**Student Advisory System**

**Table of Contents**

1. INTRODUCTION...................................................................................................... 3  
1.1. Purpose of Document ........................................................................................ 3  
1.2. Intended Audience.............................................................................................. 3  
2. OVERALL SYSTEM DESCRIPTION............................................................................ 3  
2.1. Project Background............................................................................................ 3  
2.2. Project Scope..................................................................................................... 3  
2.3. Not In Scope....................................................................................................... 3  
2.4. Project Objectives.............................................................................................. 4  
2.5. Stakeholders...................................................................................................... 4  
2.6. Operating Environment..............................................………………………………………4  
2.7. System Constraints........................................................................................... 4  
2.8. Assumptions & Dependencies .......................................................................... 4  
3. EXTERNAL INTERFACE REQUIREMENTS……………………………………………………………….4  
3.1. Hardware Interfaces........................................................................................... 4  
3.2. Software Interfaces............................................................................................ 4  
4. FUNCTIONAL REQUIREMENTS…………………………………………………………………………….5  
4.1. Use Cases.......................................................................................................... 5  
4.1.1. [Title of use case]........................................................................................... 5  
5. FUNCTIONAL REQUIREMENTS……………………………………………………………………………8  
5.1. Major Requirements......................................................................................... 8  
5.2. User Documentation......................................................................................... 8  
SDS......................................................................................................................... 8  
6. SYSTEM ARCHITECTURE..................................................................................... 8  
6.1. SYSTEM LEVEL ARCHITECTURE........................................................................ 9  
6.2. SOFTWARE ARCHITECTURE............................................................................. 9  
7. DESIGN STRATEGY............................................................................................. 10  
8. DETAILED SYSTEM DESIGN ................................................................................ 10  
8.1. DATABASE DESIGN.......................................................................................... 10  
9. APPLICATION DESIGN…………………………………………………………………………………………11

10. REFERENCES..................................................................................................... 19  
11. APPENDICES...................................................................................................... 19

**1. Introduction**  
1.1. Purpose of Document

This document is intended to concisely summarize the outcomes of the project. This is used to document project successes, lessons learned and performance. It will provide a bird eye to our Advisory System.

1.2. Intended Audience

This report is a comprehensive description of our project. It will help our instructor to understand and evaluate the advisory system.

1.3 Definition of Terms, Acronyms and Abbreviations  
ML: Machine Learning

DB: Data Base

CS: Computer Science

ME: Mechanical Engineering

1.4 Document Convention   
Font style: Calibri

Font Size:

Heading 16

Sub-Heading 14

Body 11

**2. Overall System Description**2.1. Project Background

Section and course allocation is difficult in many universities, so to overcome this problem we have designed a system which works with the help of Machine Learning that divides students into sections according to their background and mentality.

2.2. Project Scope

Project consist of two main parts one is to predict the sgpa for next semester of students based on their previous data which will help them to select courses accordingly and other is to group similar (with same background and mentality) students in same section.

2.3. Not in Scope

* Networking()
* Security()

2.4. Project Objectives  
  
In many universities, section and course are allocated, so we have developed a system that helps with learning machine and distributes students according to their educational profile. And helps them register courses.

2.5. Stakeholders

* Students
* Teachers

2.6. Operating Environment

* Java JRE
* XAMPP server for Database

2.7. System Constraints

* Students and teachers are pre registered.
* Does not have any concern with security of high level.
* No networking and server connection
* MySqL is used as Database

2.8. Assumptions & Dependencies

* Students are per-registered
* Teachers are pre-hired
* Only two departments i.e CS and ME

**3. External Interface Requirements**  
3.1. Hardware Interfaces

Vector support architecture is required to process data efficiently.

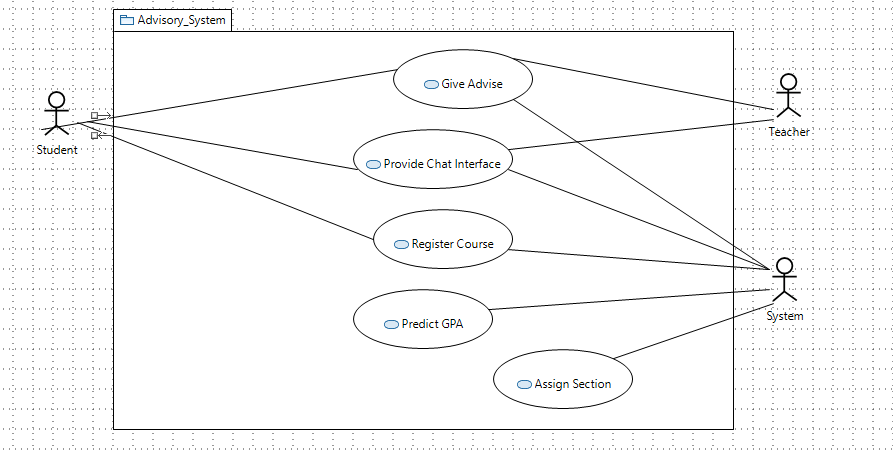
3.2. Software Interfaces

Windows: 10 or 8.1

Java SE Runtime Environment 8

XAMPP for Database

**4. Functional Requirements**  
4.1. Use Cases



4.1.1

Use Case name: Give Advice  
Use Case Description: First of all student will login to the system. After the successful login student will ask for advice from the system. System will apply machine learning and display the advice to student as well as teacher. Teacher will read the advice and can add some additional suggestion if needed, then finally display the advice to the student.

Primary actor: Student   
Other actors: Teacher, System  
Stakeholders:   
Relationships  
 Includes:  
 Extends:  
Pre-conditions: Student needs to login.   
Flow of Events:   
1. Student Logins

2. Ask advise from the system.  
3. System will apply machine learning

4. Display advice to student and teacher.

5. Logout.  
Alternative and exceptional flows:  
1.1 Login failed.

4.1 Teacher will add his suggestion.   
Post-conditions: Student must be able to get what the university is trying to advice.Teacher will get to know about the student profile.   
  
Use Case name: Provide Chat interface  
Use Case Description: Chat box is basically the interface between the student and teacher to let them communicate. Firstly students and teachers will login with their respective accounts. After the successful login they will be redirected to chat room. Where both can send the messages to each other but all the messages will be monitored by the system.

Primary actor: Student, Teacher  
Other actors: System  
Stakeholders:  
Relationships  
 Includes:  
 Extends:  
Pre-conditions: Student and Teacher must login first to access chat box.  
Flow of Events:  
1. Student and Teacher needs to login.  
2. They can open chat box.   
3. Logout.   
Alternative and exceptional flows:  
1.1 Login Failed.  
Post-conditions: Student and Teacher must be able to communicate easily.

Use Case name: Register Course  
Use Case Description: In registration process the student will login to the system. Student will enter credentials into the system. System will verify the credentials. After the credentials are successfully verified, the list of courses will be displayed to the student. Then the student will select the course according to his/her background. As the student has selected the course the system will verify if the seats are available in the particular course or not. If the seats are available the system will confirm the student registration and end the activity.

Primary actor: Student   
Other actors: System  
Stakeholders:  
Relationships  
 Includes:  
 Extends:  
Pre-conditions: Student must login first.  
Flow of Events:  
1. Student will login.  
2. List of courses displayed to student.  
3. Select the course according to his/her background.

4. System will check the seat availability.

5. Registration confirmed.

6. Logout.   
Alternative and exceptional flows:  
1.1 Login failed.

4.1 Seats not available  
Post-conditions: Students successfully registered in their respective course.

Use Case name: Predict GPA  
Use Case Description: In this use case the system must will make use of a supervised learning algorithm called linear regression. It will make use of certain features like study time, absences, travel time etc.To calculate the GPA of first semester. The data mentioned above along with GPA of first semester will be used to calculate the GPA of second semester.  
Primary actor: System  
Other actors: data Analyst  
Stakeholders: Students  
Relationships  
 Includes: Machine learning software must be installed  
 Extends:  
Pre-conditions: Students must provide data, , data analyst must clean data  
Flow of Events:  
1. Analyst cleans the data

2. Analyst feeds the system  
3. Machine fetches the data  
4. Machine performs normalization to learn parameters

5. Machine calculates gpa using linear regression   
Alternative and exceptional flows:  
4.1 ….  
Post-conditions: Database is updated

Use Case name: Assign Section  
Use Case Description: In this use case the system will make use of an unsupervised machine learning algorithm called k-means clustering algorithm. It will make groupings based on certain features like GPA, health and so on.  
Primary actor: System  
Other actors: data Analyst  
Stakeholders: Students  
Relationships  
 Includes: Machine learning software must be installed  
 Extends:  
Pre-conditions: Students must provide data, data analyst must clean data.  
Flow of Events:  
1. Analyst feeds the system.  
2. System makes clusters

3. System assigns students to clusters  
4. Extra ones are eliminated  
Alternative and exceptional flows:  
4.1 …. Not any  
Post-conditions: Students are allocated to different sections. Database must be updated.

**5. Non-functional Requirements**  
5.1. Major Requirements:

Student advisory system basically uses local host for database and there is limited amount of data in database so,

* Data entered through software is updated within 5 seconds in database depends upon connection
* Changing in parameters of ML requires restart of system
* Panel loaded within 2 seconds
* Data searches take 3-4 seconds

although it is not tested by setting database on server level but it is construct in such manner that it will able to give better performance (not best) because data abstracted from database copied to object of classes than that object is used to show data to user the reason we use this method to implement possible OOP concepts. But database scheme is fully normalized to avoid repetition of data and provide faster access of data in server end.

Possible breaches of security is through sql injection and there is no check whether it is data or query which is entered in textbox and at software level login password is not either encrypted or decrypted so if it is possible to get all data through sql injection

5.2. User Documentation:

* Student Advisory System executable file
* User manual
* Database file (which is loaded on server)

**SDS**

**6. System Architecture**   
The Student Advisory System class runs on the system. It collects data from database and assigns it to all the objects of their respective classes. These classes then continue with the functionalities i.e. assigning data to the attributes. Thus, it assures that all the data of the system is maintained by their respective classes uniformly.

It works by assigning courses to teachers and students. It also assigns teachers to students and vice versa. It has all the classes associated to it therefore acts as the main class.

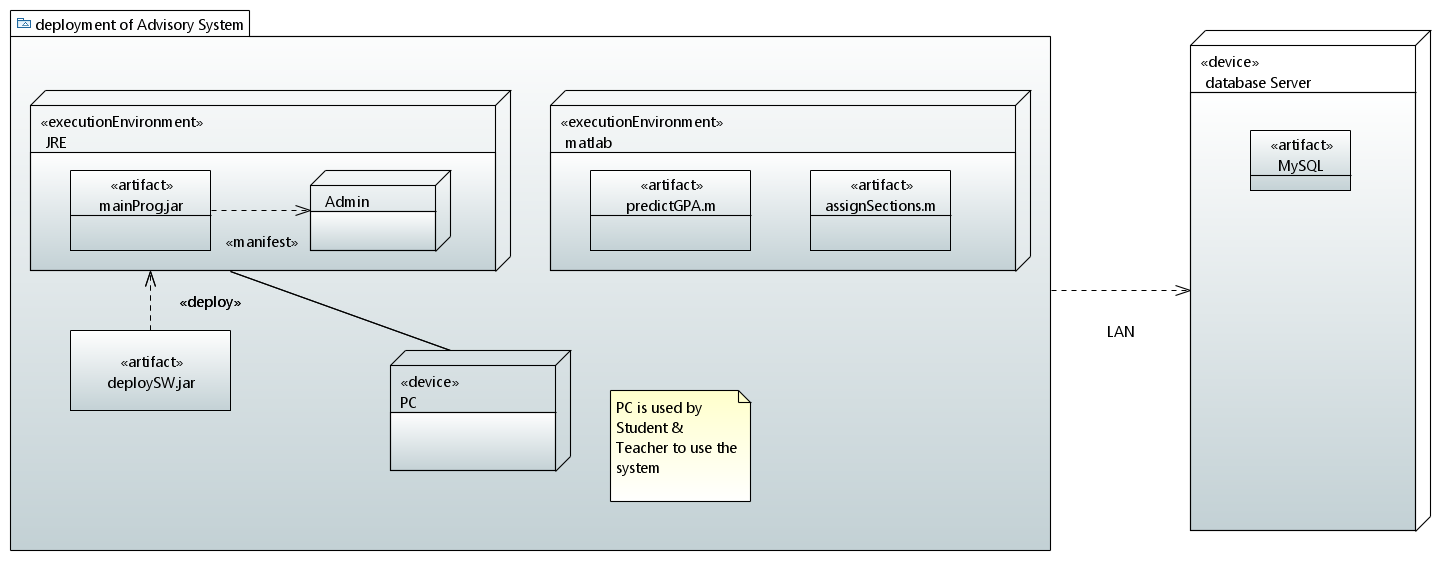
6.1. System Level Architecture  
The teacher class has the id and name of teacher. It also has student which assigned to each teacher as it is associated with the student class.

The student class has all the attributes of students. It has also teachers assigned to him/her as it is associated with the teacher class. Moreover, it also has the courses assigned to it is associated with course class. It also holds the GPA predicted through machine learning algorithm.

The course class contains all the course ids and their names.

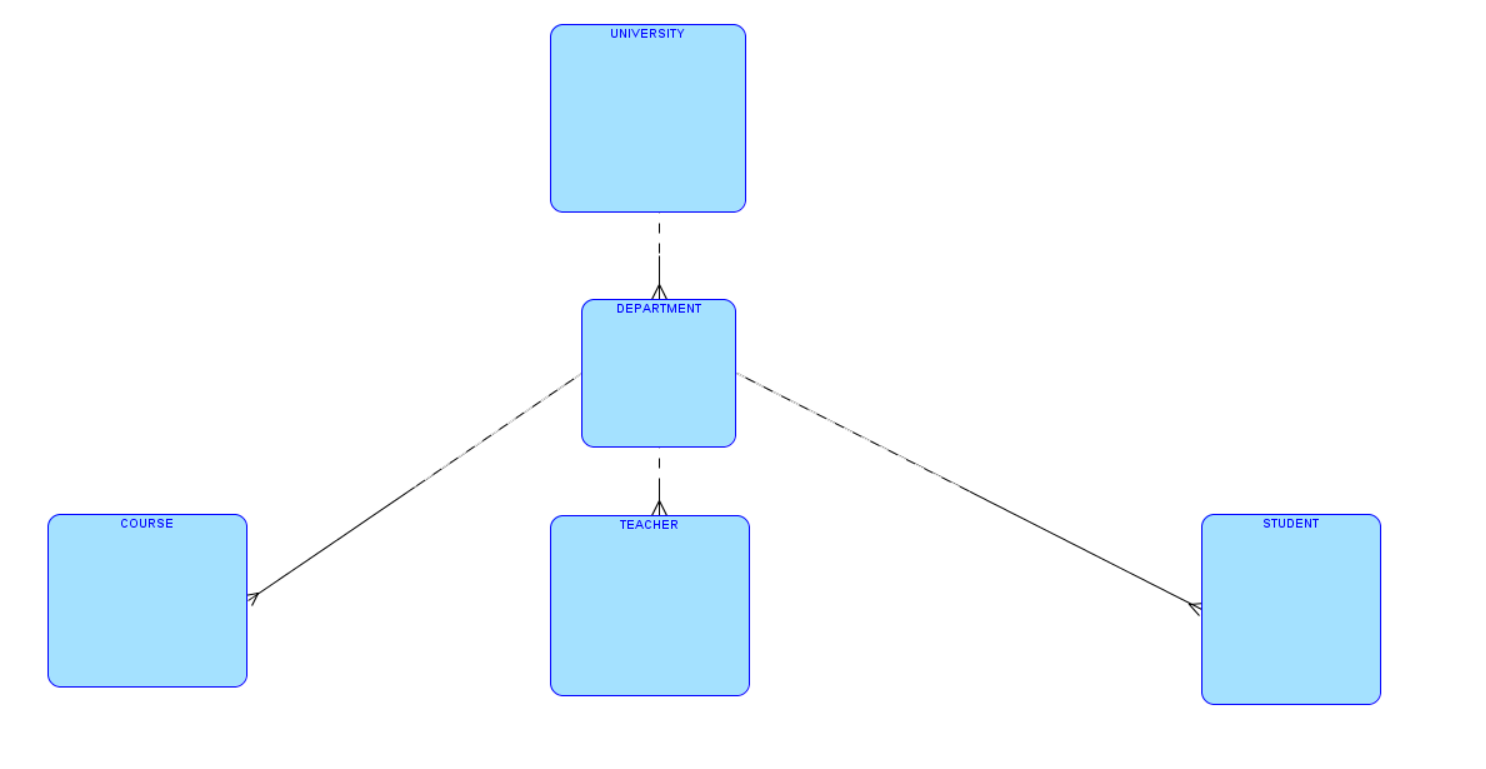
The department class has all the teachers, students, and the courses that if offers, all associated.

The Tokyo University class places all the students in their respective departments.

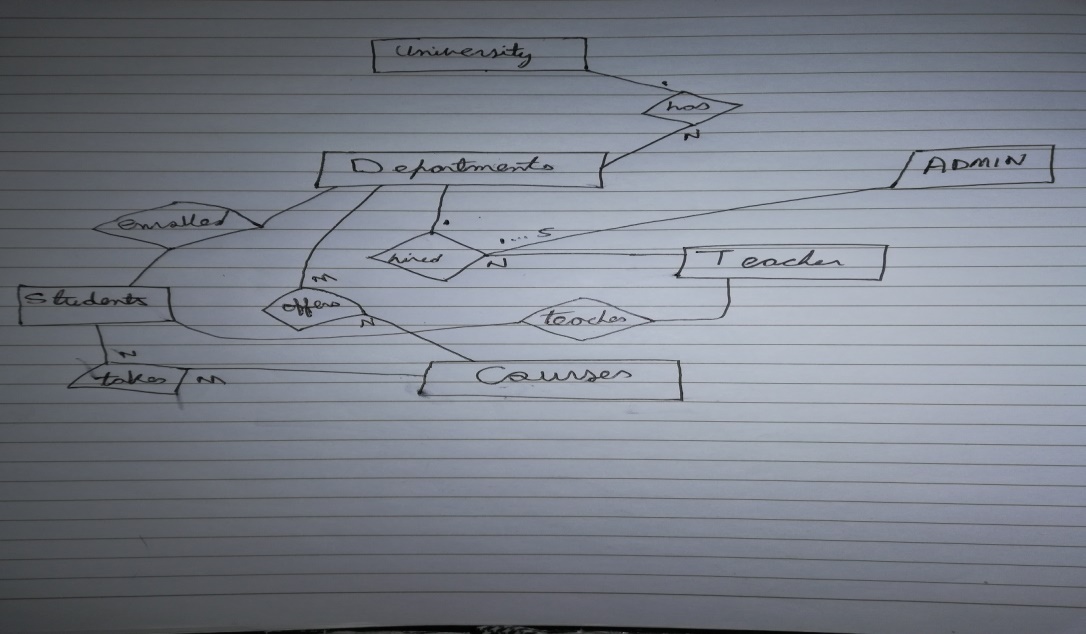


6.2. Software Architecture   
A class named DB handler is used to communicate with Database(MySQL) which is dependent on mysql-connector-java-8.0.1.3.

**7. Design Strategy**NA

**8. Detailed System Design**8.1. Database Design  


8.1.1. ER Diagram



8.1.2. Data Dictionary

NA

8.1.2.1 Data 1

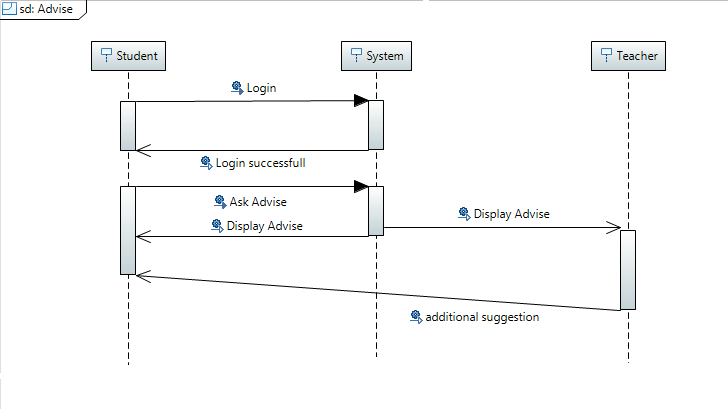
NA

8.1.2.2. Data 2

NA  
  
**9. Application Design**

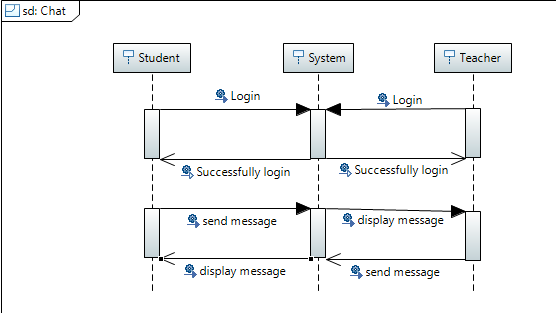
9.1.2. Sequence Diagram

9.1.2.1 <Sequence Diagram Advise>

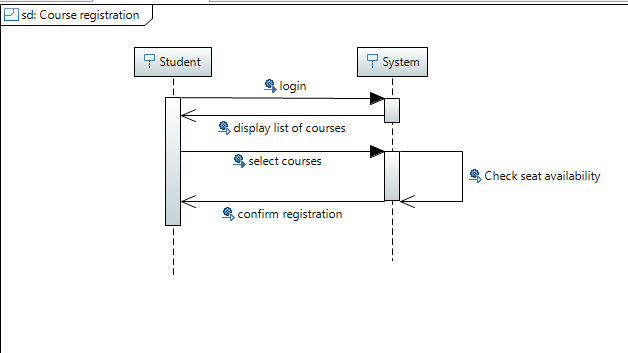


First of all student will login to the system. After the successful login student will ask for advice from the system. System will apply machine learning and display the advice to student as well as teacher. Teacher will read the advice and can add some additional suggestion if needed, then finally display the advice to the student.

9.1.2.2 <Sequence Diagram Chat >

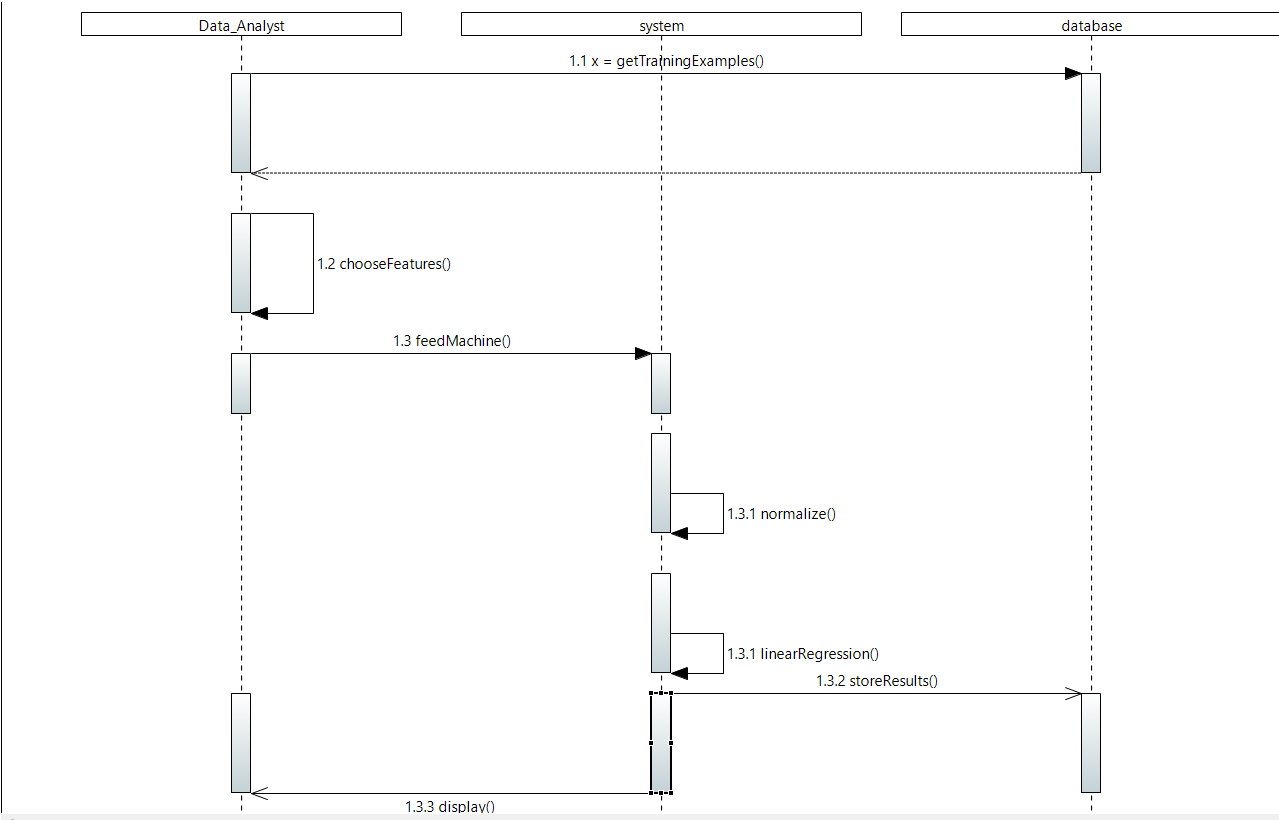


Chat box is basically the interface between the student and teacher to let them communicate. Firstly students and teachers will login with their respective accounts. After the successful login they will be redirected to chat room. Where both can send the messages to each other but all the messages will be monitored by the system.  
  
9.1.2.3 <Sequence Diagram Course Registration>

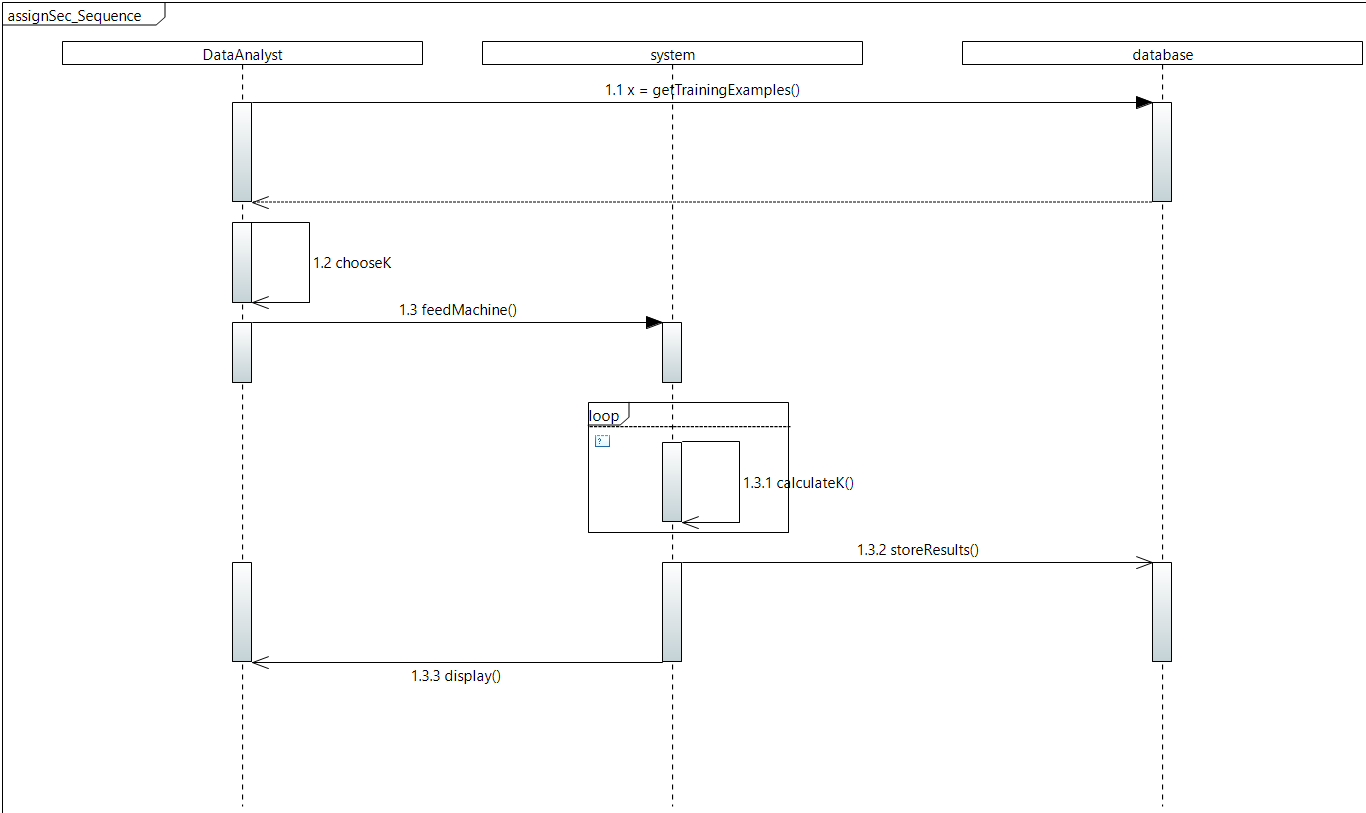


In registration process the student will login to the system. Student will enter credentials into the system. System will verify the credentials. After the credentials are successfully verified, the list of courses will be displayed to the student. Then the student will select the course according to his/her background. As the student has selected the course the system will verify if the seats are available in the particular course or not. If the seats are available the system will confirm the student’s registration and end the activity.

9.1.2.4 <Sequence Diagram Predict GPA>

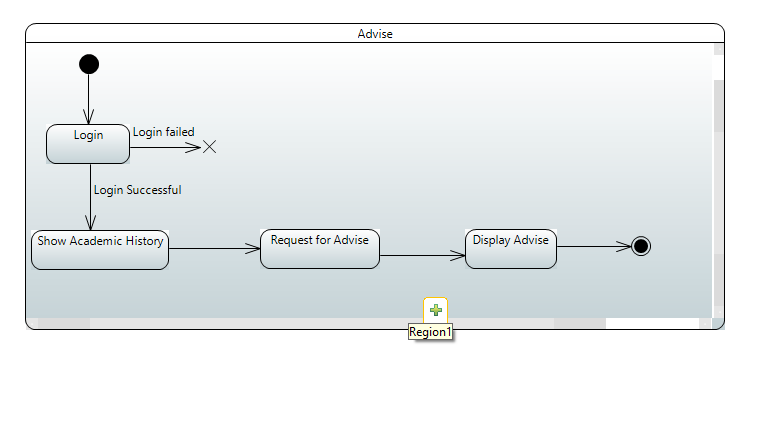


In this use case the system must will make use of a supervised learning algorithm called linear regression. It will make use of certain features like study time, absences, travel time etc. To calculate the GPA of first semester. The data mentioned above along with GPA of first semester will be used to calculate the GPA of second semester.

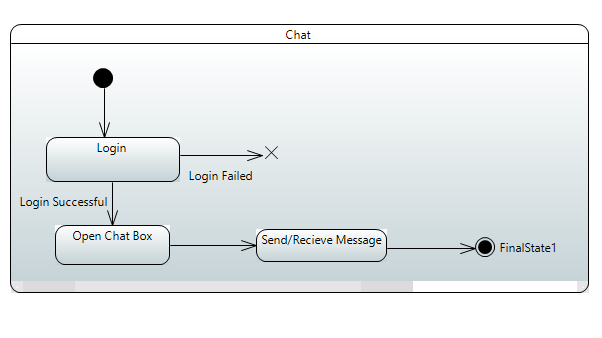
9.1.2.5 <Sequence Diagram Assign Section>  


The system will make use of an unsupervised machine learning algorithm called k-means clustering algorithm. It will make groupings based on certain features like GPA, health and so on.

9.1.3. State Diagram  
9.1.3.1 <State Diagram Advise>

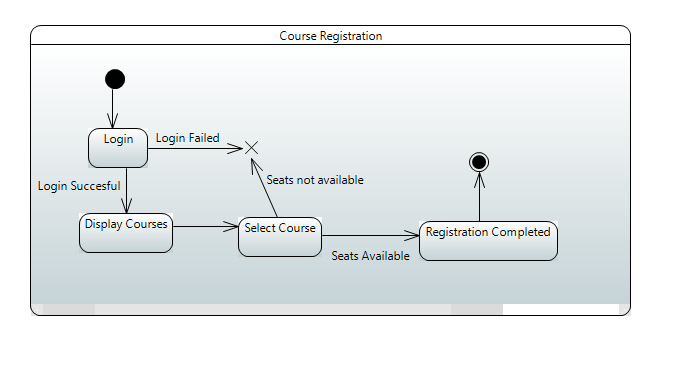


First of all student will login to the system. After the successful login student will ask for advice from the system. System will apply machine learning and display the advice to student as well as teacher. Teacher will read the advice and can add some additional suggestion if needed, then finally display the advice to the student.

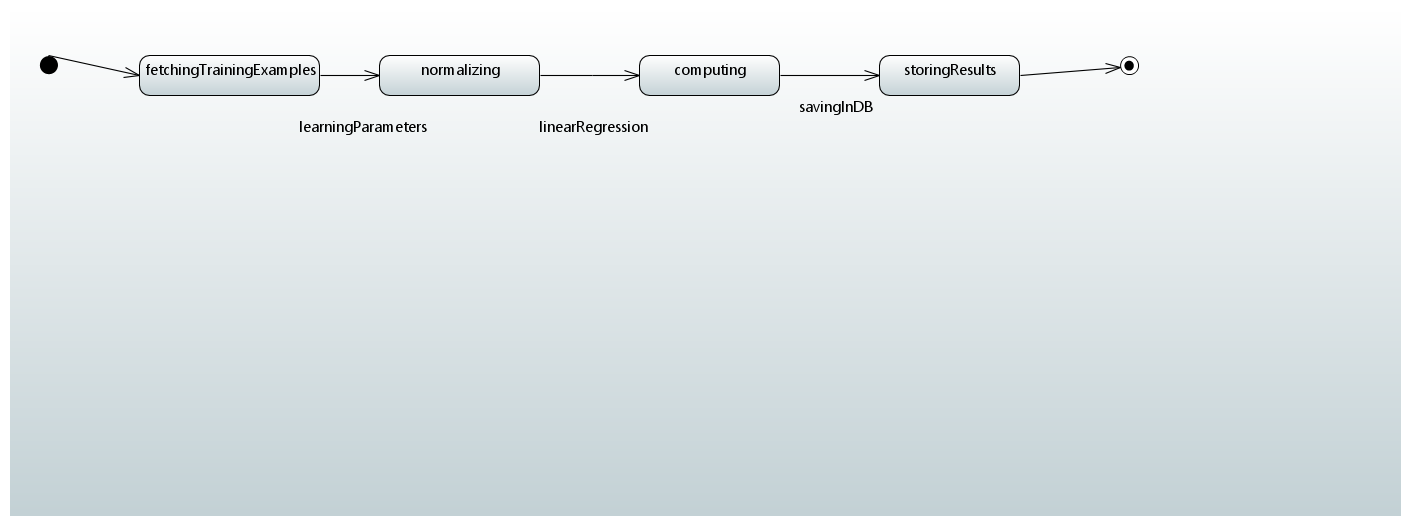
9.1.3.2 <State Diagram Chat>

Chat box is basically the interface between the student and teacher to let them communicate. Firstly students and teachers will login with their respective accounts. After the successful login they will be redirected to chat room. Where both can send the messages to each other but all the messages will be monitored by the system.

9.1.3.3 <State Diagram Course Registration>

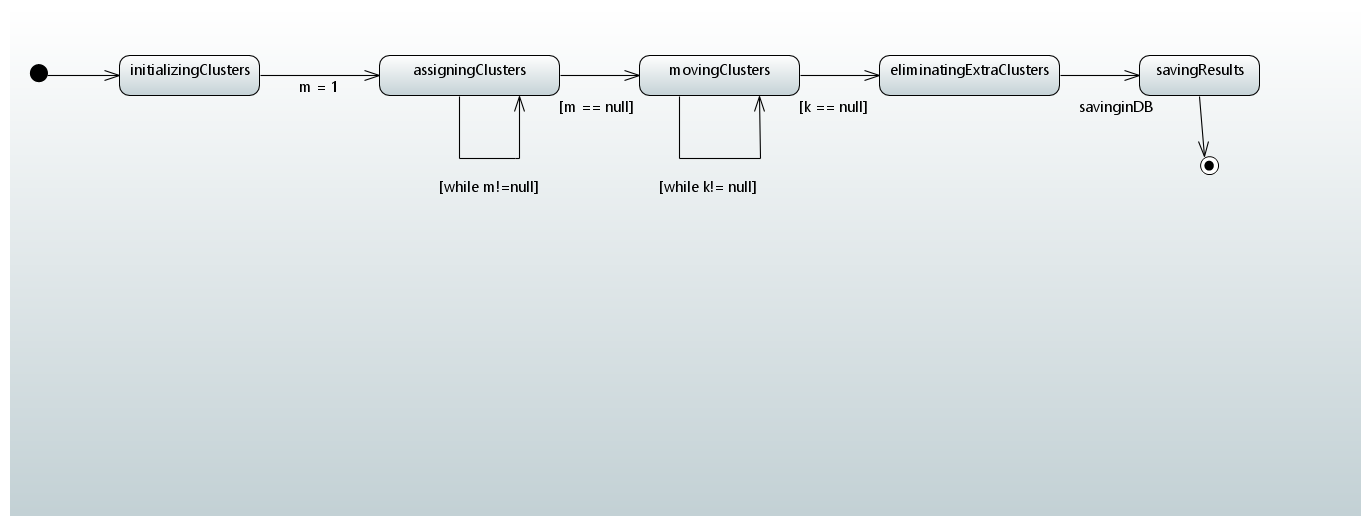


In registration process the student will login to the system. Student will enter credentials into the system. System will verify the credentials. After the credentials are successfully verified, the list of courses will be displayed to the student. Then the student will select the course according to his/her background. As the student has selected the course the system will verify if the seats are available in the particular course or not. If the seats are available the system will confirm the student’s registration and end the activity.

9.1.3.4 <State Diagram Predict GPA>  


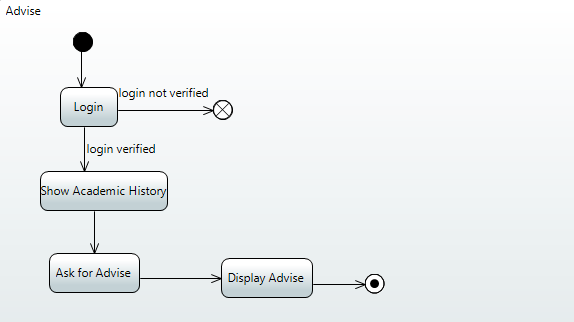
In this use case the system must will make use of a supervised learning algorithm called linear regression. It will make use of certain features like study time, absences, travel time etc. To calculate the GPA of first semester. The data mentioned above along with GPA of first semester will be used to calculate the gpa of second semester.

9.1.3.5 <State Diagram Assign Section>

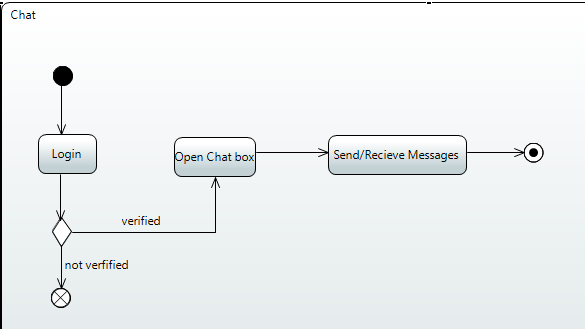


The system will make use of an unsupervised machine learning algorithm called k-means clustering algorithm. It will make groupings based on certain features like GPA, health and so on.

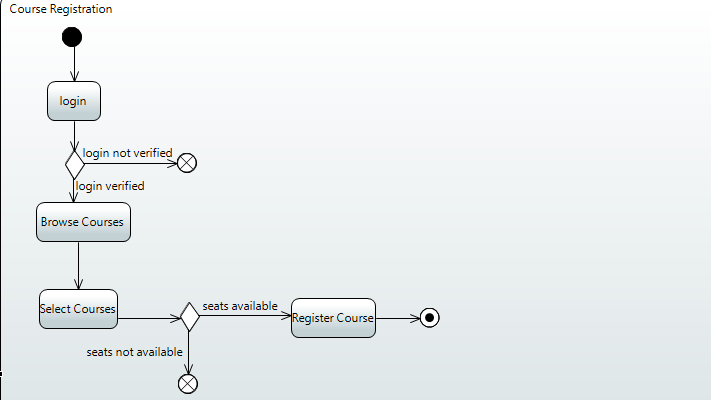
**9.1.4. Activity Diagram**  
9.1.4.1 <Activity Diagram Advice>



First of all student will login to the system. After the successful login student will ask for advice from the system. System will apply machine learning and display the advice to student as well as teacher. Teacher will read the advice and can add some additional suggestion if needed, then finally display the advice to the student.

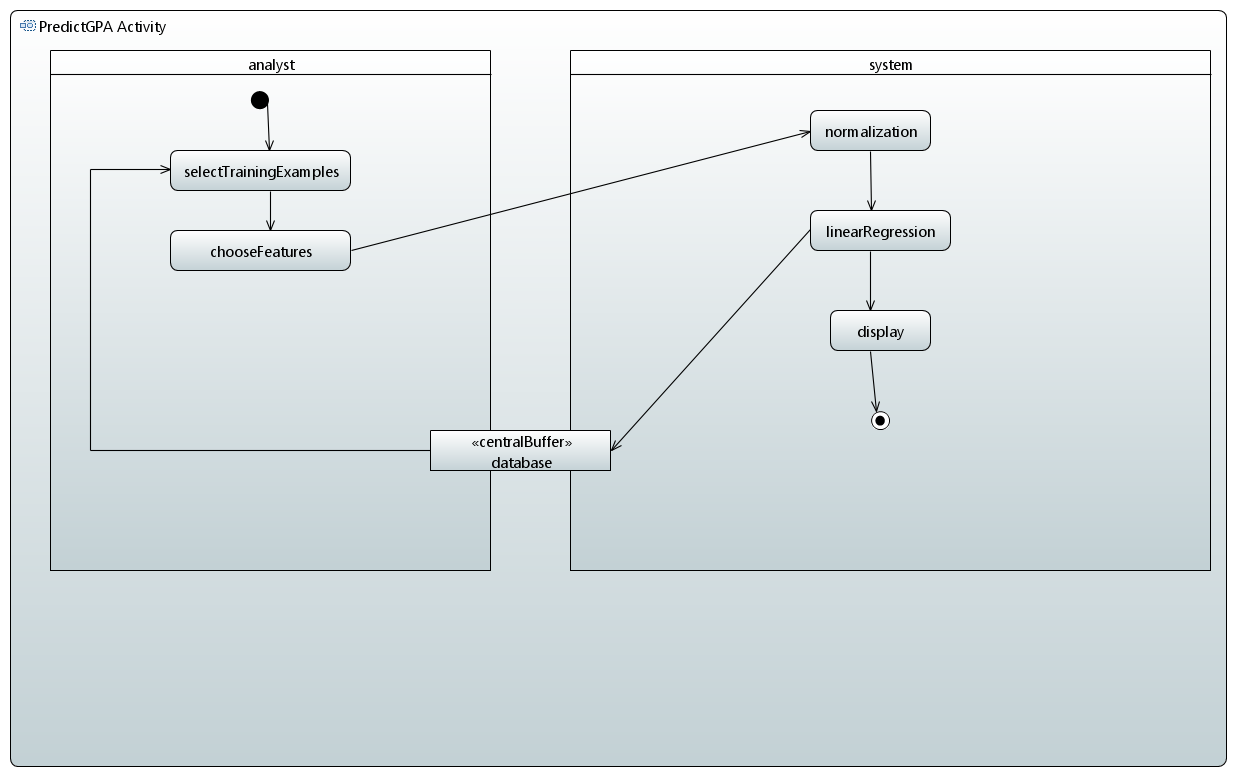
9.1.4.2 <Activity Diagram Chat>  
Chat box is basically the interface between the student and teacher to let them communicate. Firstly students and teachers will login with their respective accounts. After the successful login they will be redirected to chat room. Where both can send the messages to each other but all the messages will be monitored by the system.

9.1.4.3 <Activity Diagram Course Registration>



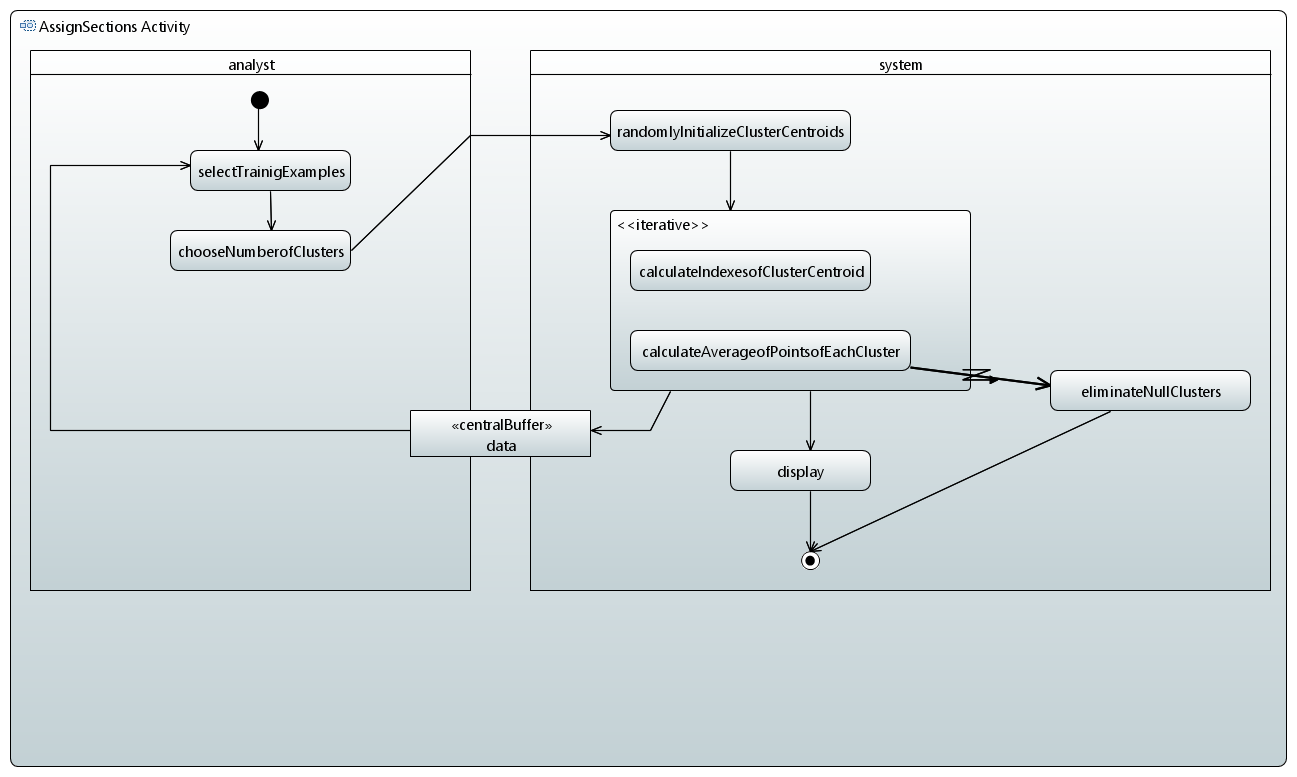
In registration process the student will login to the system. Student will enter credentials into the system. System will verify the credentials. After the credentials are successfully verified, the list of courses will be displayed to the student. Then the student will select the course according to his/her background. As the student has selected the course the system will verify if the seats are available in the particular course or not. If the seats are available the system will confirm the student’s registration and end the activity.

9.1.4.4 <Activity Diagram Predict GPA>



In this use case the system must will make use of a supervised learning algorithm called linear regression. It will make use of certain features like study time, absences, travel time etc. to calculate the GPA of first semester. The data mentioned above along with GPA of first semester will be used to calculate the GPA of second semester.

9.1.4.5 <Activity Diagram Assign Section>



The system will make use of an unsupervised machine learning algorithm called k-means clustering algorithm. It will make groupings based on certain features like GPA, health and so on.

**11. Appendices**

private javax.swing.JPanel panel: Contains javax.swing GUI components.

private const javax.swing.JLabel labelProjectName:

Constant header label, set

private const javax.swing.JLabel labelSourceFolder:

private const javax.swing.JLabel labelProgressBar:

private const javax.swing.JLabel labelSelected:

private javax.swing.JButton buttonNewProject:

private javax.swing.JButton buttonRemoveProject