[i+1].add(ptr);**

**}**

**}**

**else**

**{**

**k=i+1;**

**while(obj[i].count()!=(no/clstr))**

**{**

**if(obj[k].count()==0)**

**k++;**

**else**

**{**

**int**

**data=obj[k].max();**

**Node ptr=obj[k].remove(data);**

**obj[i].add(ptr);**

**}**

**}**

**}**

**}**

**System.out.println("\n\n");**

**for(int i=1;i<=clstr;i++)**

**{**

**System.out.print("\nCluster "+(i)+" : ");**



***77***

**obj[i].display();**

**}**

**}**

**}**

**catch(ClassNotFoundException es)**

**{**

**es.printStackTrace();**

**}**

**catch(SQLException ess)**

**{**

**ess.printStackTrace();**

**}**

**}**

**});**

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

**getContentPane().setLayout(layout);**

**layout.setHorizontalGroup(**

**layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI NG)**

**.addGroup(layout.createSequentialGroup()**

**.addGap(31, 31, 31)**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align ment.LEADING)**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align**

**ment.TRAILING)**

**.addGroup(layout.createSequentialGroup()**

**.addComponent(jLabel6,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 99, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA TED)**

**.addComponent(jTextField5,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 46, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addGap(18, 18, 18)**

**.addComponent(jButton3,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 80, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addGap(57, 57, 57))**

**.addGroup(layout.createSequentialGroup()**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align ment.LEADING, false)**



***78***

**.addComponent(jLabel1,**

**javax.swing.GroupLayout.DEFAULT\_SIZE, 100, Short.MAX\_VALUE)**

**.addComponent(jLabel2,**

**javax.swing.GroupLayout.DEFAULT\_SIZE,**

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

**.addComponent(jLabel3,**

**javax.swing.GroupLayout.DEFAULT\_SIZE,**

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

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align ment.LEADING, false)**

**.addGroup(layout.createSequentialGroup()**

**.addGap(51, 51, 51)**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align ment.LEADING, false)**

**.addComponent(jTextField1)**

**.addComponent(jTextField2,**

**javax.swing.GroupLayout.DEFAULT\_SIZE, 107, Short.MAX\_VALUE))**

**.addGap(86, 86, 86))**

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

**.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE LATED)**

**.addComponent(jTextField3,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 93, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA TED, 31, Short.MAX\_VALUE)**

**.addComponent(jLabel4,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addGap(10, 10, 10)))))**

**.addGroup(layout.createSequentialGroup()**

**.addComponent(jLabel5,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 242, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE LATED)))**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align ment.TRAILING)**

**.addComponent(jButton2,**

**javax.swing.GroupLayout.DEFAULT\_SIZE,**

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



***79***

**.addGroup(javax.swing.GroupLayout.Alignment.LEADING,**

**layout.createSequentialGroup()**

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

**.addComponent(jButton1,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 91, javax.swing.GroupLayout.PREFERRED\_SIZE))**

**.addComponent(jTextField4,**

**javax.swing.GroupLayout.Alignment.LEADING))**

**.addGap(32, 32, 32))**

**);**

**layout.setVerticalGroup(**

**layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI NG)**

**.addGroup(layout.createSequentialGroup()**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align**

**ment.LEADING)**

**.addGroup(layout.createSequentialGroup()**

**.addGap(19, 19, 19)**

**.addComponent(jLabel1,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE))**

**.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,**

**layout.createSequentialGroup()**

**.addGap(20, 20, 20)**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align**

**ment.BASELINE)**

**.addComponent(jTextField1,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 22, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addComponent(jButton1,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE))))**

**.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA TED)**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align**

**ment.BASELINE)**

**.addComponent(jLabel2,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addComponent(jTextField2,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE))**

**.addGap(45, 45, 45)**



***80***

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align**

**ment.BASELINE)**

**.addComponent(jLabel3,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addComponent(jLabel4,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addComponent(jTextField3,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addComponent(jTextField4,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE))**

**.addGap(18, 18, 18)**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align**

**ment.LEADING)**

**.addComponent(jLabel5,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 37, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addComponent(jButton2,**

**javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE))**

**.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA TED)**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Align ment.LEADING, false)**

**.addComponent(jLabel6,**

**javax.swing.GroupLayout.DEFAULT\_SIZE,**

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

**.addComponent(jButton3,**

**javax.swing.GroupLayout.DEFAULT\_SIZE,**

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

**.addComponent(jTextField5,**

**javax.swing.GroupLayout.DEFAULT\_SIZE, 39, Short.MAX\_VALUE))**

**.addContainerGap(23, Short.MAX\_VALUE))**

**);**

**pack();**

**}// </editor-fold>**

**/\*\***

* **@param args the command line arguments \*/**

**public static void main(String args[]) {**

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



***81***

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

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

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

**\*/**

**try {**

**for** **(javax.swing.UIManager.LookAndFeelInfo** **info** **:**

**javax.swing.UIManager.getInstalledLookAndFeels()) {**

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

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

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

**java.util.logging.Logger.getLogger(FrontEnd2.class.getName()).log(java.u til.logging.Level.SEVERE, null, ex);**

**} catch (InstantiationException ex) {**

**java.util.logging.Logger.getLogger(FrontEnd2.class.getName()).log(java.u til.logging.Level.SEVERE, null, ex);**

**} catch (IllegalAccessException ex) {**

**java.util.logging.Logger.getLogger(FrontEnd2.class.getName()).log(java.u til.logging.Level.SEVERE, null, ex);**

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

**java.util.logging.Logger.getLogger(FrontEnd2.class.getName()).log(java.u til.logging.Level.SEVERE, null, ex);**

**}**

**//</editor-fold>**

**/\* Create and display the form \*/ java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new FrontEnd2().setVisible(true);**

**}**

**});**

**}**

* **Variables declaration - do not modify private javax.swing.JButton jButton1; private javax.swing.JButton jButton2; private javax.swing.JButton jButton3; private javax.swing.JLabel jLabel1; private javax.swing.JLabel jLabel2; private javax.swing.JLabel jLabel3;**



***82***

**private javax.swing.JLabel jLabel4;**

**private javax.swing.JLabel jLabel5;**

**private javax.swing.JLabel jLabel6;**

**private javax.swing.JTextField jTextField1;**

**private javax.swing.JTextField jTextField2;**

**private javax.swing.JTextField jTextField3;**

**private javax.swing.JTextField jTextField4;**

**private javax.swing.JTextField jTextField5;**

**private int no;**

**private int clstr;**

**private double cluster[][];**

**private LinkedList obj[];**

**private int i;**

**private int k;**

**static Connection con;**

**static ResultSet res;**

**static PreparedStatement pst;**

**// End of variables declaration**

**}**

The above code take input of each cluster number of the student then

calculates the distance of each student from its respective cluster centres for

future calculation.

**for**(**int** i=1;i<clstr;i++)

{

**if**(obj[i].count()>=(no/clstr))

{

/\*if present cluster has more objects than the required number it transfers the excess objects to the next cluster\*/ **while**(obj[i].count()!=(no/clstr)){

data=obj[i].max();

Node ptr=obj[i].remove(data);

obj[i+1].add(ptr);

}

}

**else**

{

k=i+1;

**while**(obj[i].count()!=(no/clstr))

{

//if the cluster is empty then transfer is continued from

the next cluster

**if**(obj[k].count()==0)

k++;

**else**

{



***83***

data=obj[k].max();

Node ptr=obj[k].remove(data);

obj[i].add(ptr);

}

}

}

}

The above code equalizes the cluster into equal heterogeneous groups.

public class LinkedList

{

protected Node start;

public LinkedList()

{

start=null;

}

public LinkedList(Node start)

{

this.start=start;

}

public void add(int roll, double dist)

{

Node temp=new Node(roll,dist,null),ptr=start;

if(ptr==null)

start=temp;

else

{

while(ptr.getLink()!=null)

ptr=ptr.getLink();

ptr.setLink(temp);

}

}

public void add(Node temp)

{

Node ptr=start;

while(ptr.getLink()!=null)

ptr=ptr.getLink();

ptr.setLink(temp);

}

public Node remove(int x)

{



***84***

Node ptr1=start,ptr2;

if(ptr1==null)

return null;

else

{

if(ptr1.getRoll()==x)

{

start=ptr1.getLink();

ptr1.setLink(null);

return ptr1;

}

else

{

ptr2=ptr1.getLink();

while(ptr2.getLink()!=null)

{

if(ptr2.getRoll()==x)

{

ptr1.setLink(ptr2.getLink());

ptr2.setLink(null);

return ptr2;

}

ptr1=ptr1.getLink();

ptr2=ptr2.getLink();

}

ptr1.setLink(null);

ptr2.setLink(null);

return ptr2;

}

}

}

public int count()

{

Node ptr=start;

int c=0;

while(ptr!=null)

{

c++;

ptr=ptr.getLink();

}

return c;

}

void display()



***85***

{

Node ptr=start;

while(ptr!=null)

{

System.out.print(ptr.getRoll()+" ");

ptr=ptr.getLink();

}

}

public int max()

{

Node ptr=start;

double max=0.0;

int data=-1;

while(ptr!=null)

{

if(ptr.getDist()>max)

{

max=ptr.getDist();

data=ptr.getRoll();

}

ptr=ptr.getLink();

}

return data;

}

}

The above code is for the linked list structure in the program.

public class Node

{

protected int roll;

protected double dist;

protected Node link;

public Node()

{

roll=0;

dist=0.0;

link=null;

}

public Node(int roll, double dist, Node link)

{

this.roll=roll;

this.dist=dist;

this.link=link;



***86***

}

public void setRoll(int roll)

{

this.roll=roll;

}

public void setDist(double dist)

{

this.dist=dist;

}

public void setLink(Node link)

{

this.link=link;

}

public int getRoll()

{

return roll;

}

public double getDist()

{

return dist;

}

public Node getLink()

{

return link;

}

}

The above code is for the node declaration in the program.



***87***

5.2. **Coding Efficiency**

The code that is being implemented here are basically java based and thus it helped us to code with quite ease, the data structure that is followed here is mixed between the linked list, array, array of object and other simple structures. The efficiency of the code could have been increased if it could have been implemented by other form like tree structure. But as we have used a simple algorithm like K means the effective result would not have been quite changed.

**5.3. Testing Approach**

Testing Approach: Testing should be according to the scheme presented in the system design chapter and should follow some suitable model – e.g., category partition, state machine-based. Both functional testing and user- acceptance testing. As per to the Schema presented in the Chapter 4 related to the Design of the system it is quite prominent that the different module are coded and tested in the mean time for the correct result. At the end when the User Date sets were used they were feed into the system first then they were analyzed accordingly after that. Thus each module was coded and was tested in intervals.

**5.4. Unit Testing**

Unit Testing: Unit testing deals with testing a unit or module as a whole. This would test the interaction of many functions but, do confine the test within one module. The unit testing phase states that the particular unit here 4 modules were tested perfectly and were run before the final integration took place. First the front end, then the R, then the equalization and finally implementation

**5.5. Integrated Testing**

Integrated Testing In the integrated testing phase all the modules were brought together and the system was run as a whole and the tested result were calculated to forward it to the equalization phase which ultimately gave the final output. The final output was tallied with the initial data to get the effectiveness of the system.



***88***

**5.6. Modification and Improvement**

Initial testing of the Equalisation algorithm generated few errors.

1. Transferring one object from would result in more than one object being transferred together.
2. Seldom would the clusters get exhausted pre-emptively leading to unequal and null groups.

Cause for errors.

1. Since each objects of every cluster links one other object removing the object having such links would result in removing all the successive object as well.
2. Transferring objects from clusters having lesser number of objects than what is required for equalization would transfer all the objects to the target cluster from the source cluster. This may lead to clusters having unsatisfied number of objects.

Modifications.

1. Since we are using Linked Data structure to store each object in each cluster, we had to reassign each link while removing an object anywhere other than the end of the list. This solved the problem of transferring multiple objects at a time.
2. We put in a check where if a cluster is empty and the target cluster does not have the required number of objects, the next cluster would transfer its objects to the target cluster.



***89***

**CHAPTER 6**

**RESULTS AND DISCUSSION**

**6.1.TEST REPORT**

The result of the Project is divided into three sections and the first output or the

Test Output and the next two are the User Output 1 & 2 respectively:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **6.1.1.** | **TEST OUTPUT**: | | |  |
|  |  |  |  |  |
| serial no. |  | A value |  | B value |
|  |  |  |  |  |
| 1 |  | 7 |  | 15 |
|  |  |  |  |  |
| 2 |  | 27 |  | 34 |
|  |  |  |  |  |
| 3 |  | 18 |  | 6 |
|  |  |  |  |  |
| 4 |  | 15 |  | 24 |
|  |  |  |  |  |
| 5 |  | 13 |  | 7 |
|  |  |  |  |  |
| 6 |  | 17 |  | 23 |
|  |  |  |  |  |
| 7 |  | 21 |  | 6 |
|  |  |  |  |  |
| 8 |  | 5 |  | 15 |
|  |  |  |  |  |
| 9 |  | 6 |  | 11 |
|  |  |  |  |  |
| 10 |  | 14 |  | 8 |
|  |  |  |  |  |
| 11 |  | 8 |  | 3 |
|  |  |  |  |  |
| 12 |  | 2 |  | 25 |
|  |  |  |  |  |
| 13 |  | 17 |  | 27 |
|  |  |  |  |  |
| 14 |  | 7 |  | 6 |
|  |  |  |  |  |
| 15 |  | 28 |  | 35 |
|  |  |  |  |  |
| 16 |  | 26 |  | 37 |
|  |  |  |  |  |
| 17 |  | 9 |  | 15 |
|  |  |  |  |  |
| 18 |  | 16 |  | 21 |
|  |  |  |  |  |
| 19 |  | 25 |  | 39 |
|  |  |  |  |  |
| 20 |  | 19 |  | 12 |
|  |  |  |  |  |
| 21 |  | 4 |  | 35 |
|  |  |  |  |  |
| 22 |  | 22 |  | 21 |
|  |  |  |  |  |
| 23 |  | 9 |  | 18 |
|  |  |  |  |  |
| 24 |  | 1 |  | 17 |
|  |  |  |  |  |

The Test Data table consisting of the Serial number and the A value and B value.



***90***

Then the process of all the module are applied on it the result is:

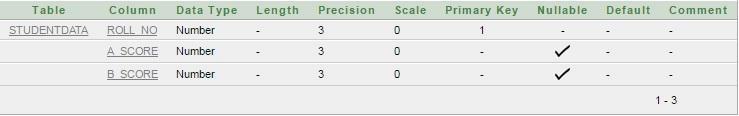


Fig: Database Table definition

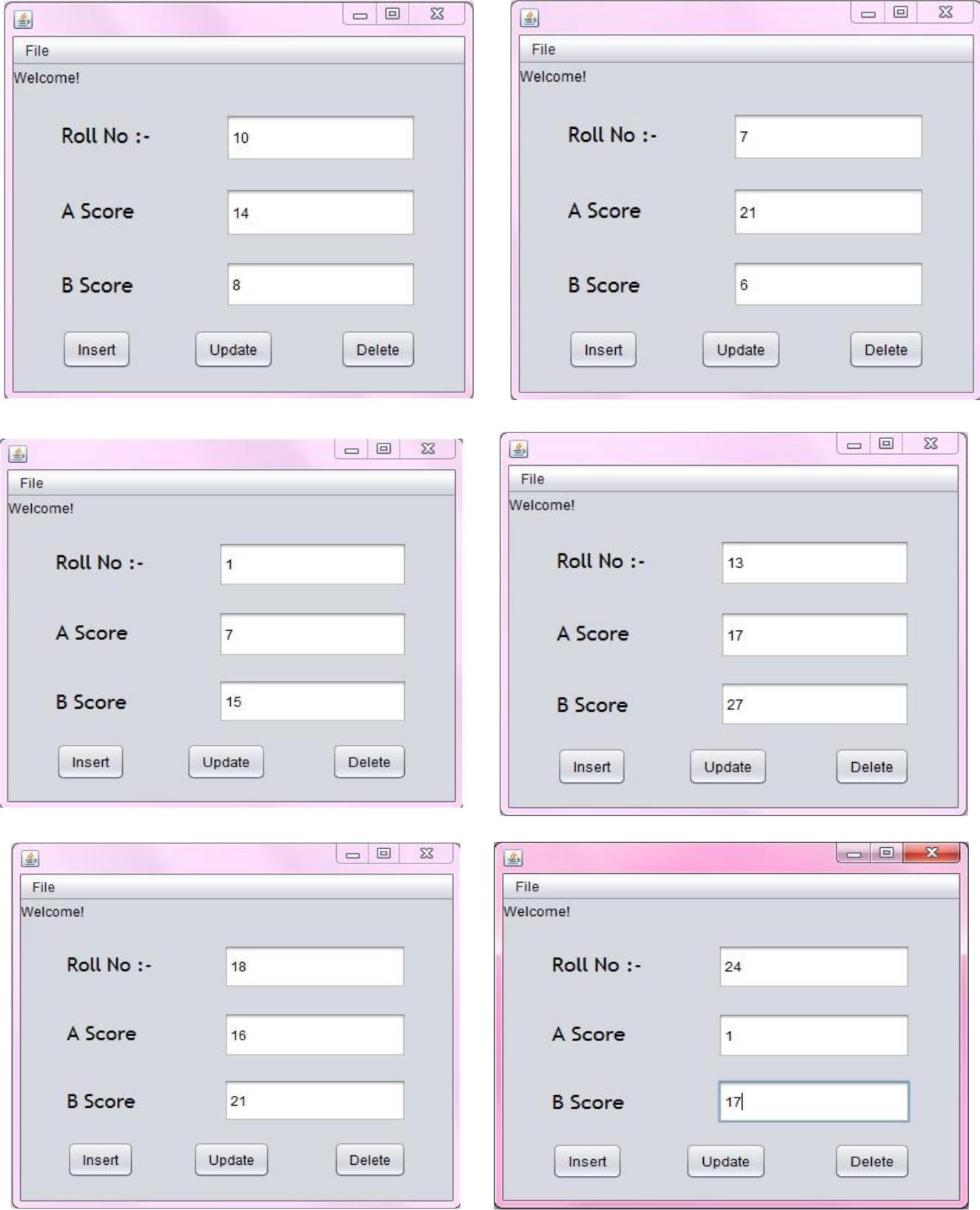


Fig: The insertion of the A score and B score into the DB via Frontend 1.



***91***

As the Data is stored into the DB the next phase is to make sure it is run properly in R, Module 2.

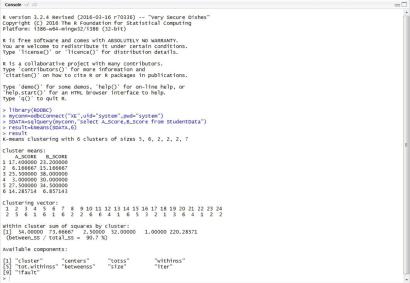


Fig: Shows the Data is run in R programming language.

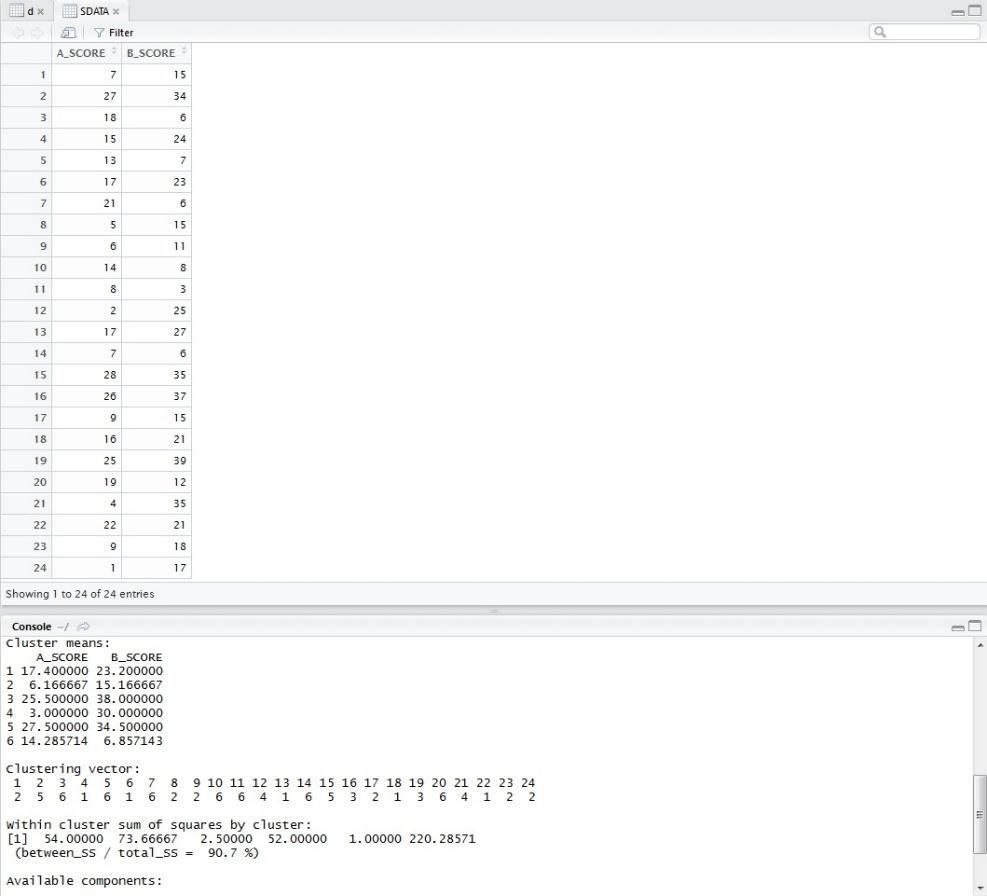


Fig : The end unequaled clusters generated by the R.



***92***

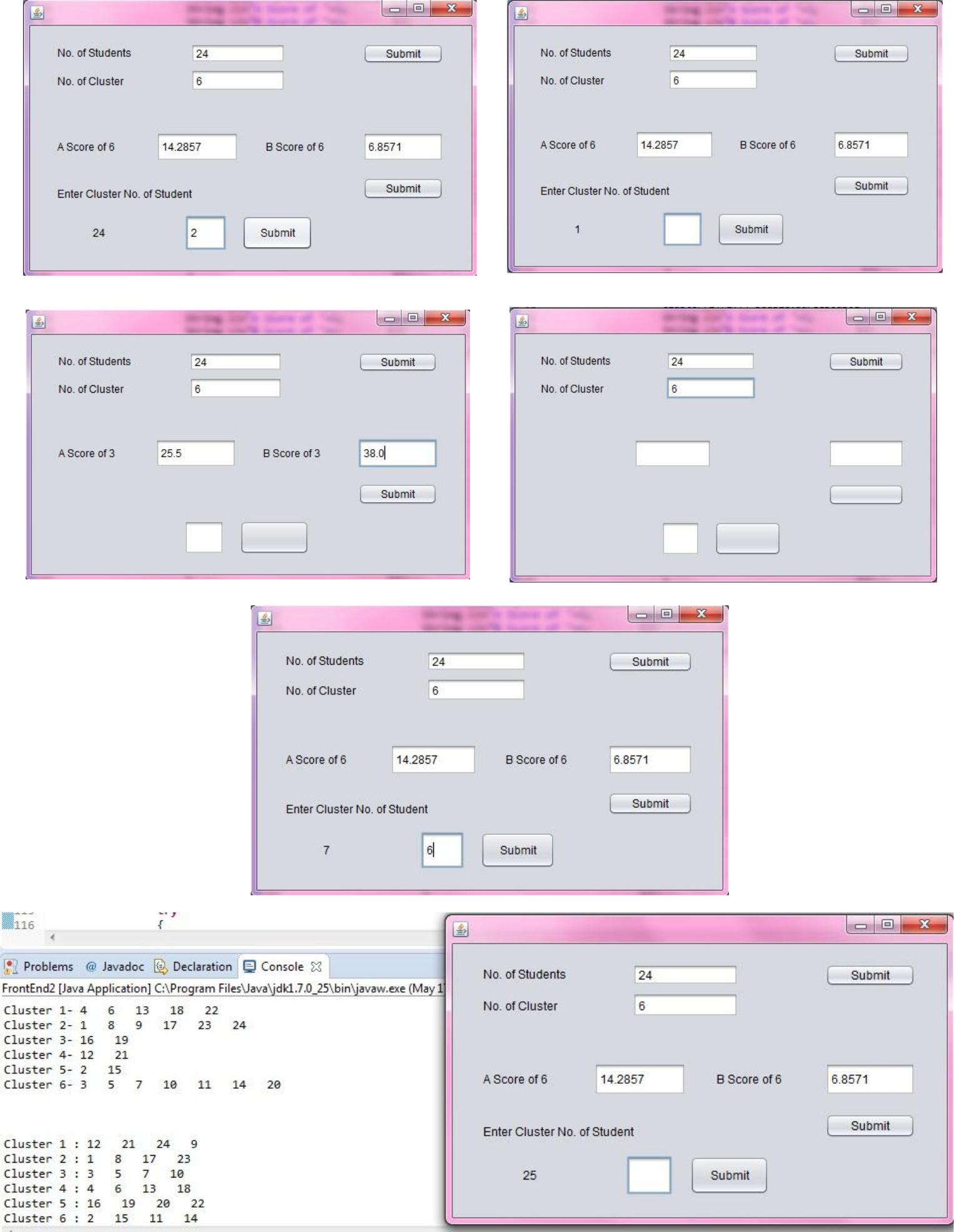
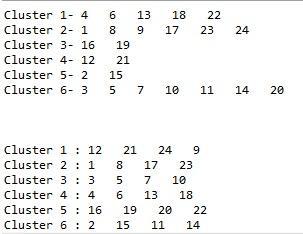


Fig: The Equalization process done through the Front End 2



***93***



Final Equalized Clusters.

**6.1.2. USER DATA SET 1:**

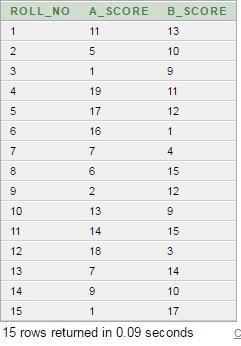


Fig: Table containing Data.



***94***

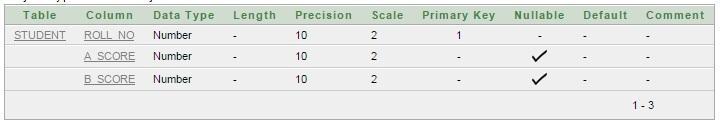
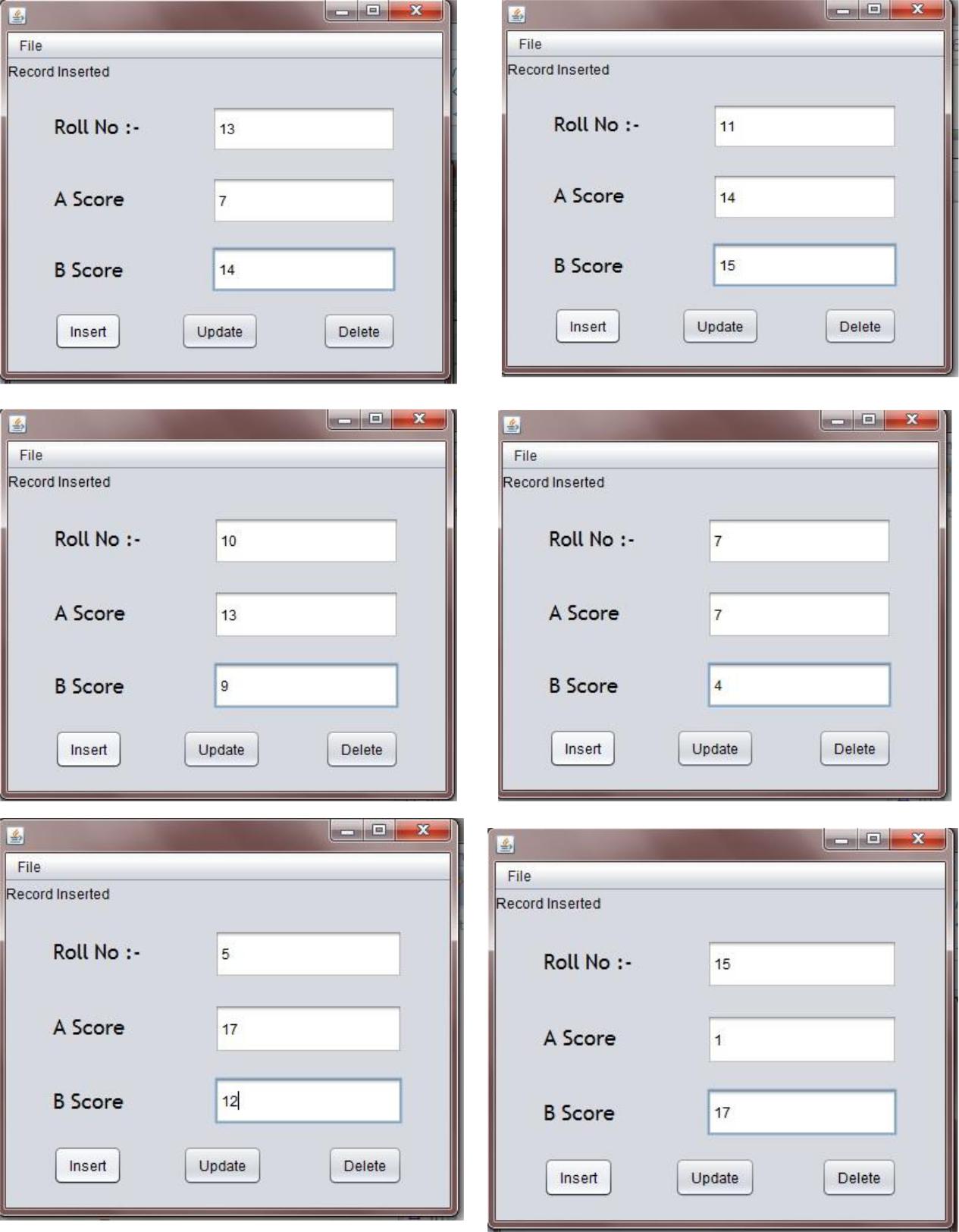


Fig: Description of the Table STUDENT



**Fig: The insertion of Data in the DB vi the Front End 1.**



***95***

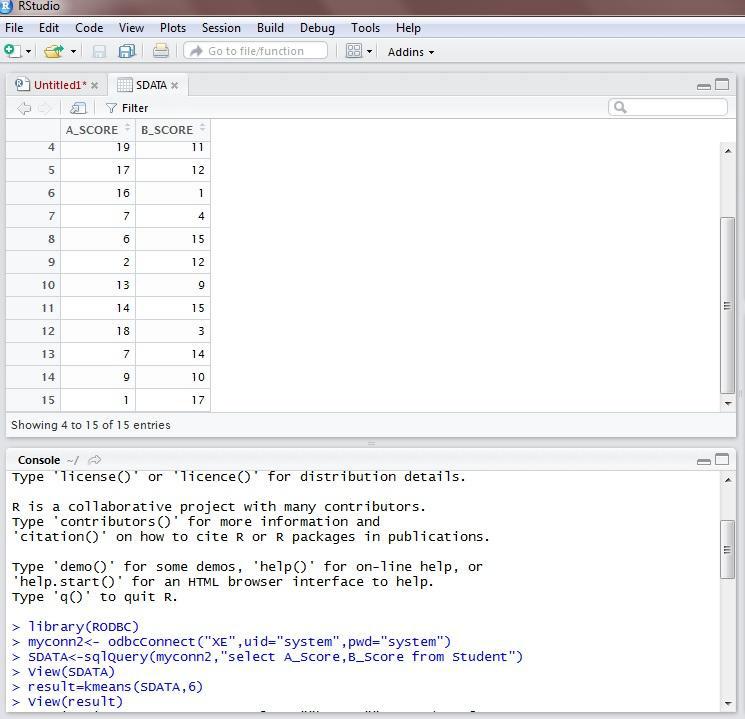


Fig: The running of the script in the R programming language

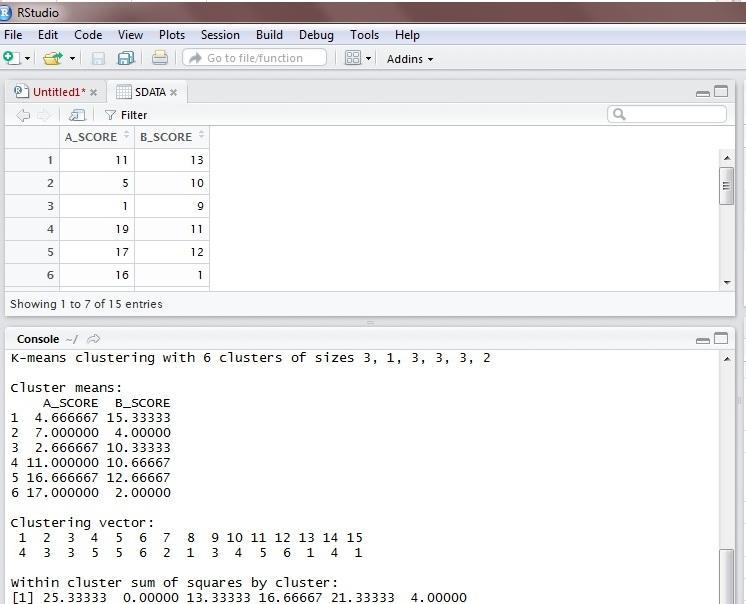


Fig: The Unequalled Cluster formed by the R.



***96***

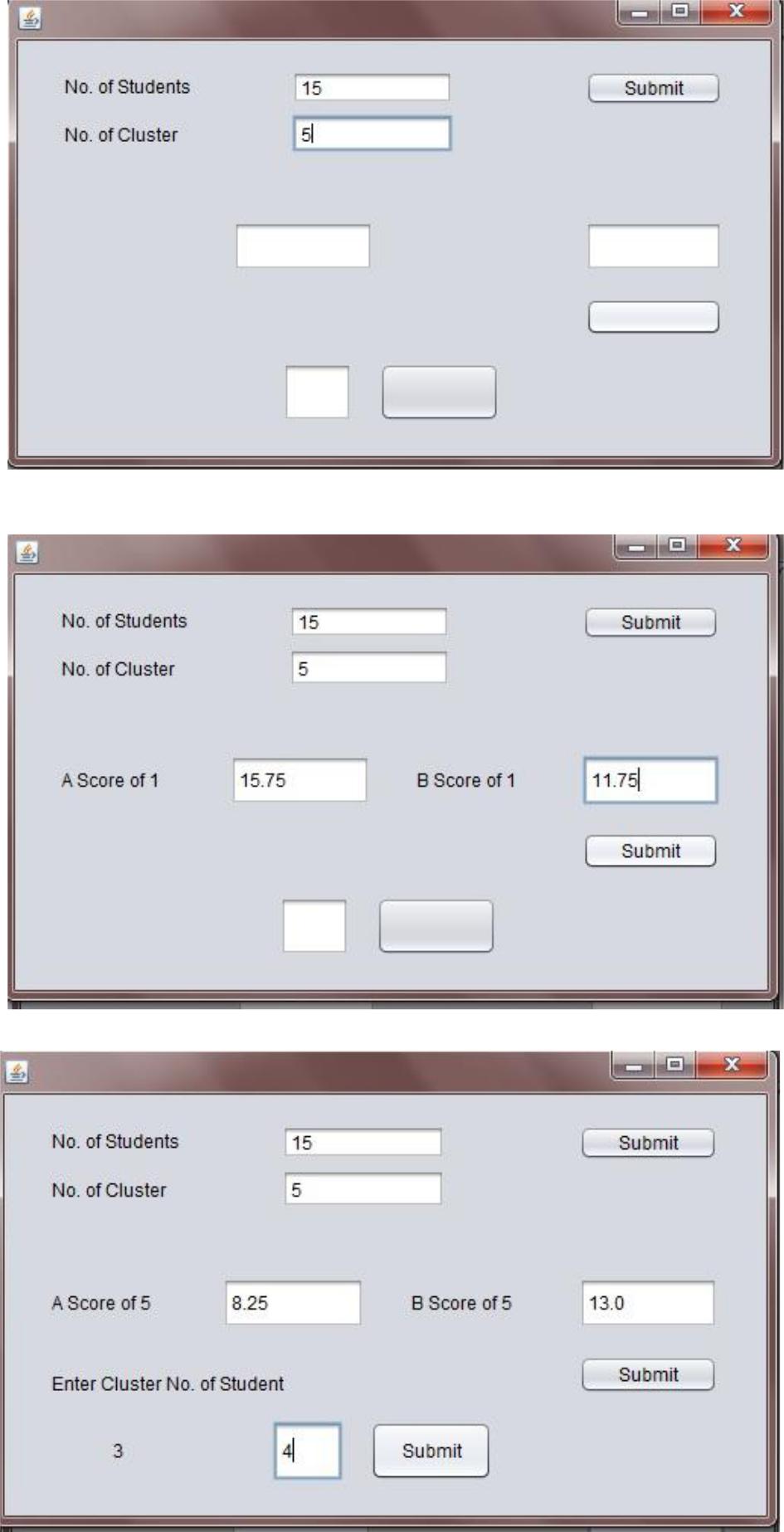


Fig: Tha Equalization Process through the Front End 2, Module 03.



***97***

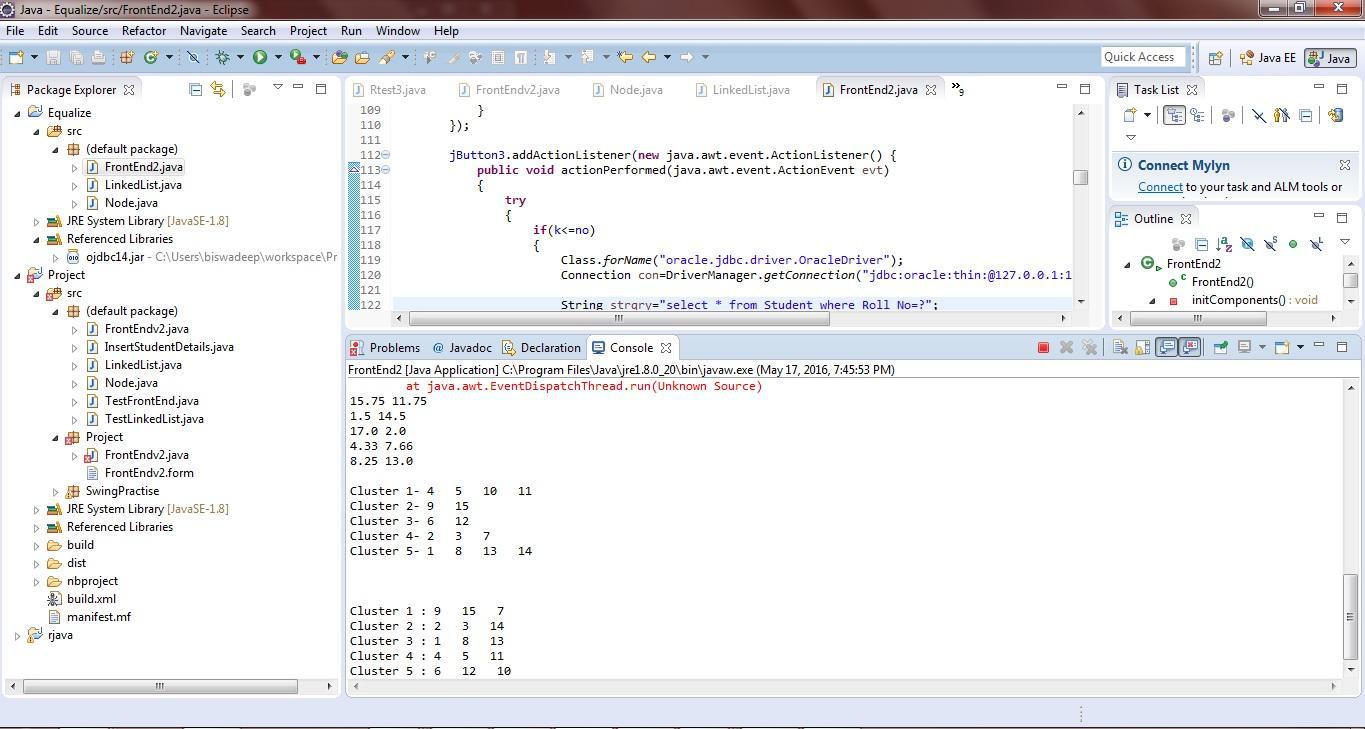


Fig : the end result with equaled clusters.

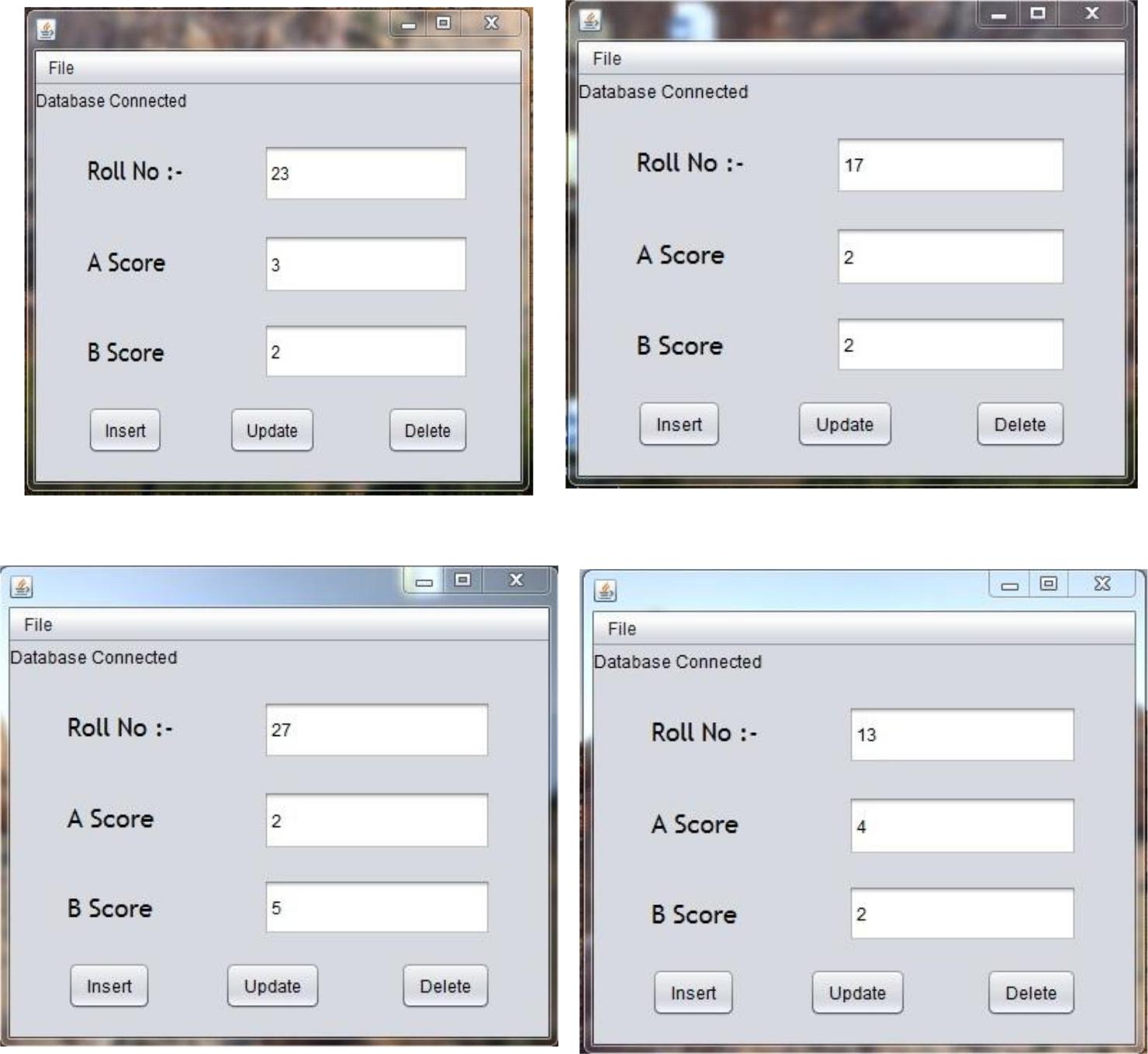
|  |  |  |
| --- | --- | --- |
| **6.1.3.** | **USER SET 02:** | |
|  |  |  |
| **A.value** | **B.value** | **total** |
|  |  |  |
| **3** | **4** | **7** |
|  |  |  |
| **5** | **2** | **7** |
|  |  |  |
| **5** | **4** | **9** |
|  |  |  |
| **4** | **3** | **7** |
|  |  |  |
| **3** | **3** | **6** |
|  |  |  |
| **4** | **1** | **5** |
|  |  |  |
| **4** | **1** | **5** |
|  |  |  |
| **3** | **1** | **4** |
|  |  |  |
| **2** | **0** | **2** |
|  |  |  |
| **3** | **2** | **5** |
|  |  |  |
| **3** | **3** | **6** |
|  |  |  |
| **4** | **3** | **7** |
|  |  |  |
| **4** | **2** | **6** |
|  |  |  |
| **2** | **3** | **5** |
|  |  |  |
| **2** | **3** | **5** |
|  |  |  |
| **2** | **1** | **3** |
|  |  |  |
| **2** | **2** | **4** |
|  |  |  |
| **1** | **1** | **2** |
|  |  |  |
| **2** | **4** | **6** |
|  |  |  |
| **3** | **3** | **6** |
|  |  |  |
| **3** | **4** | **7** |
|  |  |  |
| **3** | **4** | **7** |
|  |  |  |



***98***

|  |  |  |
| --- | --- | --- |
| **3** | **2** | **5** |
|  |  |  |
| **5** | **4** | **9** |
|  |  |  |
| **3** | **5** | **8** |
|  |  |  |
| **4** | **5** | **9** |
|  |  |  |
| **2** | **5** | **7** |
|  |  |  |
| **2** | **2** | **4** |
|  |  |  |
| **3** | **3** | **6** |
|  |  |  |
| **3** | **3** | **6** |
|  |  |  |

**Fig: Table contain the set of data, entered into the DB.**



***99***

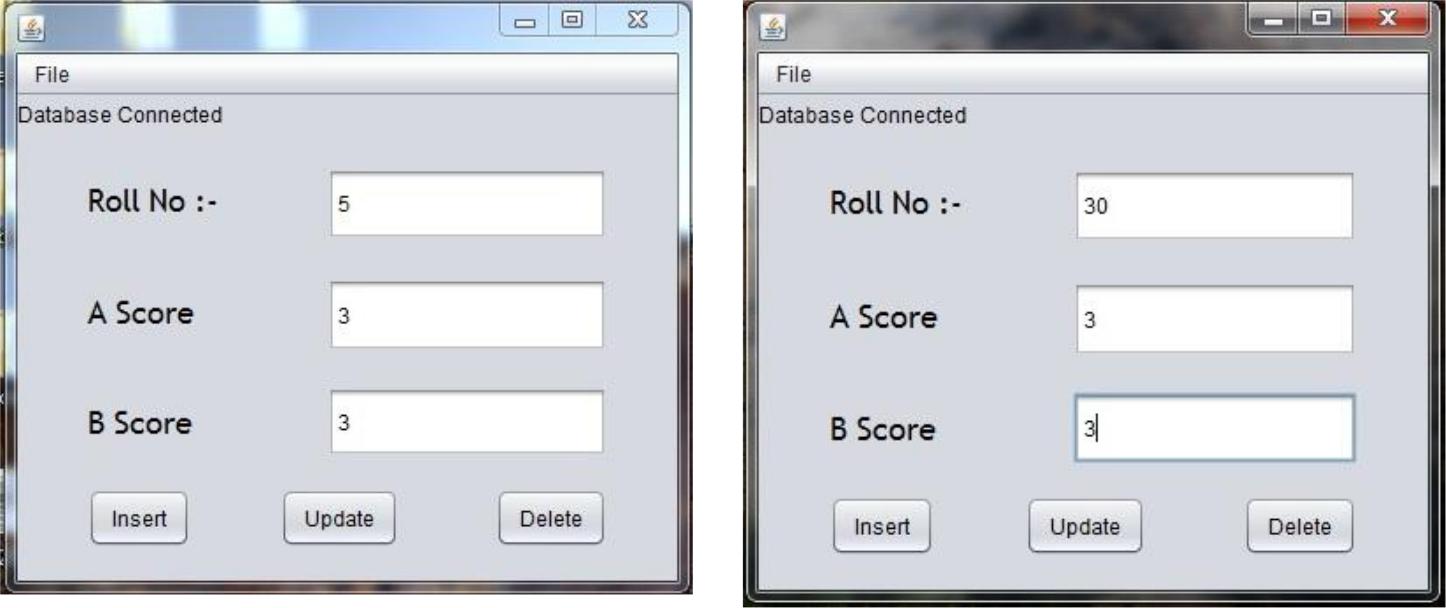


Fig: Insertion of data into DB, Module 01.

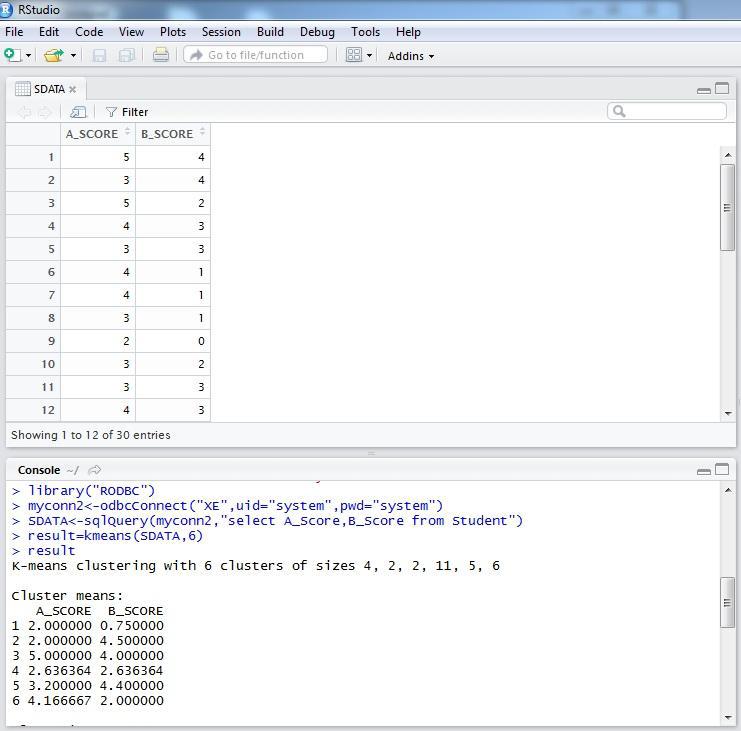


Fig: Processing of data done in R, Module 2.



***10***

***0***

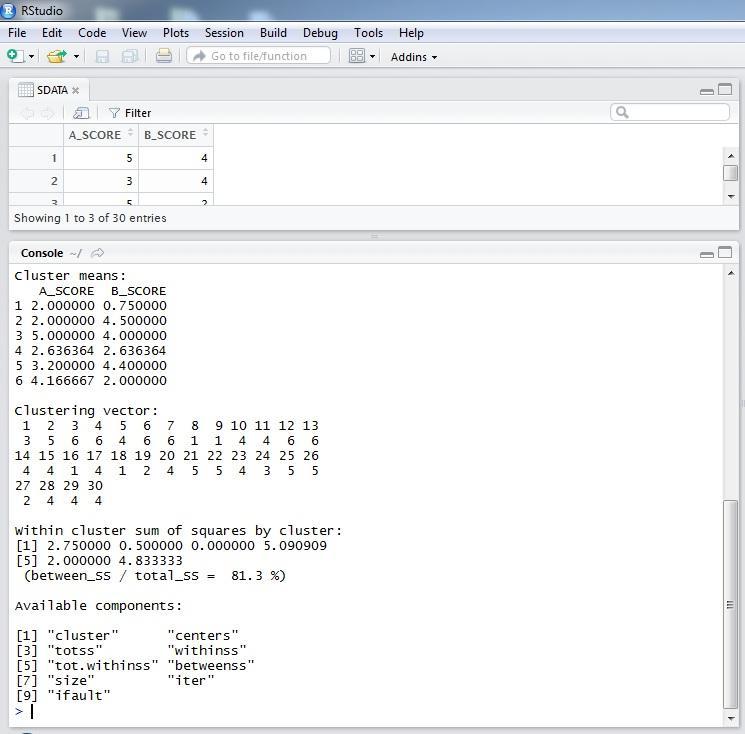


Fig: clusters formed but unequalized.



***10***

***1***

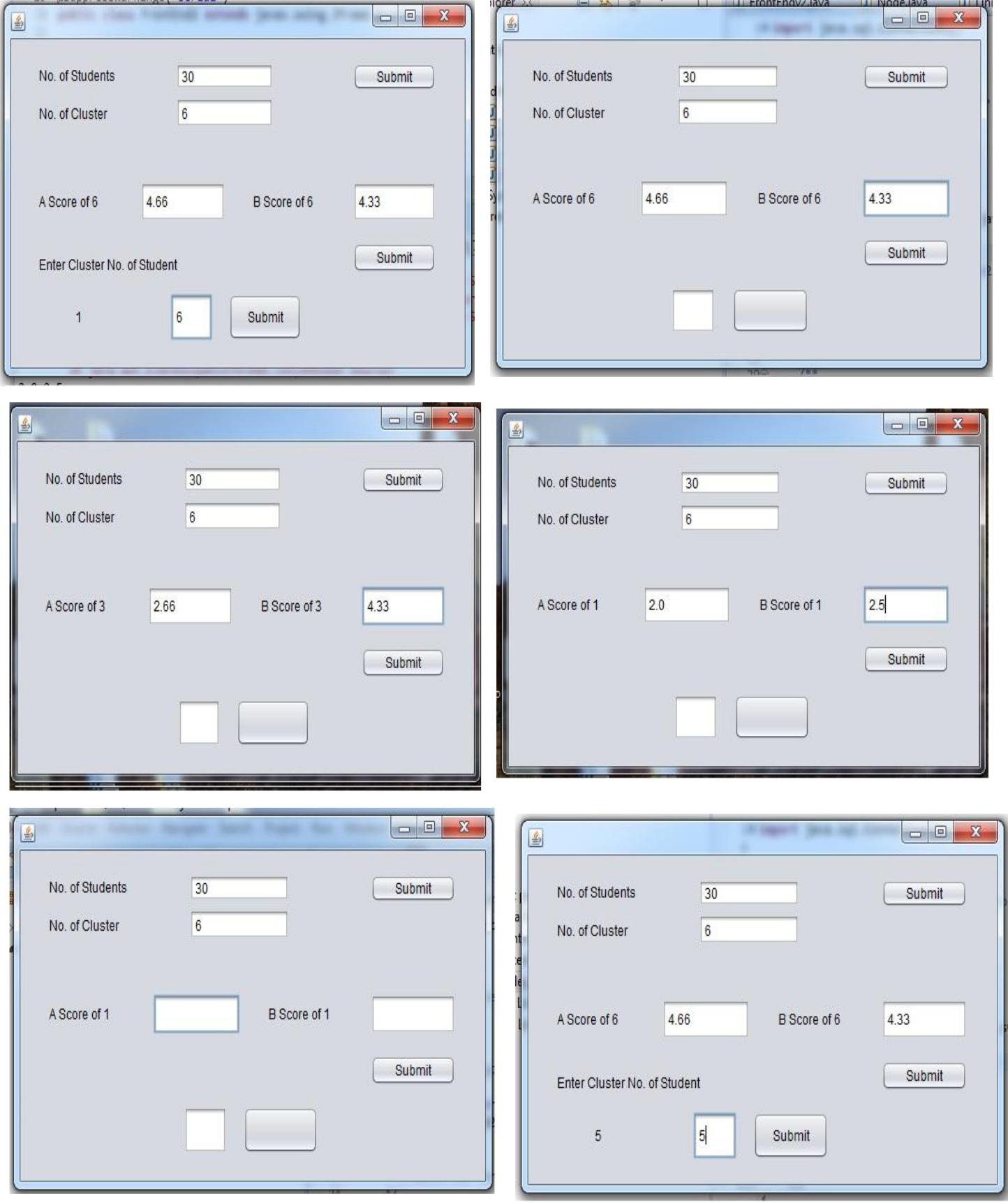


Fig: Thr Front End 2 accepting the cluster details to Equalize



***10***

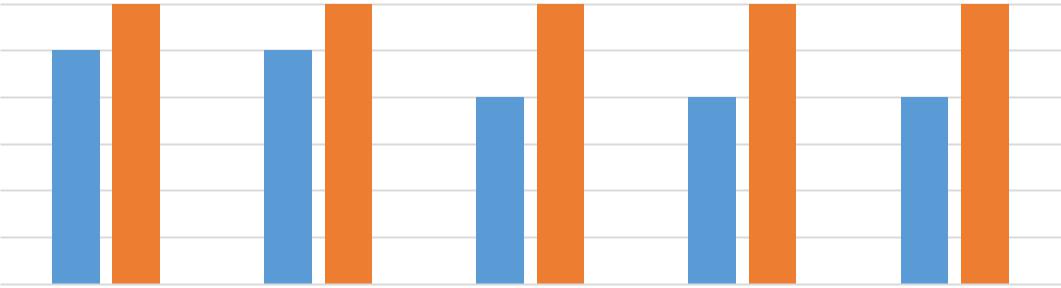
***2***

|  |  |  |
| --- | --- | --- |
| **6.2.** | **DISCUSSION:** | |
|  |  |  |
| Group 1 |  |  |
|  |  |  |
| Serial | X2 | Y2 |
| number | Marks | Marks |
| 14 | 5 | 6 |
|  |  |  |
| 15 | 5 | 6 |
|  |  |  |
| 17 | 4 | 6 |
|  |  |  |
| 28 | 4 | 6 |
|  |  |  |
| 8 | 4 | 6 |
|  |  |  |

Chart Title

7

6



5

4

3

2

1

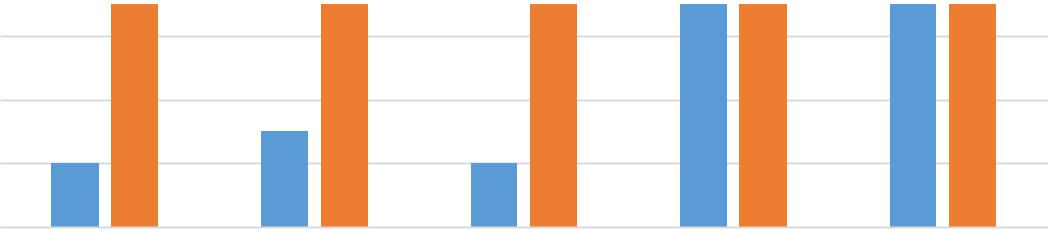
0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 14 | 15 | 17 | | | 28 | 8 |  |
|  |  |  | X2 Marks |  | Y2 Marks |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Group 2 |  |  |
|  |  |  |
| Serial | X2 | Y2 |
| number | Marks | Marks |
| 9 | 2 | 7 |
|  |  |  |
| 16 | 3 | 7 |
|  |  |  |
| 18 | 2 | 7 |
|  |  |  |
| 2 | 7 | 7 |
|  |  |  |
| 27 | 7 | 7 |
|  |  |  |

Chart Title

8



6

4

2

0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 16 | 18 | | | 2 | 27 |  |
|  |  |  | X2 Marks |  | Y2 Marks |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |



***10***

***3***

|  |  |  |
| --- | --- | --- |
| Group 3 |  |  |
|  |  |  |
| Serial | X2 | Y2 |
| number | Marks | Marks |
| 19 | 6 | 8 |
|  |  |  |
| 21 | 7 | 8 |
|  |  |  |
| 22 | 7 | 8 |
|  |  |  |
| 25 | 8 | 8 |
|  |  |  |
| 4 | 7 | 8 |
|  |  |  |

Chart Title

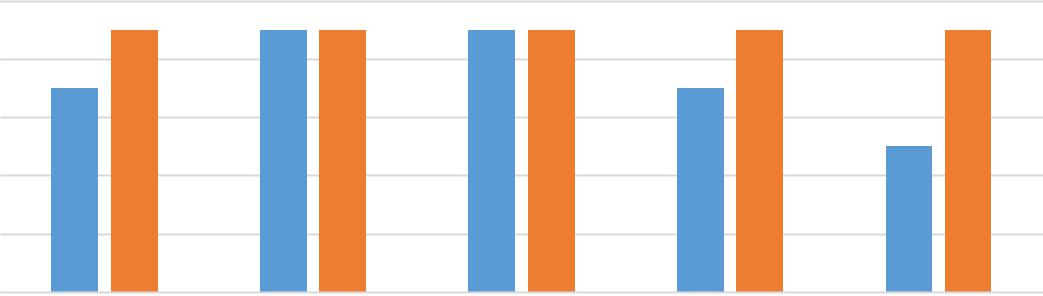
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 |  |  |  |  |
| 8 |  |  |  |  |
| 6 |  |  |  |  |
| 4 |  |  |  |  |
| 2 |  |  |  |  |
| 0 |  |  |  |  |
| 19 | 21 | 22 | 25 | 4 |
|  | X2 Marks |  | Y2 Marks |  |



|  |  |  |
| --- | --- | --- |
| Group 4 |  |  |
|  |  |  |
| Serial | X2 | Y2 |
| number | Marks | Marks |
| 5 | 6 | 8 |
|  |  |  |
| 11 | 6 | 8 |
|  |  |  |
| 20 | 6 | 8 |
|  |  |  |
| 29 | 6 | 8 |
|  |  |  |
| 30 | 6 | 8 |
|  |  |  |

Chart Title

10



8

6

4

2

0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 24 | 26 | | | 3 | 10 |  |
|  |  |  | X2 Marks |  | Y2 Marks |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |



***10***

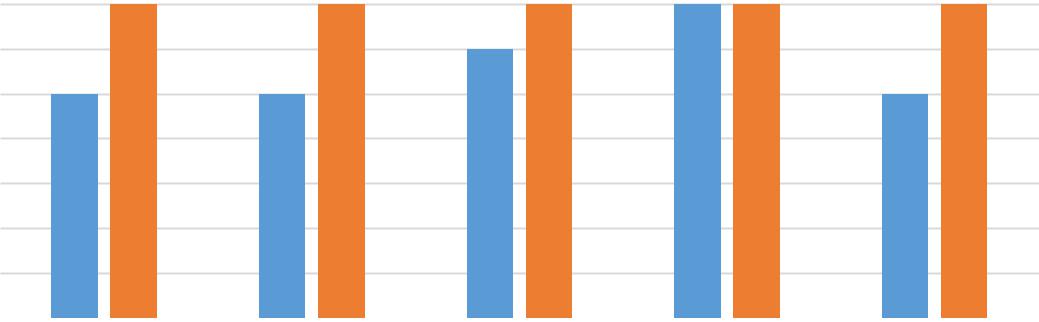
***4***

|  |  |  |
| --- | --- | --- |
| Group 5 |  |  |
|  |  |  |
| Serial | X2 | Y2 |
| number | Marks | Marks |
| 6 | 5 | 7 |
|  |  |  |
| 7 | 5 | 7 |
|  |  |  |
| 13 | 6 | 7 |
|  |  |  |
| 12 | 7 | 7 |
|  |  |  |
| 23 | 5 | 7 |
|  |  |  |

Chart Title

8

7



6

5

4

3

2

1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 6 | | 7 |  |  | 13 | | 12 | 23 |  |
|  |  |  |  | X2 Marks |  |  | Y2 Marks |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Group 6 |  |  |
|  |  |  |
| Serial | X2 | Y2 |
| number | Marks | Marks |
| 1 | 7 | 9 |
|  |  |  |
| 24 | 9 | 9 |
|  |  |  |
| 26 | 9 | 9 |
|  |  |  |
| 3 | 7 | 9 |
|  |  |  |
| 10 | 5 | 9 |
|  |  |  |

Chart Title

10



8

6

4

2

0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 24 | 26 | | | 3 | 10 |  |
|  |  |  | X2 Marks |  | Y2 Marks |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |



***10***

***5***

Figures: Bar graphs of USER DATA SET 2, X vs. Y

The test process was conducted in two ways which is reflected in the USER DATA SET 2, where the students were asked to take test individually and is represented by the marks obtained in X and the later portion they were made to sit in groups and were instructed to take another test of similar standard and its marks is reflected in Y. Then the marks were compared in MS. EXCEL with X, and Y being two bar graph parameters. The resultant report is quite remarkable.

The above records stored in Table format shows that the students gradually increased in performance when worked with their peers rather than when worked alone. The same is represented as in the Bar graphs which almost clear distinction between their previous performance and their performance when done in groups. Thus this shows how effective this system can be, if it’s implemented in real life then then can be quite impactful in the classrooms giving room for pupils to grow their knowledge in an innovative and effective manner. Needless to say this process will also help the students to nurture their own self more efficiently leading to an overall growth in them.

**6.3.USER DOCUMENTATION:**

The user documentation primarily aims to help and guide the use to run the whole system smoothly. To do so as it was mentioned earlier that the whole software process is divided into 4 distinct modules:

**6.3.1.** **Module 1: Insertion of Data into the DB via the Front End 1**

It is important to note that before going to the module 1, and thus the Front End 1 and insert, delete, update data into the DB, a proper test must be conducted among students or test subjects to get parameters on whose basis the data are to be inserted. Like subject marks for students or grades of Degree courses etc etc, with a DB table. The Simple interface is as follows:



***10***

***6***

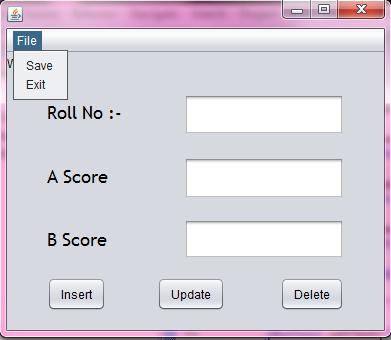


Fig: The front end 1, simply enter Roll no. in roll no field and the two scores in the respective fields.

**6.3.2.** **Module 2: Running the program in R:**

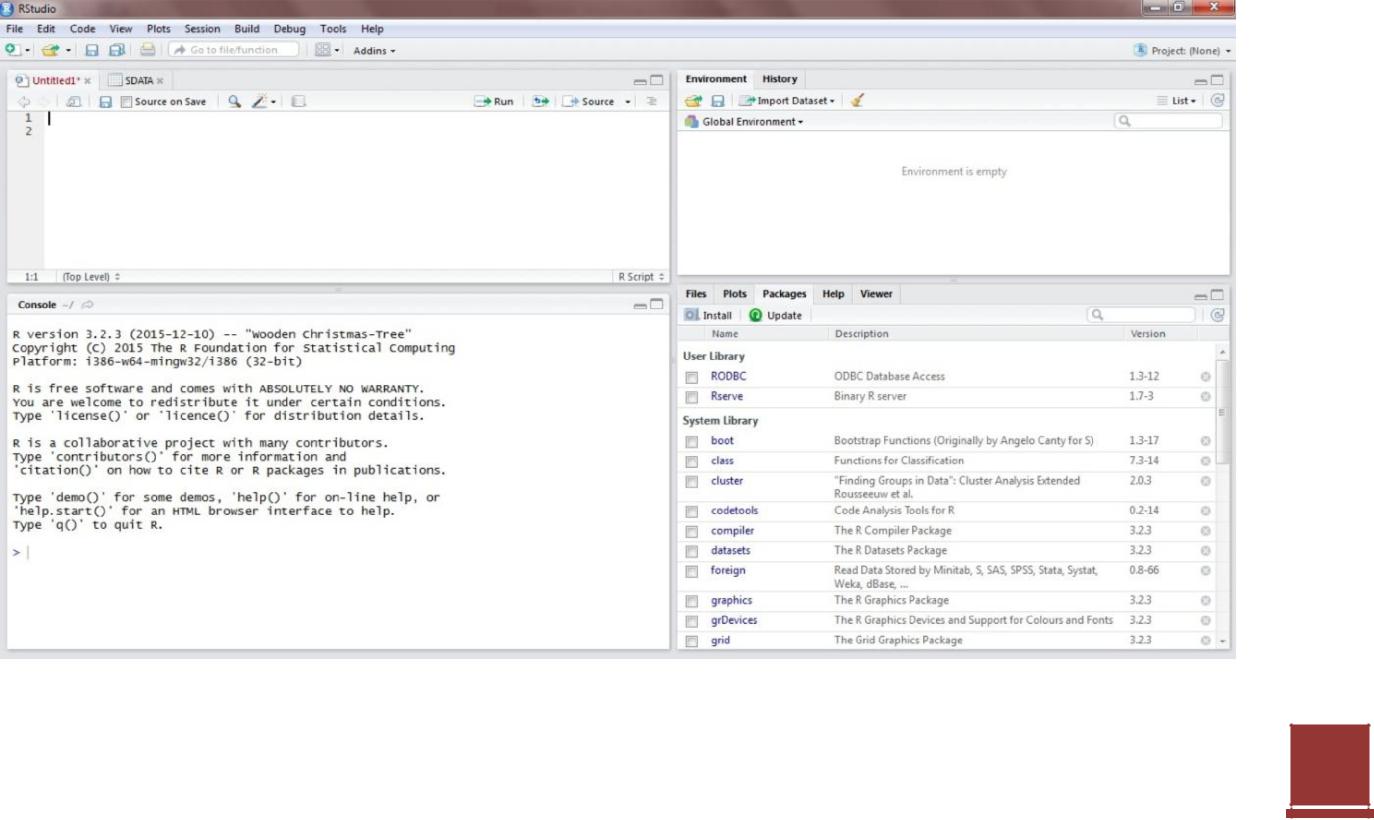
The phase is the one which needs few technical knowledge for the user,

Rstudio software is to be installed and the following command is to be

executed:

library("RODBC")

* myconn2<- odbcConnect("XE",uid="system",pwd="system")
* SDATA<-sqlQuery(myconn2,"select A\_Score,B\_Score from Student")
* result=kmeans(SDATA,6)
* plot(SDATA[c("A\_SCORE","B\_SCORE")],col=result$cluster)
* result

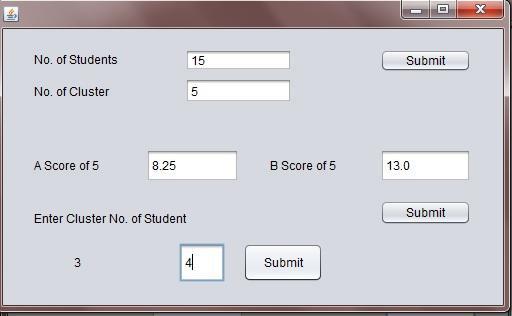


***10***

***7***

**6.3.3. Module 3: Equalization Phase**

The Equalization phase ensures that the all the clusters which are generated by the R, may contain unequal number of objects in them, but this phase ensures that they are equally distributed. The front End 2 is used over here.



Fig; The Front End 2 design the Equalization phase

**6.3.4.** **Module 4: Implementation, the effectiveness:**

The last phase of the project deals with actually implementing the process in real life like a class room. This phase also requires the teacher or the supervisor to conduct a test or to make the student work in groups to bring out the best from them. Then tally the result with the previous result or the Report from phase 1, it is recommended to plot this result in MS. EXCEL.

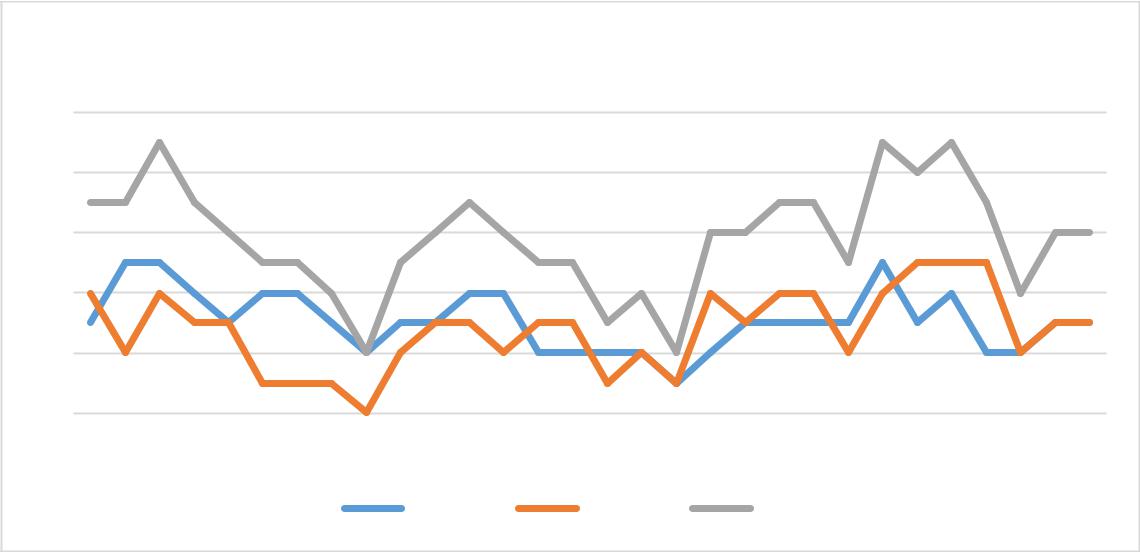


Chart Title

10

8

6

4

2

0

1. 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

A.valueB.valuetotal

Fig : Likewise MS. Excel chart for a comparison.



***10***

***8***

**7. CONCLUSION**

**7.1 Conclusion**

The project concentrates on the fact that how the Groups can be formed among students in a class containing pupils of different academic capabilities. A student can be good in learning new things at the first time when he/ she is made to understand certain concepts. On the other hand another student can have difficulties in learning new concepts as easily as it was in the case of the previous student. In the similar fashion one pupil can be talented in extracurricular activities whereas another is not, the same is reflected in case of the academics. Thus to ensure an overall development and growth of all the students present in the class it is necessary to look into their progress in the academic as well as in other co-academic fields leading to overall personality development.

The differences of such capabilities in students can be observed in a class consisting of normal pupils itself. This project has aimed to find out a solution to such a problem, on doing that small groups among students are planned to be formed. The group formation and there after the Equalization ensure that no similar kinds of student are placed in a group, rather they are formed keeping in mind the principle of heterogeneity. Then the students seated in such groups have abilities unlike to each other, such that one can be a quick learner academically whereas another can be a quick learner in other extra/co-curricular activities or sports. Thus they can help each other where they find their group-mate lack in efforts and can be helped. Therefore giving them a scope for overall development and in their upbringing and personality building.

Ultimately it can be easily said that in this project which is being implemented with the help of 4 module are planned, designed and coded properly, not only that the last module which deal with the implementation of the overall system has also been done with ease, thus proving the success of this project and making it possible to implement in any educational environment through which such result is intended to be achieved. Which eventually leads to betterment of the students making the education environment enriching and joyful while learning at the same time.



***10***

***9***

**7.2 Limitation of the system**

This project has the following limitations to it:

* The User ID and Password for the Oracle Database is fixed for the code. Change in wither the User ID or Password would lead to error and no connection between the Front End Tools and the Database.
* Similarly the table containing all the data is fixed. Change in its attributes or dropping the table would result in no connection with the table.
* ‘R’ though being a very powerful statistical computing software environment, it cannot be easily be implemented using Java Programming. Hence we had to resort to using a separate IDE named

“RStudio” to calculate the initial clusters using k-means algorithm.

* Sometimes the second Front End tools will produce an error-“Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException”. This error does not hinder the follow of the program and can be easily ignored while executing the equalization process.
* The User may face another error where the Front End would produce an SQL Exception- DatabaseNotFoundException. This occurs when the front end tool temporarily loses connection with the SQL database. The user must not panic as waiting it out for a few moments would reconnect the tool to the database or he/she can restarting the ‘OracleServiceXE’ under ‘services.msc’ or restarting the front end would rectify the problem.

**7.3 Future Scope of the project**

* ‘R’ is difficult to be implemented through java but not impossible. In the future we are working and the compatibility of ‘r’ through java programming so that all the modules can be coded through java and java alone.
* “Exception in thread “AWT-EventQueue-0” java.lang.NullPointerException” is being looked into and will be dealt in future versions of the front end tool. Meanwhile, the user however won’t face any problem while in executing the Equalization even after the error occurs.
* Provision of asking user of the User ID, Password and table name through the front end would be looked into.



***11***

***0***

**REFERENCES**

1. Clustering and Sequential Pattern Mining of Online Collaborative Learning Data by Dilhan Perera, Judy Kay, Irena Koprinska, members of IEEE Computer Society. And by Kalina Yacef, and Osmar Zaiane.
2. Cooperative Learning Structures Can Increase Student Achievement Jeanie M. Dotson, Culminating Project , (Kagan Online Magazine, Winter 2001)
3. Group Formation Algorithms in Collaborative Learning Contexts: A Systematic Mapping of the Literature By Wilmax Marreiro Cruz and Seiji Isotani
4. Data Mining and Algorithms by Rajan Chattanvelli.
5. Data Mining Concepts and Techniques by Jiawei Han, Micheline Kamber and Jian Pei.
6. McCarthy's definition of AI (2007) and the intelligent agent paradigm: Russell & Norvig 2003, pp. 27, 32–58, 968–972, Poole, Mackworth & Goebel 1998, pp. 7–21
7. Computers in Human Behavior, Page 23, Page 496-514 by Dewiyanti, Brand-Gruwel, S.Jochems & Broers (2007)

[8]The Benefits of Collaborative Learning in the Elementary Classroom. By Crystal Daugherty Submitted on March 19, 2014 in fulfillment of final requirements for degree St. Catherine University St. Paul, Minnesota

1. Virtual Collaborative Learning paper by Michael J. Sipusic, Robert L. Pannoni, Randall B. Smith, John Dutra, James F. Gibbons, and William R. Sutherland.
2. Data Minig Concepts and Techniques, 3rd Edition, By: Jiawei Han, Micheline Kamber, Jian Pei., Indian Reprint ISBN: 978-93-80931-91-3, Original ISBN: 978-0-12-381479-1
3. Dynamic Group Formation as an Approach to Collaborative Learning Support, [van Srba,](https://www.researchgate.net/profile/Ivan_Srba) [Maria Bielikova](https://www.researchgate.net/profile/Maria_Bielikova)

IEEE Transactions on Learning Technologies (Impact Factor: 1.28). 01/2014; 8(2):1-1. DOI: 10.1109/TLT.2014.2373374



***11***

***1***

https://www.researchgate.net/publication/273296270\_Dynamic\_Group \_Formation\_as\_an\_Approach\_to\_Collaborative\_Learning\_Support, Date: 14/12/2015.

1. Density based Clustering. https://en.wikipedia.org/wiki/File:DBSCAN-Gaussian-data.svg , Date: 15/12/2015.



***11***

***2***

**APPENDIX I**

This section states the process by which the survey was conducted such that to get the initial data for the system and also to get the data after the groups were formed. The second phase made the students sit in groups and let them discuss and answer the question. The instructions which were provides to them are as follows:

Instructions:

1. Please fill the S paper, and keep it with you till the end of this Procedure.
2. Read the X1 paper and answer the questions in the X2 paper (A & B) individually.
3. Write the marks of X2A & X2B, both in S and X2 paper.
4. Submit X1 and X2, wait for the group formation.
5. Sit together with friends in the same group number and Mark group number in S and Y2 paper.
6. Read the Y1 paper and answer the questions in the Y2 paper (A & B) By discussing among your group members.
7. Write the marks of Y2A & Y2B, both in S and Y2 paper.
8. Submit Y1, Y2 and Z paper in the envelope.

The following pages contain the questionnaires and the sheets those were to be filled.

Group Formation in Collaborative Learning

By: Arkit Sen, Sagnik Raychowdhuri, Biswadeep Das

Instructions:

1. Please fill the S paper, and keep it with you till the end of this Procedure.
2. Read the X1 paper and answer the questions in the X2 paper (A & B) individually.
3. Write the marks of X2A & X2B, both in S and X2 paper.
4. Submit X1 and X2, wait for the group formation.
5. Sit together with friends in the same group number and Mark group number in S and Y2 paper.
6. Read the Y1 paper and answer the questions in the Y2 paper (A & B) By discussing among your group members.
7. Write the marks of Y2A & Y2B, both in S and Y2 paper.
8. Submit Y1, Y2 and Z paper in the envelope.

**Name:**

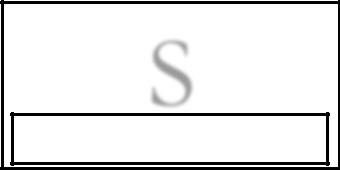
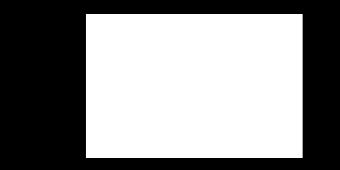
**Roll Number:**

**Signature:**

THANK YOU

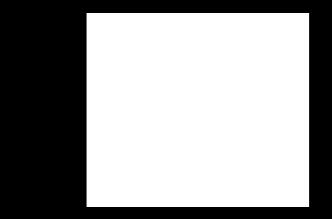
**Reference**

**Section**

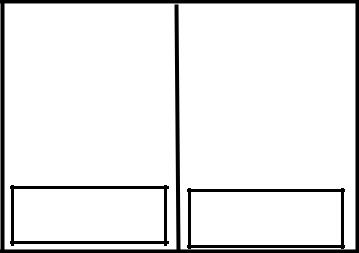


S

**Paper Code**



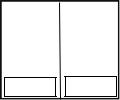
**SERIAL NO.**



**X2-A** **X2-B**

**MARKS X2**

**GROUP**

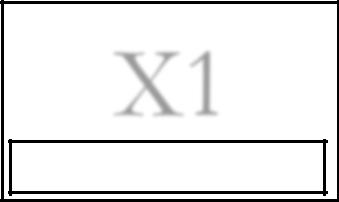


**Y2-A** **Y2-B**

**MARKS Y2**



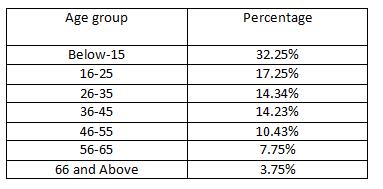
**Group** **aveg.**

X1

**Paper Code**



**Group No.**



**(A).**

1. The device used for measuring altitudes is? [**A.**](javascript:%20void(0)) altimeter

[**B.**](javascript:%20void(0)) ammeter[**C.**](javascript:%20void(0)) audiometer[**D.**](javascript:%20void(0)) galvanometer

1. D.D.T. was invented by?
2. Mosley
3. Rudeolf
4. Karl Benz
5. Dalton
6. The chemical name of Chloroform is? [**A.**](javascript:%20void(0)) Sulphuricacid

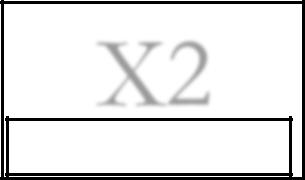
[**B.**](javascript:%20void(0)) SodiumChloride[**C.**](javascript:%20void(0)) SodiumCarbonate[**D.**](javascript:%20void(0)) Trichloromethane

1. The hottest planet in the solar system? [**A.**](javascript:%20void(0)) Earth

[**B.**](javascript:%20void(0)) Venus[**C.**](javascript:%20void(0)) Mars[**D.**](javascript:%20void(0)) Mercury

1. Full abbreviation of “SIM” card: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

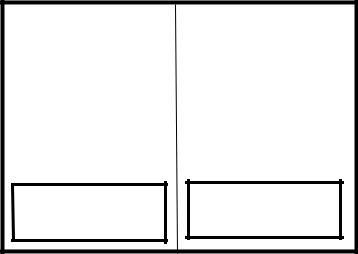
X2

****

**Paper Code**

****

**SERIAL NO.**

****

**X2-A** **X2-B**

**(B).** Refer to the X1 paper to answer the followings

1. Which age group accounts for the second maximum population in the state?

|  |  |
| --- | --- |
| A.) 26-35 | B.) 36-45 |
| C.) 46-55 | D.) 16-25 |
| 2. Out of every 6400, find the number of persons below 25 years? | |
| A.) 3186 | B.) 3168 |
| C.) 3185 | D.) 3178 |

1. If the difference between the numbers of people in the age group 56-65 and 66 above is 12 million. Then find the total population of the state?

A.) 250 Million

B.) 277 Million

C.) 300 Million

D.) 289 Million

1. I f 18 million people in the age group 36-45 and what is the difference between the total number Of people in age group below 15 and 16-25?

A.) 7.4 Million

B.) 8.4 Million

C.) 7.2 Million

D.) 8.2 Million

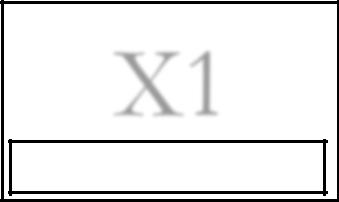
1. There are 400 Million people below 26 years. How many Million people (approximately) are in the 36-45?

A.) 112.48 Million

B.) 114.34 Million

C.) 114.98 Million

D.) 112.98 Million

X1

**Paper Code**



**SERIAL NO.**

**Expenditures of a Company (in Lakh Rupees) per Annum Over the given Years.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Item of Expenditure** | | |  |  |  |
| **Year** |  |  |  |  |  |  |  |
| **Salary** | **Fuel and Transport** | **Bonus** |  | **Interest on Loans** | **Taxes** |  |
|  |  |  |
|  |  |  |  |  |  |  |  |
| 1998 | 288 | 98 | 3.00 |  | 23.4 | 83 |  |
|  |  |  |  |  |  |  |  |
| 1999 | 342 | 112 | 2.52 |  | 32.5 | 108 |  |
|  |  |  |  |  |  |  |  |
| 2000 | 324 | 101 | 3.84 |  | 41.6 | 74 |  |
|  |  |  |  |  |  |  |  |
| 2001 | 336 | 133 | 3.68 |  | 36.4 | 88 |  |
|  |  |  |  |  |  |  |  |
| 2002 | 420 | 142 | 3.96 |  | 49.4 | 98 |  |
|  |  |  |  |  |  |  |  |

**(A).**

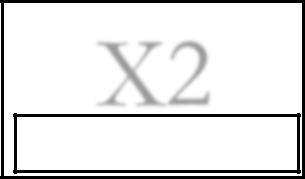
1. Which is the **coldest planet** in the solar system? : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   1. How many **bones are there in human skull**? : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. The **spherical shape of a drop** is due to:
   * 1. Surface tension.
     2. Van der wall force.
     3. Angular momentum.
     4. Viscosity.

4.How many **fundamental duties** are laid in the **constitution of India**?

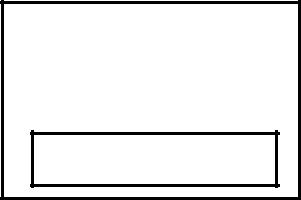
a. 4. b. 9 c. 11 d. 8

5. [**Solid carbon dioxide** is known as?](http://importantquestionsabout.blogspot.in/2012/11/psc-questions-and-answers.html) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

X2



**Paper Code**



**SERIAL NO.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | **X2-A** |  | **X2-B** |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |



**(B).** Refer to the X1 paper to answer the followings

1. What is the average amount of interest per year which the company had to pay during this period?

**A.** Rs. 32.43 lakhs **B.** Rs. 33.72 lakhs

**C.** Rs. 34.18 lakhs **D.** Rs. 36.66 lakhs

1. The total amount of bonus paid by the company during the given period is approximately what percent of the total amount of salary paid during this period?

|  |  |  |  |
| --- | --- | --- | --- |
| [**A.**](javascript:%20void%200;) | 0.1% | [**B.**](javascript:%20void%200;) | 0.5% |
| [**C.**](javascript:%20void%200;) | 1% | [**D.**](javascript:%20void%200;) | 1.25% |

1. Total expenditure on all these items in 1998 was approximately what percent of the total expenditure in 2002?

|  |  |  |  |
| --- | --- | --- | --- |
| [**A.**](javascript:%20void%200;) | 62% | [**B.**](javascript:%20void%200;) | 66% |
| [**C.**](javascript:%20void%200;) | 69% | [**D.**](javascript:%20void%200;) | 71% |

1. The total expenditure of the company over these items during the year 2000 is?

**A.** Rs. 544.44 lakhs **B.** Rs. 501.11 lakhs

**C.** Rs. 446.46 lakhs **D.** Rs. 478.87 lakhs

1. The ratio between the total expenditure on Taxes for all the years and the total expenditure on Fuel and Transport for all the years respectively is approximately?

|  |  |  |  |
| --- | --- | --- | --- |
| [**A.**](javascript:%20void%200;) | 4:7 | [**B.**](javascript:%20void%200;) | 10:13 |
| [**C.**](javascript:%20void%200;) | 15:18 | [**D.**](javascript:%20void%200;) | 5:8 |