

**School of Computer Science & Engineering (SCOPE)**

**Project Topic: Efficient Clustering Techniques in the presence of noise.**

**Course Name : Software Engineering**

**Course Code: CSE0325**

**Slot : L43-43**

**Faculty : Prof Sathyaraj R.**

**Submitted by :**

Purva Singh (14BCE0136)

Kashish Arora (13BCE0144)

Shivin Sinha (14BCE0130)

SOFTWARE ENGINEERING

WINTER SEM 2016-17

Efficient Clustering Techniques in Presence of Noise Project

Software Design Specification

Date Created: 2017.04.1

List of Contributors

| **Name** | **Initials** | **Project Title** | **Email** |
| --- | --- | --- | --- |
| Purva Singh | PS | Efficient Clustering Techniques in presence of noise | purva.singh2014@vit.ac.in |
| Kashish Arora | KA | Efficient Clustering Techniques in presence of noise | Kashish.arora2014@vit.ac.in |
| Shivin Sinha | SS | Efficient Clustering Techniques in the presence of noise | shivin.sinha2014@vit.ac.in |

Change History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Description** |
| 1.0 | 2017.03.09 | Initial Revision |
| 1.1 | 2017.04.10 | Second Revision |
| 1.2 |  | First Final Revision |

Preface

This document presents the Software Design Specification for the Efficient Clustering Techniques in the presence of noise project. The major sections of the document address the system decomposition by module, concurrent process, and data entity. The system dependencies are also described.

Section 2, Decomposition Description, gives a view of the whole system design including concurrent processes and data entities that are common amongst all system modules. The discussion includes a UML Class Diagram that depicts the entire system.

Section 4, Interface Description, goes into detail about the user interface for each module of the System. This is followed by an important discussion of the processes implemented in logic for each module of the system.

Section 5, Detailed Design, extends the design discussion found in Section 2 and describes the design for each system module in more detail. A UML Class diagram is included for each module design discussion. This is followed by a description of the data requirements for each module and the design of those data elements.

# Introduction

## Purpose

Research activity in clustering analysis initially focused on defining different types of algorithms to efficiently group large volumes of data in to clusters. The data to be clusterized is presented in database and in binary files.

To efficiently extract the data from disk and to divide into cluster from memory we have different cluster algorithms like

a) Hierarchical clustering algorithm

b) Partition algorithm

Some of these algorithms use distance as parameter and some other use density as parameter to cluster the data into groups. Later new techniques such as neural network based techniques, fuzzy techniques are developed to improve the clustering process efficiently. However noise is a problem in clustering process. The presence of noise reduces the performance of clustering process. So far no attempt is made to eliminate the noise. As a part of true integration we are presenting enhanced clustering techniques that eliminates problem of noise*.*

## Scope

“Efficient Clustering techniques in the presence of noise” is an application based on .NET framework which also includes user interaction. Our project is going to provide communication environment for users (testers, doctors). The GUI layer will be responsible for interaction with user and various calls to different graphical and visualization tools. This module also provides an interface to display performance results and comparison reports showing the performance of K-means, fuzzy c-means, K MEDIODS and enhanced C-means algorithms. The database server layer will use database server for maintaining transactional data items for use in K-MEDIODS clustering. As of now, we will be using ORACLE as data server.

In the end, the user will be able to compare the three different algorithms by seeing its time and computational complexity, by comparing the graph that displays clusters using fuzzy c means, k-means, K-MEDIODS and enhanced algorithms. In high level details, the system will use PHP for server side management, MYSQL DBMS to store and manipulate data and GUI to interact with users.

It is within the scope of this document to make user familiar to the the facilities available to them through this system developed in this project. The facilities include, generation of clusters from dataset by using K-Means, K-Mediods, enhanced fuzzy means and fuzzy C means, displaying time and space complexity of the mentioned algorithms, to visualize the generated line chart, bar chart and the table format displaying the time and space complexity analysis for different algorithms.

## Definitions and Acronyms

Table of Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Definition, Acronym, or Abbreviation** | **Description** |
| SDS | Software Design Specification. |

## References

Table of References

|  |  |
| --- | --- |
| **References** | **Description** |
| Software Development Plan | The Software Development Plan from the Efficient Clustering techniques in the presence of noise project was referenced. |
| Software Requirements Specification | The Software Requirements Specification from the Efficient Clustering techniques in the presence of noise project was referenced. |

# Decomposition Description

## Module Decomposition

The Efficient clustering techniques in the presence of noise project has been decomposed into the following modules.

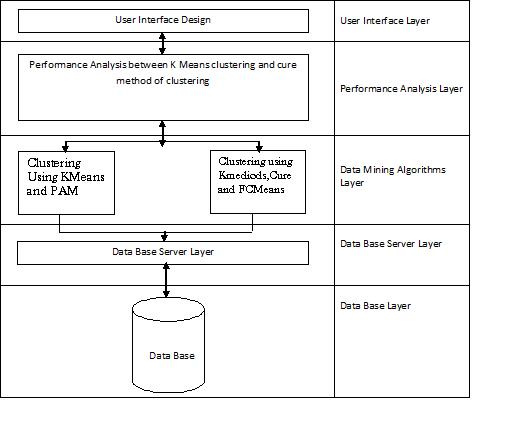
## 1) User interface layer

## 2) Performance analysis layer

## 3) Data mining algorithms implementation layer

## 4) Data base server layer

## 5) Data base layer



1.) **User Interface Layer**:

This layer is responsible for interaction with user and various calls to different graphical and visualization utilities. This module provides an interface for user to invoke with system and to execute queries based clustering algorithms. This module also provides an interface for user to display the performance results and comparison reports showing performance of Kmeans, FCM and enhanced FCM algorithms. The following are different services that this module offers.

* To design an interface for FCM clustering
* To design an interface for K-Medoids clustering
* To generate graph that displays clusters using enhanced FCM algorithm
* To generate graph that displays clusters using KMedoids algorithm
* Computing time and space required to generate clusters.
* Generating graph showing clusters of KMedoids algorithm
* Generating graph showing clusters of FCM algorithm

2.)**Performance analysis layer:**

This layer presents the computational complexity and time complexity for generating clusters using Kmeans algorithm, FCM and enhanced FCM algorithm. The computational and time complexities are generated for different sizes of data.

3.) **Data mining algorithm implementation:**

This layer is main layer that connects all system components together. This layer is divided into sub modules.

1. Clustering using K Means
2. Clustering using K MEDOIDS
3. Clustering using enhanced FCM
4. Clustering using CURE.

4.) **Database server layer:**

This layer uses a database server for maintaining transactional dataitems for use in K MEDOIDS clustering, here we are using ORACLE 10G as database server.

5.) **Database layer:**

This layer is collection of database tables that maintains transactional data items. We can also datasets for maintaining transactional data items. For implementing this system we are maintaining student’s semester marks data

## Data Decomposition

The following are the three major data components, the User Registration Information, the main data set and Comparison Information.

**User Registration Information**: This is a database that contains the following data items;

* id: The id for a user.
* username: The username of user .
* password: The password for user's account

**The main data set**: This is a database that contains the data that the user wants to be clustered. For ex. The doctors can feed in a data set of scanned brain images for clustering due to the presence of noise in them, faculty members can feed in the student’s semester wise marks to cluster it into groups.

**Comparison Information:**

This table contains the following data items:

* clustering technique’s name
* clustering technique’s time complexity (time taken to perform clustering)
* Clustering technique’s space complexity (space taken to perform clustering).

# Dependency Description

## Inter-module Dependencies

### Independent Modules

The following modules are independent and do not rely on any other modules to initiate them or to provide data.

* User Login Interface Module.

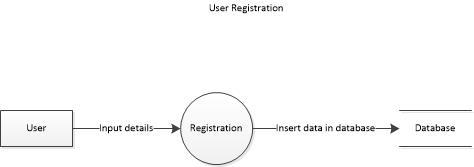
### Dependent Modules

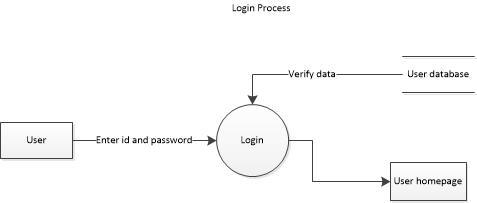
The following modules are dependent on one another for their functioning.

* Performance Analysis Module: This module is executed only when the user has entered the details asked in the User Interface module properly.

## Data Dependencies

The following Data Flow Diagram shows the data dependencies between the various entities and modules.





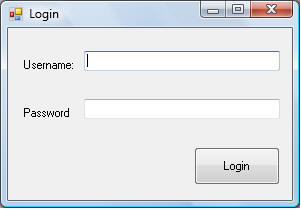
# Interface Description

## Module Interface

### User Registration Module Description

#### User Interface Design

**REGISTRATION PROCESS:**

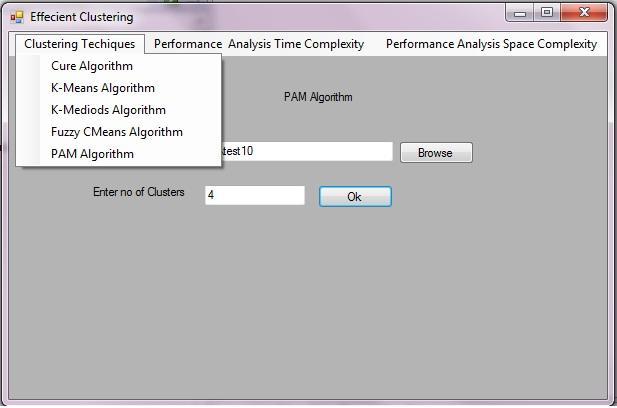


**Description:**

If the user enters “admin” as the username, then he will be directed to an admin page.

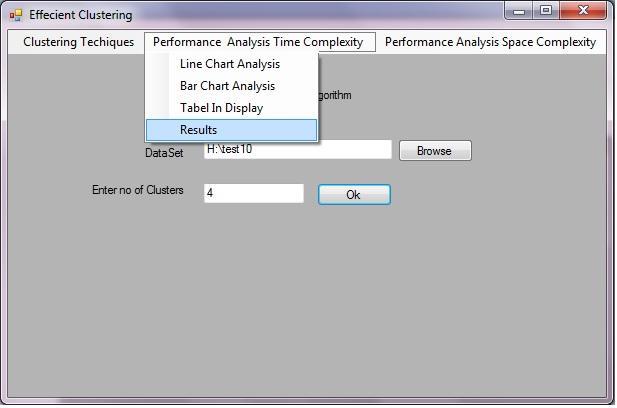
If the user enters “staff” as the username, then he will be directed to the staff page.

**THE STAFF PAGE:**

****

**Description:** In this Screen, we will select the different clustering techniques of the data set and the Number of clusters to be formed**.**

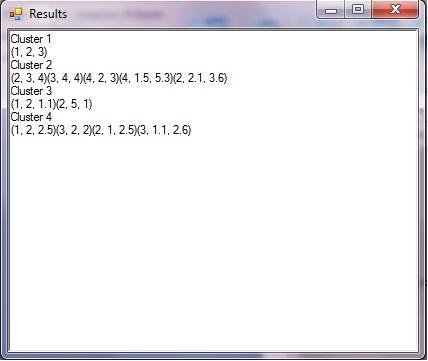
**TO CHOOSE PERFORMACE ANALYSIS ACCORDING TO TIME COMPLEXITY:**

****

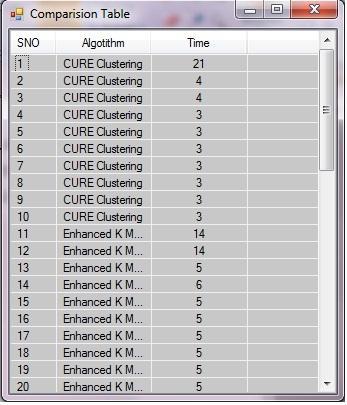
**Description:** In this page, to choose the performance analysis according to time complexity for the different clustering techniques for the number of clusters.

**RESULTS:**

**Description:** In this the noise data can be divided into user defined number of clusters**.**

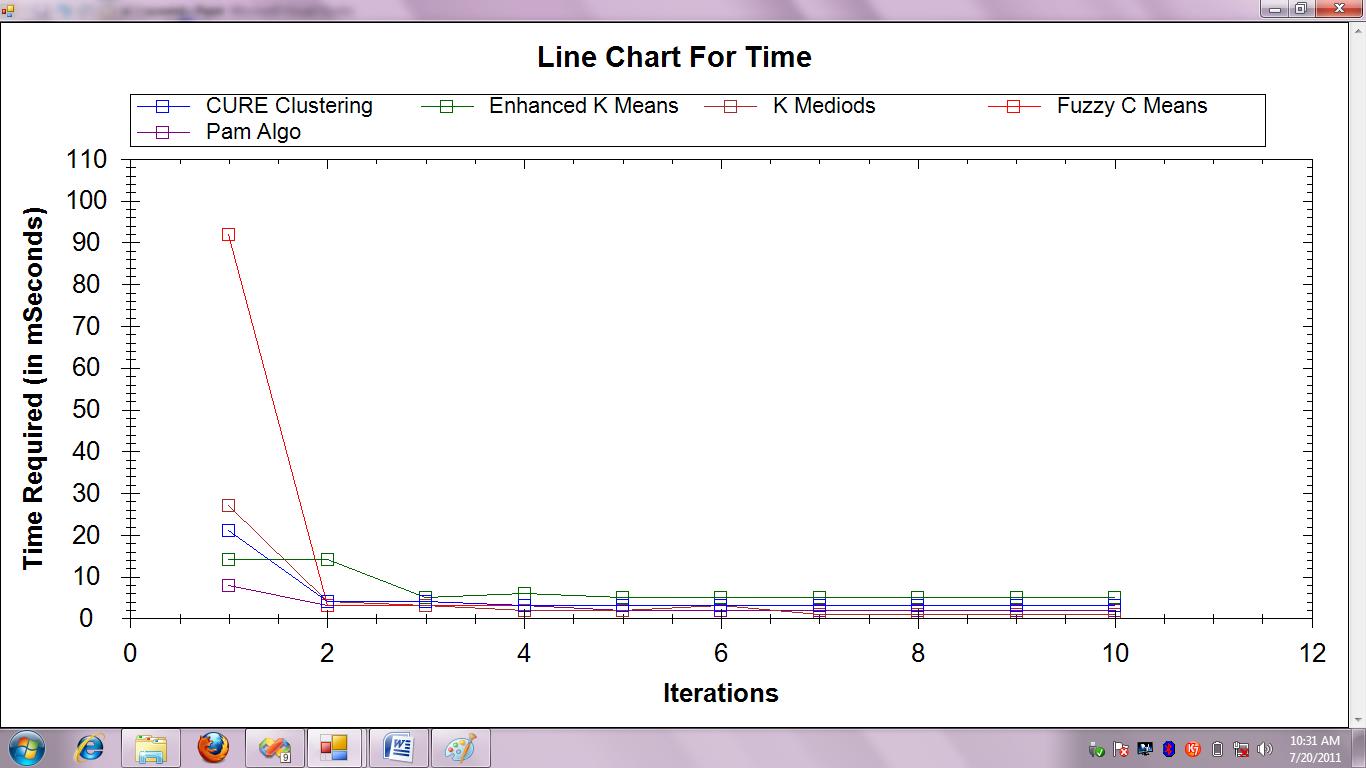
****

**TABLE DISPLAY FOR TIME COMPLEXITY:**

****

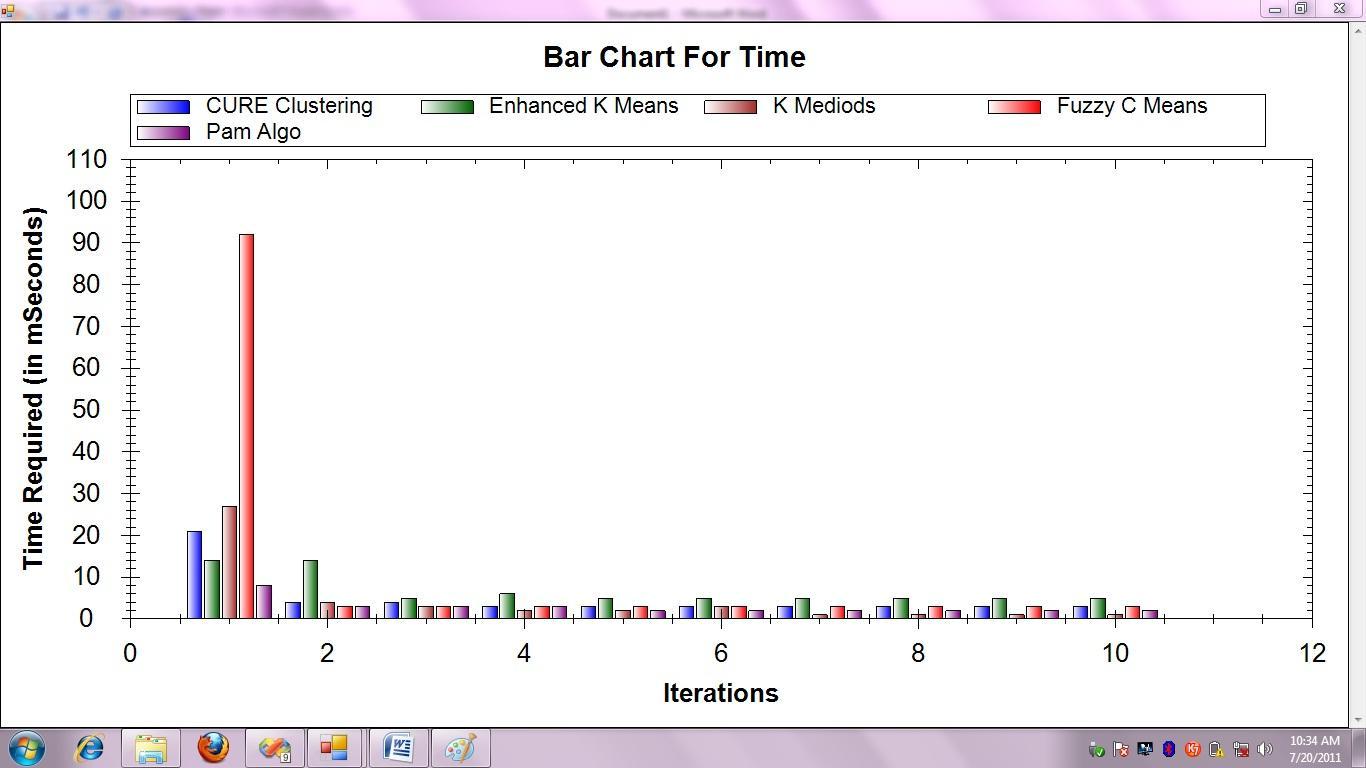
**Description***:* In this Comparison table, the time taken by the algorithm for generating clustering for the noise data. In this the time taken by the generating different clusters is first high and then reduced in to uniform time.

**LINE CHART FOR TIME COMPLEXITY**:



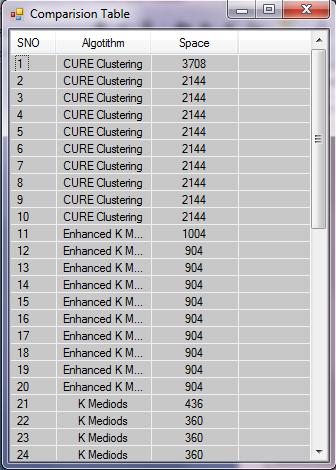
**Description***:* In this Graph, the time complexity of the different clustering algorithms can be plotted in the time required (in mseconds) and iterations in the Line chart format.

## BAR CHART FOR TIME COMPLEXITY:

****

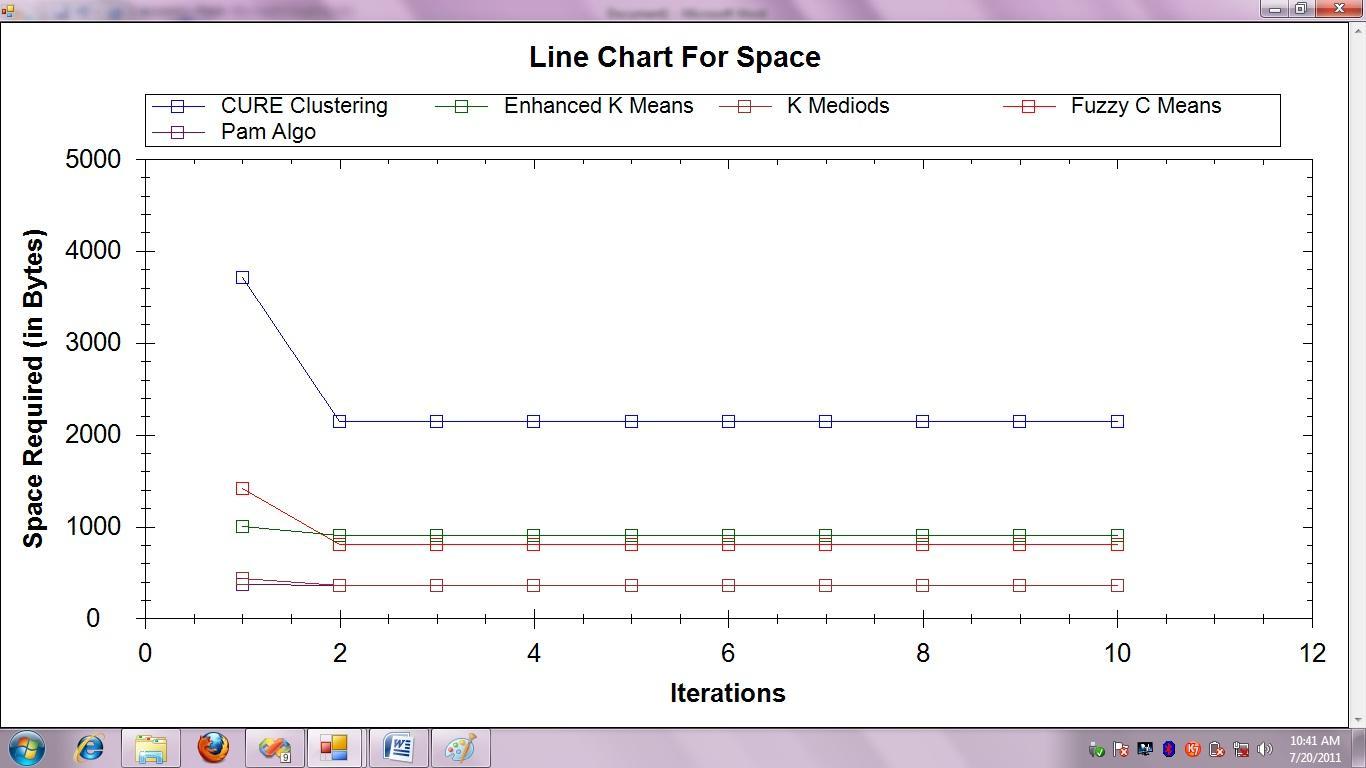
**Description**: In this Graph, the time complexity of the different clustering algorithms can be plotted in the time required (in mseconds) and iterations in the Barchart Graph format.

**TABLE DISPLAY FOR SPACE COMPLEXITY**:



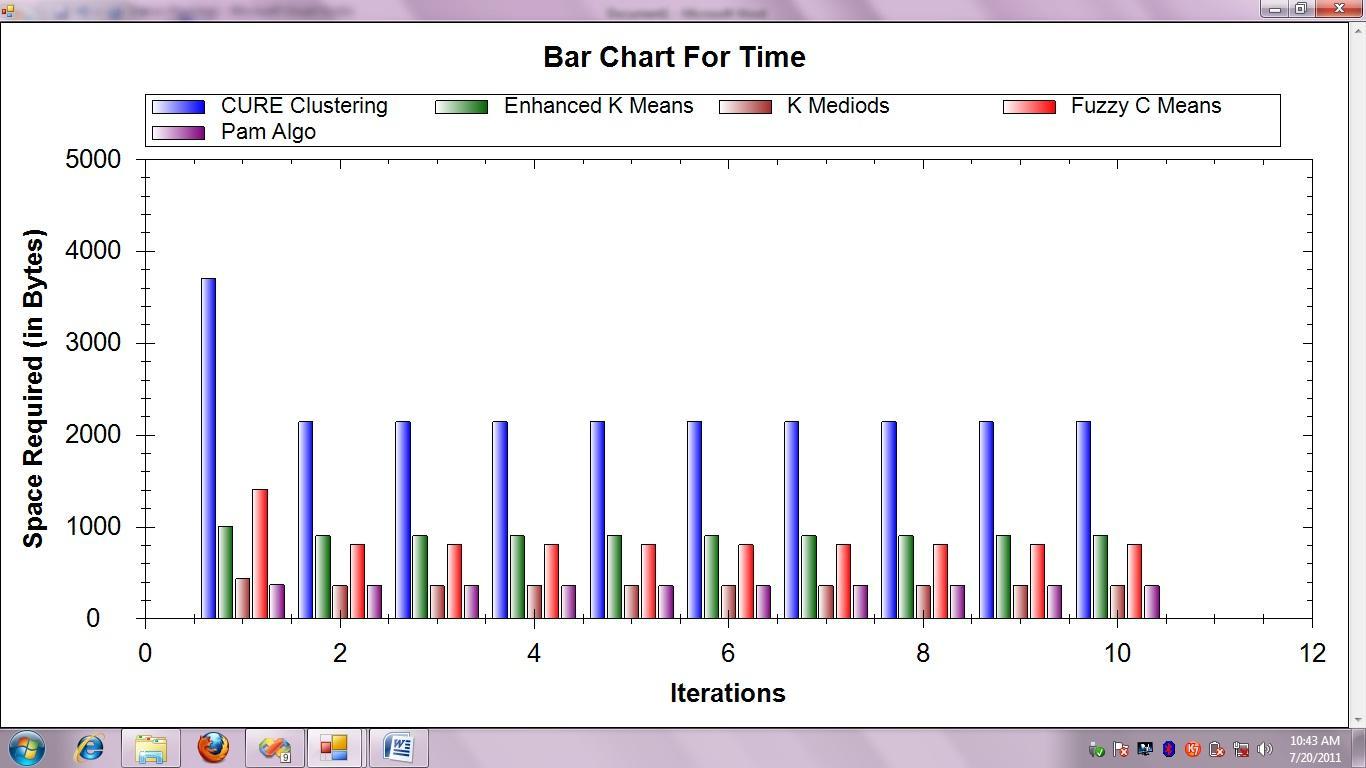
**Description**: In this Comparison table, the time taken by the algorithm for generating clustering for the noise data. In this the space taken by the generating different clusters is first high and then reduced in to uniform space.

**LINE CHART FOR SPACE COMPLEXITY**:



## Description: In this Graph, the time complexity of the different clustering algorithms can be plotted in the time required (in mseconds) and iterations in the Line chart format.

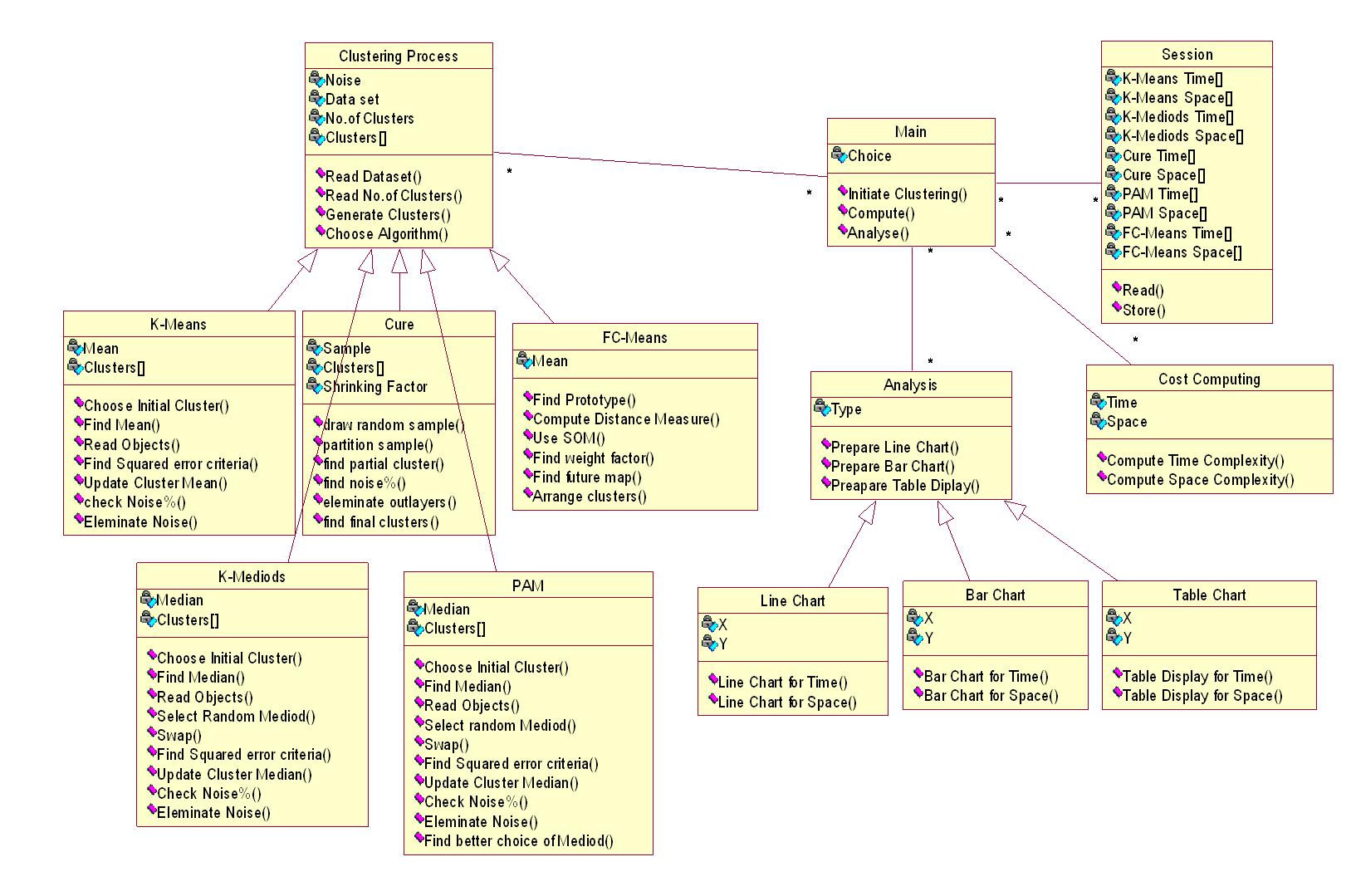
**BAR CHART FOR SPACE COMPLEXITY**:



**Description**: In this Graph, the time complexity of the different clustering algorithms can be plotted in the time required (in mseconds) and iterations in the Barchart Graph format.

#### 4. Design Description

The main menu gives the user the options of choosing the type of clustering they want to go for. After selecting the option, the Main menu initiates the clustering process, computes the time and space complexity of the technique and then analyzes it using the bar and line chart as well as in a tabular format.

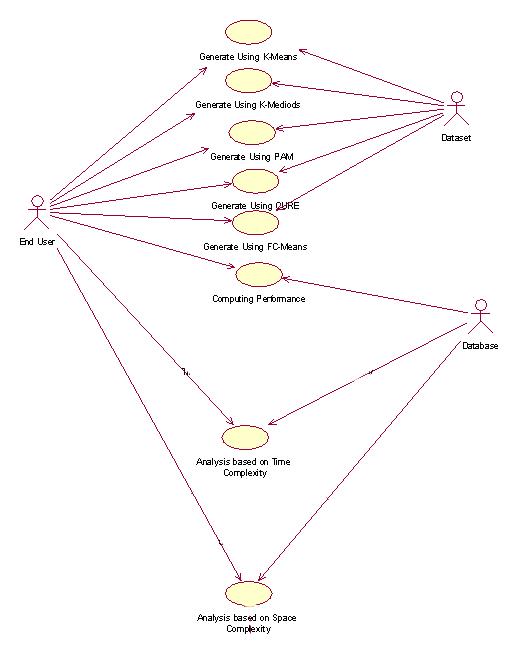


**5.) Use Case Diagram:**

**In our system we identify the following use cases/**

1. Generate using K-Means
2. Generate using K-Mediods
3. Generate using PAM
4. Generate using CURE
5. Generate using FCMeans
6. Computing Performance
   1. Compute Time Complexity
   2. Compute Space Complexity
7. Analysis based on Time Complexity
   1. Using Bar Chart
   2. Using Line Chart
   3. Using Table Display
8. Analysis based on Space Complexity
   1. Using Bar Chart
   2. Using Line Chart
   3. Using Table Display

**5.1 Usecase Diagram:**

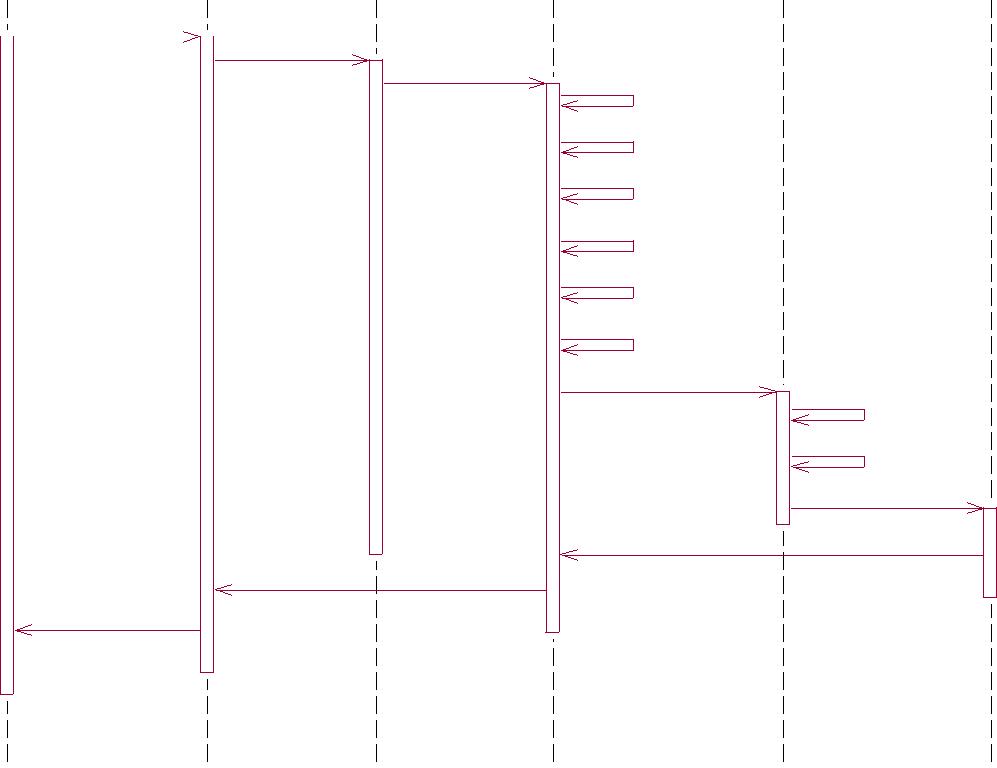
****

**6. Sequence Diagram**

**Sequence diagram:**

K-MEANS CLUSTERING:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| : End User | | | |  | Main | |  | Cluster |  | K Means |  | Cost |  | Session |  |
|  |  |  |  | Process |  |  |  | Computing |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 1.Start | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | 2.Initiate Cluster | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |



3.Use K-Means

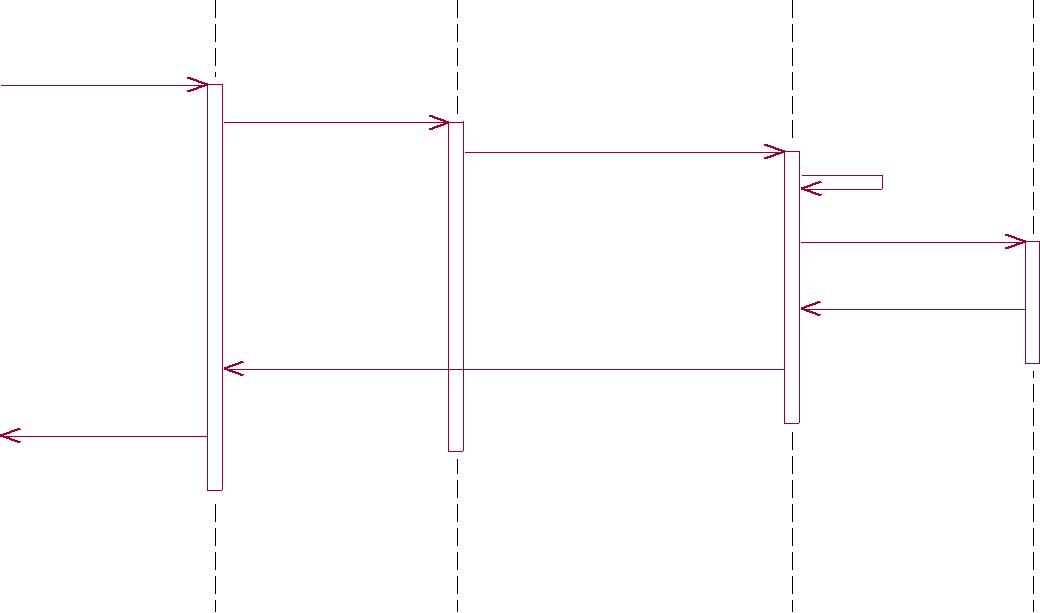
4.Choose Initial Cluster

5.Find Mean 6.Read Objects

7.Update Cluster Mean

LINE CHART:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Main |  | Session |  | Analysis |  | Line Chart |
|  |  |  |  |  |  |  |



1.Show Analysis

2.Read

3.Read Time of All Algorithms

4.Plot

5.Draw Line Chart

6.Chart Created

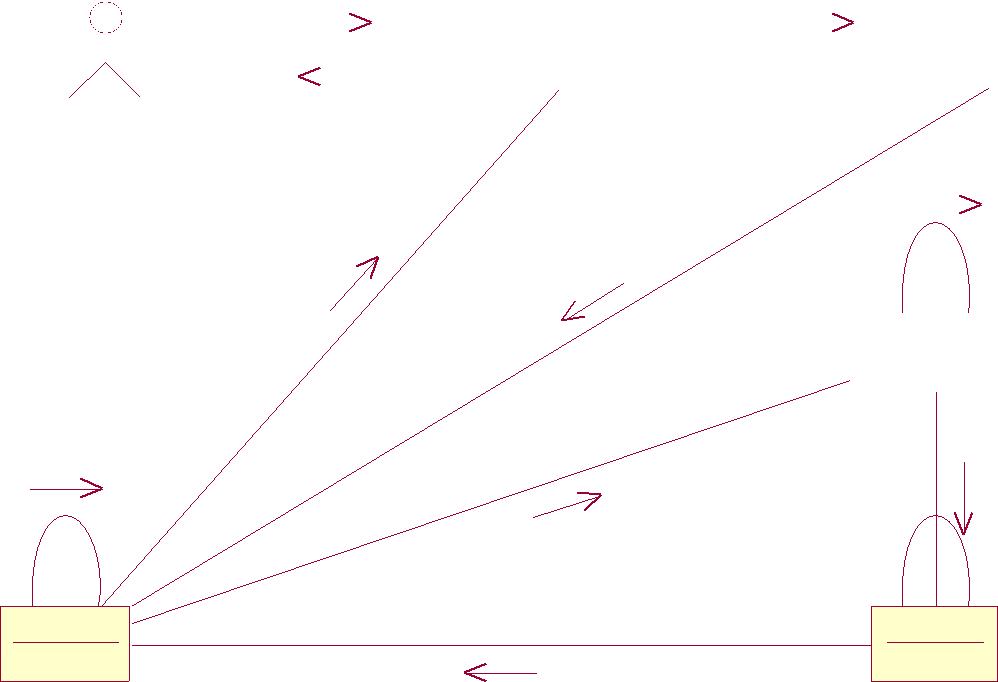
7.Display Line Chart

8.View Line Chart

COLLABORATION DIAGRAM:

K-MEANS CLUSTERING:

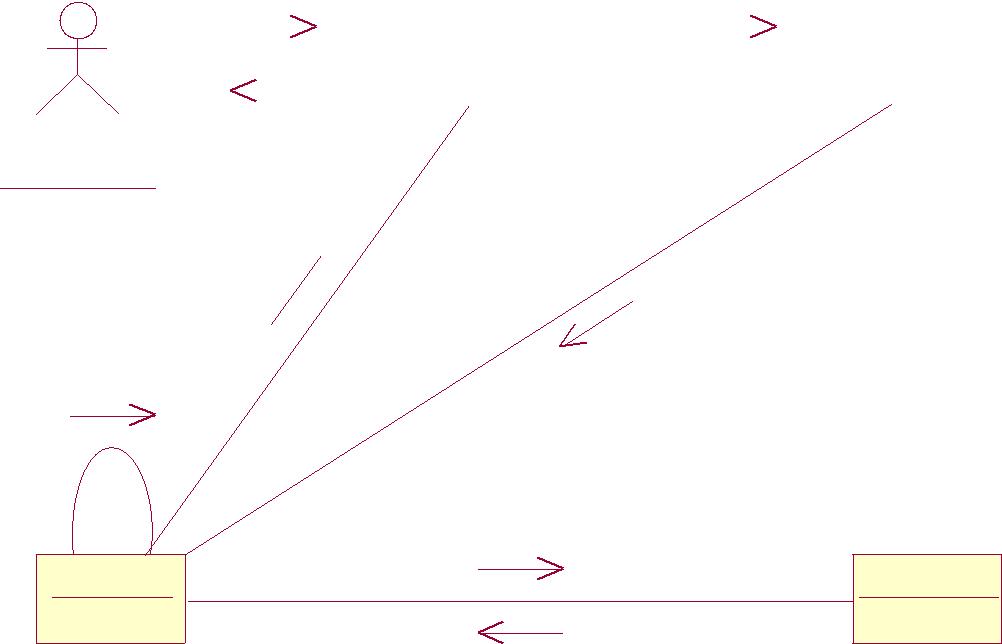
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | 1: Start | | | |  |  |  |  |  | 2: Initiate Cluster | | | | | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | Main | |  |  |  |  |  |  |  |  | Cluster | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Process | |  |  |  |
|  |  |  |  |  |  |  | 16: Results to User | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | : End User | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | 15: Display Clusters | | | | 3: Use K-Means | | | | |  |  |  |  |  | 11: Compute Time | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 12: Compute Space | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4: Choose Initial Cluster | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | Cost | | | | |  |  |  |  |
| 5: Find Mean | | | | | | | | | | |  |  |  |  |  |  |  |  |  | | |  |
|  |  |  |  |  |  |  |  | Computing | | | | |  |  |  |  |
| 6: Read Objects | | | | | | | | | | |  |  |  |  |  |  |  |  |  | | |  |

1. Update Cluster Mean
   * 1. Check Noise%
   1. Eleminate Noise

|  |  |  |  |
| --- | --- | --- | --- |
|  | 10: Compute | 13: Store |  |
|  |  |  |
| K Means |  | Session |  |
|  | 14: Back to Process |  |  |

LINE CHART:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1: Show Analysis | | | | | | 2: Read | | | | | |  |
|  |  |  |  | Main | |  |  |  |  | Session | |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |



8: View Line Chart

* End User
  1. Display Line Chart 

3: Read Time of All Algorithms

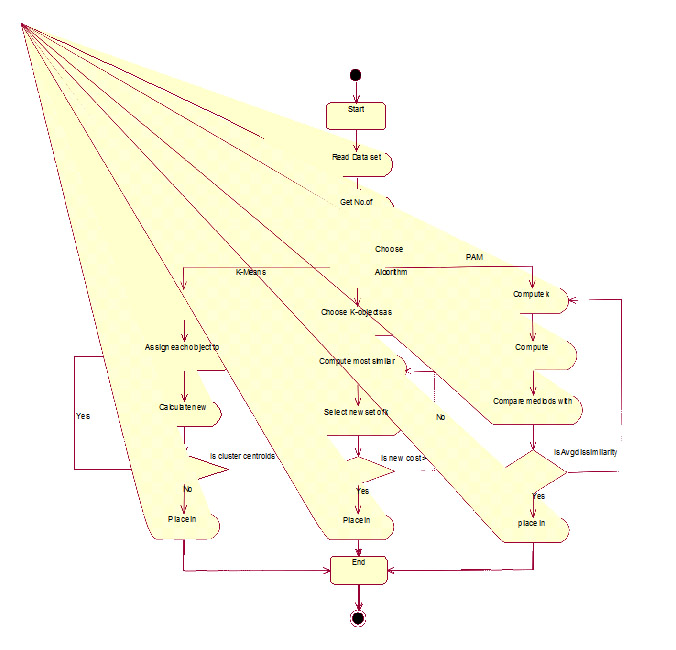
4: Plot

|  |  |
| --- | --- |
|  | 5: Draw Line Chart |
| Analysis | Line Chart |

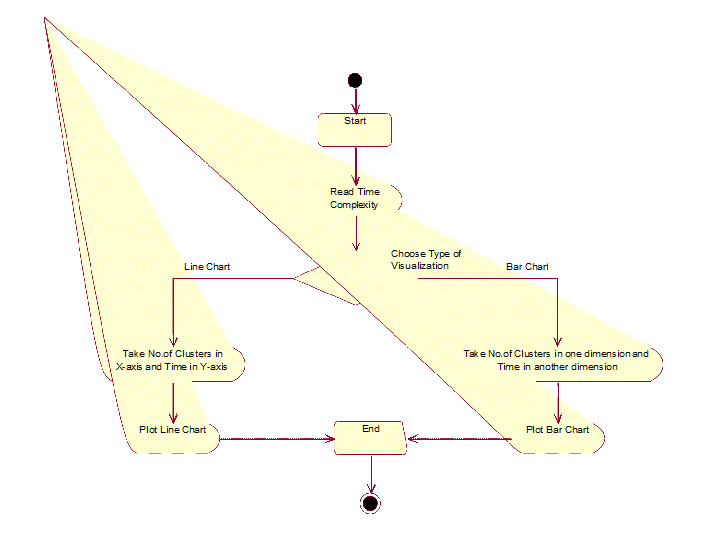
6: Chart Created

**ACTIVITY DIAGRAM** :

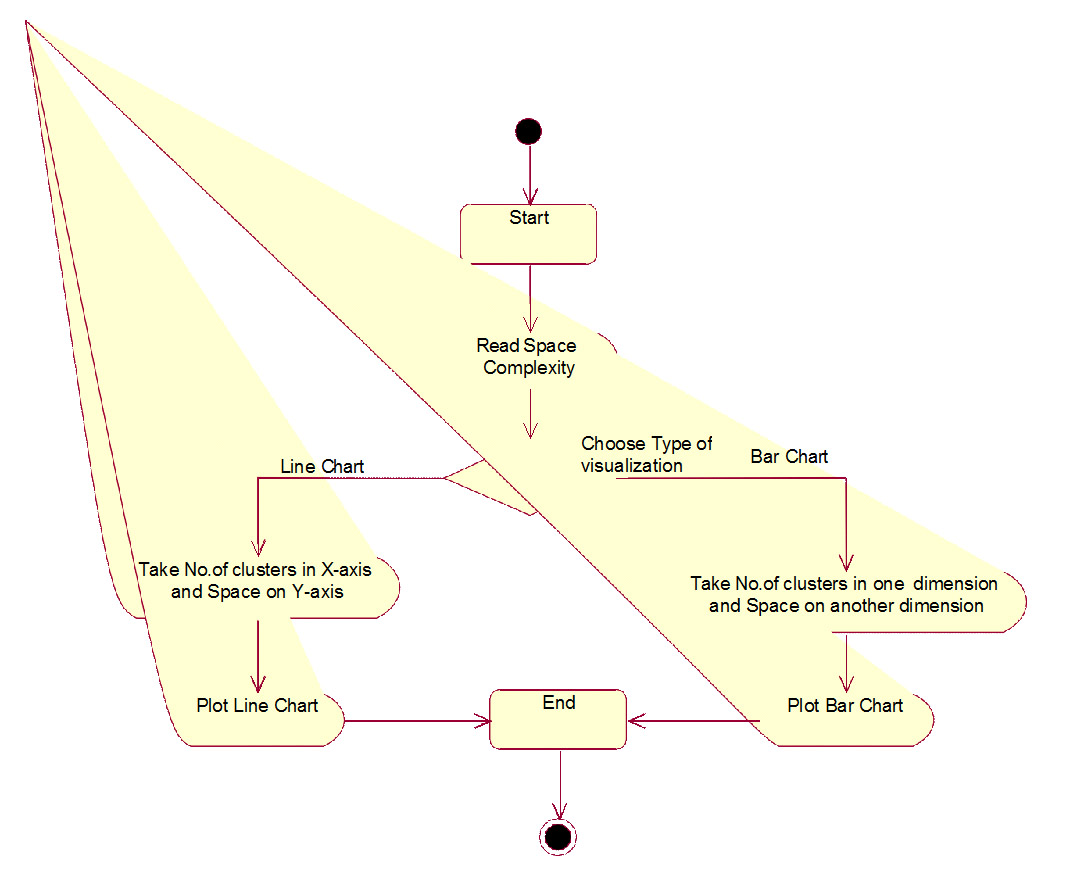
Activity Diagram for K-Means Algorithm:



**Time Complexity Diagram**:



**Space Complexity Diagram**:



**State Chart Diagram:**

