Uni-Miner

By

Zohair Abbas

Danish Iqbal Shaikh

Aqeel Ahmed Shaikh

Submitted to the Faculty of Department of Computer Software Engineering,
Military College of Signals, National University of Sciences and Technology, Islamabad
In partial fulfillment for the requirements of B.E Degree in
Software Engineering JUNE 2016

Abstract

Uni-Miner

Uni-Miner is a smart, cross-functional, centralized and history-enabled data-warehouse and Business Intelligence system for the students of MCS.

The application will help users find an efficient way to integrate data from different source systems in university. It will be used to show centralized students' academic database and generate business intelligence system about their performance, based on which the system will give suggestions for any student.

Uni-Miner can be a valuable resource for all University users that need information to provide support for performance management, decision support at strategic, tactical and operational level, compliance reporting and day to day operations.

This will increase the awareness of teachers and the administration in reference to the students and the students will be able to self-help themselves.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that the work contained in the thesis – Web Application for Uni-Miner carried out by Zohair Abbas, Danish Iqbal Shaikh and Aqeel Ahmed Shaikh under the supervision of Lecturer Ayesha Naseer for partial fulfillment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

Lec. Ayesha Naseer

CSE DEPARTMENT
MCS

____

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent.
To our parents, without whose unflinching support and unstinting cooperation, a work of this magnitude would not have been possible.

Table of Contents

List of Figures

List of Table

Chapter 1: Introduction

Uni-Miner is a smart, cross-functional, centralized and history-enabled data-warehouse and Business Intelligence system for the students of MCS.

Uni-Miner can be a valuable resource for all University users that need information to provide support for performance management, decision support at strategic, tactical and operational level, compliance reporting and day to day operations.

This will increase the awareness of teachers and the administration in reference to the students and the students will be able to self-help themselves.

1.1 Problem Statement

Currently, the systems are totally manual and one needs to search through bulk of data to get required data. After this one has to perform complex calculation to get the required result. Obviously this is not only tedious, but time consuming as well.

1.2 Scope

The document only covers the requirements specifications for the Uni-Miner. All the external interfaces and the dependencies are also identified in this document.

For	Administration, Teachers and students
What	Data analyzing, generating business intelligence based results
The	Uni-Miner
Is	A web application
That	Provides smart, cross-functional, centralized and history-enabled data-warehouse and Business Intelligence system

Table 1-1: Project Scope

The application will be used to show centralized students' academic database and generate business intelligence system about their performance, based on which the system will give suggestions for any student. The scope of this project is limited to academic database of the students. Extended scope includes the library management system and admission system of the students.

1.3 Objectives

The objectives of the project include:

1. Find an efficient way to integrate data from different source systems in university.
2. Design and Develop Extract, Transform, Load (ETL) routines to integrate data into central data-warehouse.
3. Identify, Design and Develop Key Performance Indicators (KPIs).
4. Learn to design and develop BI Dashboard and Reports.

1.4 Deliverables

Sr.	Tasks	Deliverables
1.	Literature Review	Literature Survey
2.	Requirements Gathering	SRS Document
3.	Application Design	Design Document
4.	Implementation	Web based application implementation
5.	Testing	Evaluation plan and test document
6.	Deployment	Complete application along with necessary documentation

Table 1-2: Project Deliverables

Chapter 2: Literature Review

2.1 Introduction

Today the systems to evaluate and analyze the results of students are totally manual. Students and teachers needs to search through loads of data in form of mark sheets or even on online results to get the relevant data. To find any particular result, systems are available which is obviously not efficient in case of combined analysis of performance of students. Even after this, users have to perform complex calculation to get the required result. Obviously this is not only tedious, but it requires a lot of time as well.

Uni-Miner will help the users in following major cases:

1. Students' graphical based results
2. Overall class graphical based results
3. Prediction of students results in future
4. Overall class prediction

2.2 Background

Parents and higher authorities including the teachers themselves are very much concerned about the students. How they perform and where do they stand is the most concerned question. Often teachers have to present the students' results to either the parents or higher authorities of any institute.

In such cases Uni-Miner helps them to present the results in a more precise way and they can even tell and warn the students about their future result prediction and particularly about how they should perform in future to maintain good results.

Students can also get to know about future prediction of their results and where they stand. This will help them boost themselves in the area they are weak and get good results.

2.3 Related Work

Previously application based on information management system and library management systems have been made.

Work done includes students' database systems, student information management system, and Library Management System.

This work of introducing business intelligence (data analysts) in the students' information, and getting wanted information about students, is being done for the first time.

Chapter 3: Software Requirement Specification

3.1 Overall Description

3.1.1 Product Perspective

Uni-Miner provides the users to perform analysis on students' academic profile. A database of the academic profile of currently studying students' will be maintained by the system.

3.1.2 Product Function

1. Insert or update students' profile (main/academic)
2. Displaying students' main profile
3. Displaying students' previous academic profile
4. Displaying students' current academic profile
5. Displaying students' previous, current and future academic analysis
    a. Bar graphs
    b. Line graphs
    c. Pie graphs
    d. Attendance graphs
    e. Displaying any suggestions for any particular student

3.1.3 Operating Environment

3.1.3.1 Hardware

Uni-Miner operates, either directly or indirectly, with the following external hardware:

Workstations:

Servers with at least 10 GB RAM, 500 GB hard disk will be capable to host the server application, run server application and maintain the database.

Customer Peripherals:

The PCs, PDAs, and smart phones used by end-users over the Internet using a web interface.

Internet:

The global network used for communication.

3.1.3.2 Software

Uni-Miner operates, either directly or indirectly, with the following software:

Operating System:

Windows based, only with networking allowed, through which this app shall communicate with internet. Windows XP or superior is required.

Browser:

Website should be compatible to the following browsers:

The software tool that runs on users' personal computers that allows them to communicate over the internet. The browser version will have to be at least IE6, Mozilla Firefox 2 and equivalent.

3.1.4 Design and Implementation Constraints

3.1.4.1 Data and Content Constraints

1. Alpha numeric type fields cannot hold number data types and vice versa.
2. Each data field will have its type data.
3. Business intelligence graphs cannot be edited.
4. Each and every student data will be stored in data base and will be present till deleted.
5. Only authorized users
    a. Will be allowed to edit any information
    b. Will be able to access all the information about students.
6. Each student will be allowed to access only his/her profile less editing.
7. Students' future performance data shall be purely generated on basis of its previous performance.

3.1.4.2 Hardware Constraint

Pentium core based computers, used by operator, to perform their tasks. These will have to be equal to or superior than the Pentium 4 core with minimum of 1 GB storage space and 256 MB memory.

3.1.4.3 Software Constraints

The software should be compatible with popular databases (Oracle, SQL Server etc.)

3.1.4.4 High-Level Languages

Following languages will be used:-

1. Java Script
2. PHP
3. HTML & CSS

3.1.5 User Documentation

A few documents shall be delivered along with the software:

1. Installation guides.
2. Usage manuals with pictures and text for using the software.
3. Help manuals for removing errors and bugs.
4. Help links.

3.1.6 Assumptions and Dependencies

The group is appreciative to state the following assumptions while making this SRS.

3.1.6.1 Internet

As mentioned earlier this is a web application, so internet is necessary.

3.1.6.2 Technical Knowledge Limitation

The group is only well versed with one language of the 4th generation languages today (e.g. C#, JAVA, .NET) that enable creation of GUIs and modern APIs. The group has received formal training in C and has studied Object Oriented Programming, networking in java and also has studied networking or web applications that would enable them to develop networked applications that would run on the internet and telephone lines. The group shall, therefore, in accordance with the training they have and the one they are undergoing in their 7th semester of their undergraduate degree, try to develop as good a product as they can according to their expertise level.

3.2 External Interface Requirements

This section contains the requirements specification for interfaces among different modules of the software and their external capabilities. It presents a high level of abstraction of the basic elements of the Graphical User Interface (GUI) and the display screens of the application.

3.2.1 User Interfaces

The screen formats and menu structure should be in such a way that users find it easy to use. The product must be user-friendly and very interactive. The functionality provided by the system like displaying dash-boards should adapt itself to the different users of the software

1. First of all login page will appear.
2. Depending on the user ,respective analysis will be shown
3. Sequence will be as follows
4. Login page

5. Home page
6. Respective analysis dashboard/report

3.2.2 Hardware Interfaces

N/A – No Hardware interfaces are required in our system.

3.2.3 Communications Interfaces

N/A – No Communications interfaces are required in our system.

3.3 System Features

As the customer opens the app, he/she is given the option to select whether he/she will be using the application as administrator, faculty or student. Then the user will be allowed further respective functionalities will be provided after selection.

3.3.1 Login

3.3.1.1 Description and Priority

1. The purpose is to know whether the admin, faculty or the student is trying to access the data, so that the access shall be granted accordingly.
2. Priority Level: HIGH

3.3.1.2 Stimulus/ Response Sequences

1. User will enter his/her credentials.
2. A new window will be shown, allowing the user to access limited data accordingly.

3.3.1.3 Functional requirements

1. If the user enters wrong credentials, access would not be granted.
2. User can access the data, if correct credentials are provided.

3.3.2 Insert/Edit student data

3.3.2.1 Description and Priority

1. The purpose is to allow the customer to insert any students' data or update it if some changing is required.
2. Priority Level: HIGH

3.3.2.2 Stimulus/ Response Sequences

1. First user will enter students' personal data.
2. Secondly user will enter students' academic data.
3. And save the students record.

3.3.2.3 Functional requirements

1. If user has left any data field(s), the application will not further proceed and will generate error box.

3.3.3 Display student profile

3.3.3.1 Description and Priority

1. This feature of application will display the students profile to the user.
2. Priority Level: HIGH

3.3.3.2 Stimulus/ Response Sequences

1. Once the user has accessed the students' information, the data will be visible in a new window.
2. User can also edit the data based on, if the user is authorized to do so.

3.3.3.3 Functional requirements

1. User will not be able to edit data if he/she is unauthorized user. While editing if the user enters some invalid data, so the data will not be accepted and will show error box.
2. The user will have to re-enter the data about the student.

3.3.4 Display student academic data

3.3.4.1 Description and Priority

1. This feature of application will display the student's academic data to the user.
2. Priority Level: HIGH

3.3.4.2 Stimulus/ Response Sequences

1. Once the user has accessed the students' academic information, the data will be visible in a new window.
2. User can also edit the data based on, if the user is authorized to do so.

3.3.4.3 Functional requirements

1. User will not be able to edit data if he/she is unauthorized user. While editing if the user enters some invalid data, so the data will not be accepted and will show error box. The user will have to re-enter the data about the student.
2. In case the user is not an authorized user to edit the data but stills tries to do so, so the user will not be allowed to edit the data and the system will show warning box.

3.3.5 Display business intelligence reports

3.3.5.1 Description and Priority

1. This feature of application will display the student's academic data but in the form of graphs to the user.
2. Priority Level: VERY HIGH

3.3.5.2 Stimulus/ Response Sequences

1. Once the user has accessed the students' academic information in form of graphs, BI system, the data will be visible in a new window.
2. User can NOT edit the data in any case. More over the user can see the suggestions given by the system based on BI System.

3.3.5.3 Functional requirements

1. The user will not be allowed to edit the business intelligence reports. Moreover the students' academic details would be used to generate the business intelligence reports.

3.3.6 Display suggestions

3.3.6.1 Description and Priority

1. The purpose is to allow the customer to check the suggestions given by the system based on the BI systems.
2. Priority Level: HIGH

3.3.6.2 Stimulus/ Response Sequences

1. Once the user has accessed the students' suggestions, the data will be visible in a new window.
2. User can NOT edit the data in any case.

3.3.6.3 Functional requirements

1. The user will not be allowed to edit the suggestions. Moreover the business intelligence would be used to generate the suggestions.

3.4 Other Nonfunctional Requirements

3.4.1 Performance Requirements

3.4.1.1 Response Time

1. The response time of the system would be less than a second as soon as it gets the relevant data from database.
2. The response time for saving the data may be 3-4 seconds based on the capacity of Database.

3.4.1.2 Capacity

1. The system can save multiple data at a time.

3.4.1.3 Accuracy

1. The system would be able to save the changes and show the Business Intelligence Systems and suggestions based on those BI Systems.

3.4.1.4 Availability

1. The systems' helping document will be provided in English.

3.4.1.5 Safety Requirements

1. The system shall not accidently lose/delete the data associated with it, such as files created after summary.

3.4.2 Security Requirement

1. Only authorized user would be able to add/edit the students' data.
2. The students will only be authorized to see their data and check BI system based suggestions and their reports.

3.4.3 Software Quality Attributes

3.4.3.1 Availability

1. The system can be used 24/7 as long as the computer system works properly.

3.4.3.2 Correctness

1. If the user gives right/defined data for a particular entry, the reports will be fine.

3.4.3.3 Extensibility and Maintainability

1. Changes required will be applied in at least a month.
2. The system can be further extended based on extension of time.
3. The system can be improved to generate BI Reports of students, of the data other than academics, for example library management system, sports, medical fitness, PET, etc.

3.4.3.4 Portability

1. The system can be installed on any compatible hardware meeting the requirements.

3.4.3.5 Reliability

1. The system will be available to the users for 24 hours
2. The factors needed to establish the software expected reliability are:
3. The user inputs should be valid and within the given range.
4. Normal termination of the program

3.4.3.6 Usability

1. The system will be very easy to use by the users after being trained. However, the output of the system will be understandable to all the end-users.

Chapter 4: Design and Development

4.1 Introduction

The main objective of this chapter is to illustrate the design of the project. It will give detailed description of the design and patterns of system's interface and the restrictions under which it must function. This will help the executive users to have a centralized view of students' academic database and presentation of business intelligence system.

Currently, the systems are totally manual and one search through bulk of data to get required data. After this one has to perform complex calculation to get the required result. Obviously this is not only tedious, but time consuming as well.

4.2 System Architecture Description

Detailed description of system architecture and design pattern which this system is going to use is discussed later in the document in section 5 'Design Decisions and Tradeoffs'.

4.2.1 Overview of modules/components

Uni-Miner comprises of following components:

1. Graphical User Interface
2. Accounts Management
3. Processor
    a. Attendance
    b. CGPA
    c. Subjects

    d. Categories

4. Data Base Management System
5. Graphs API

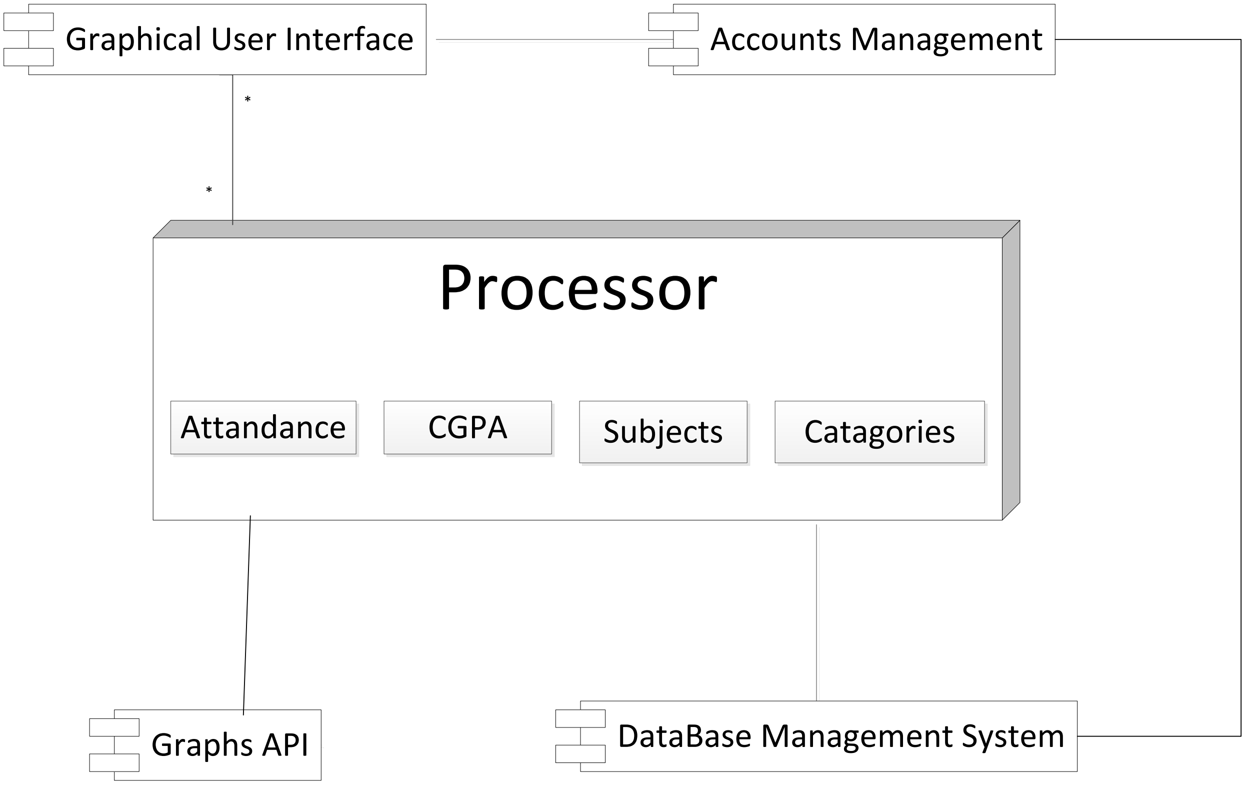


Fig 4-1: Components of Uni-Miner

Graphical User Interface: The user interacts directly with this component and provides an input to the application specifying the action required. The component then displays an output according to that user action.

Accounts Manager authenticate the user and checks whether he/she is allowed to perform particular task or not.

Processor gets data from database and performs calculation .It has four sub-components: attendance module manages the student’s attendance, CGPA module calculate CGPA based on his previous performance in past semester. Subjects calculates grade in that subject based on his already scored marks .Categories calculate overall performance of particular students category based on their previous performances.

Graphs API will get processed data from processor and will show that result in the form of graphs.

Data Base Management System is responsible for managing data in the database.

4.2.2 Structure and Relationships

This section covers the overall technical description of Uni-Miner. It shows the working in perspective of different point-of-views and also shows relationships between different components.

4.2.2.1 System Block Diagram

This diagram shows the higher level description of the system. It shows generic working of the system and interaction with the user. In this system, User will interact with the UI of the system, UI will interact with the processor which will then interact with database and Graphs API Processor will perform required calculations The Graphs API will get results from processor and will return graphs.

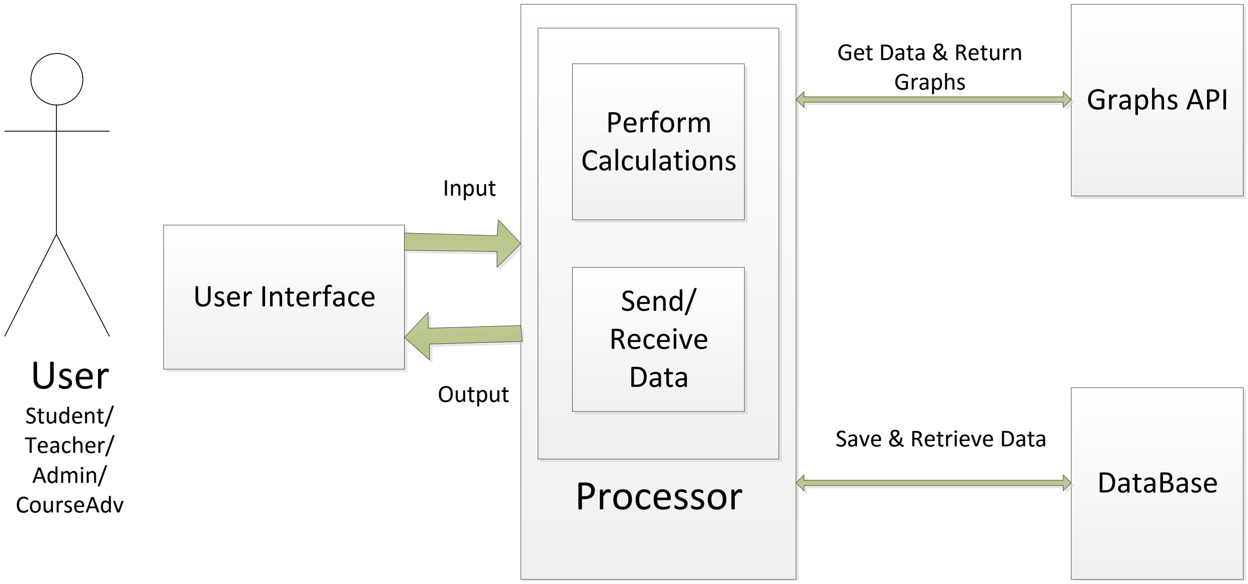


Fig 4-2: Block diagram for Uni-Miner

4.2.2.2 Database Design

Uni-Miner requires user’s credentials, course data along with subject’s info, which are being taught.

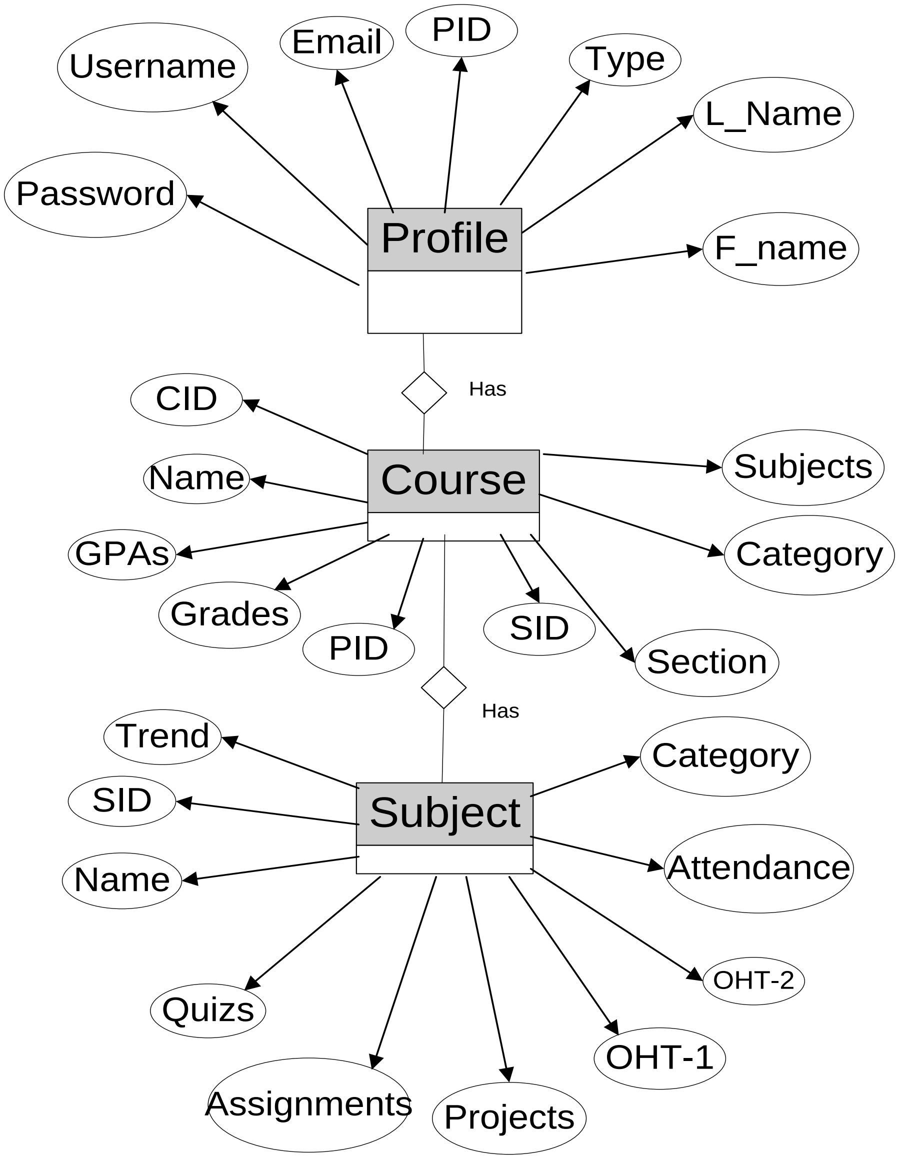


Fig 4-3: Entity-Relationship diagram for Uni-Miner

4.2.2.3 User view (Use Case Diagrams)

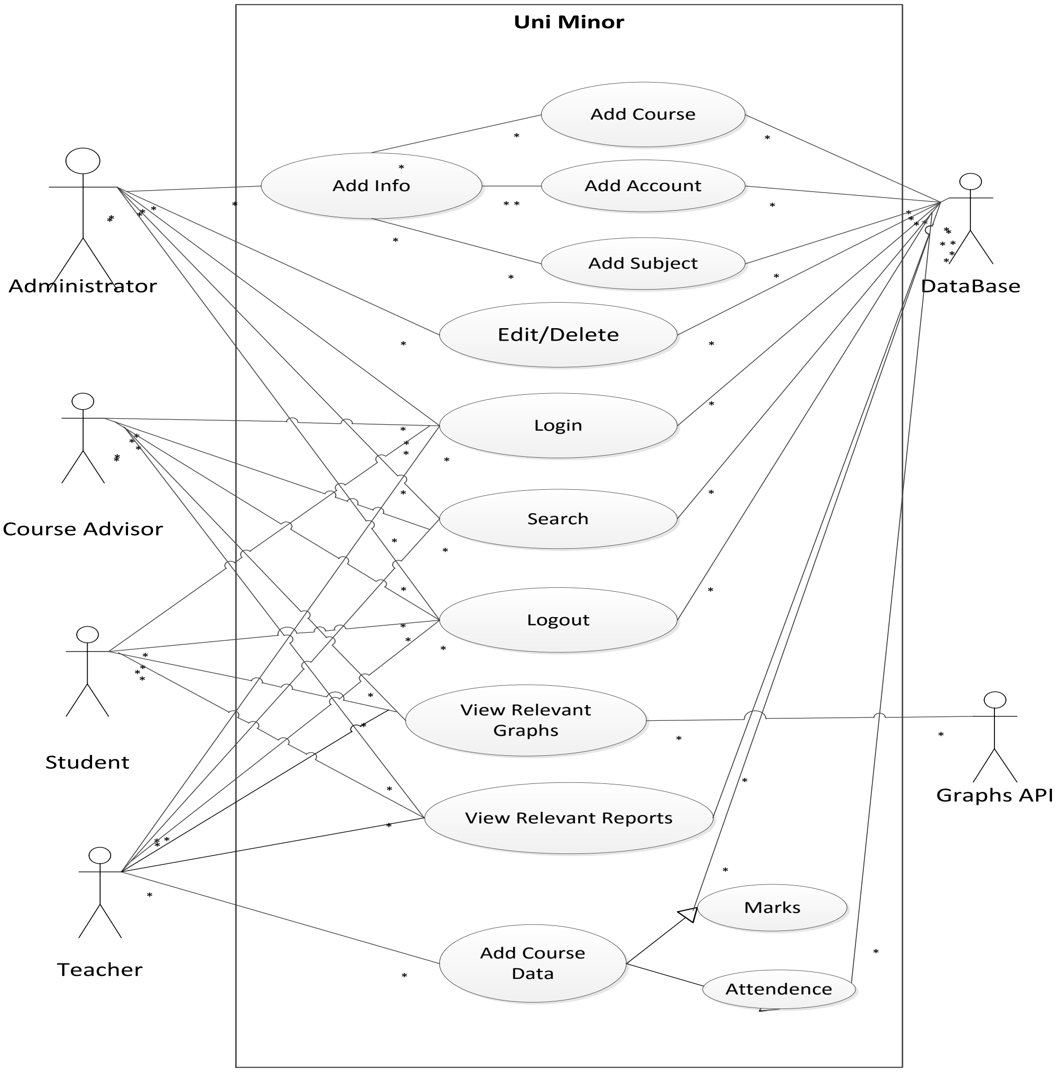


Fig 4-4: Use Case diagram for Uni-Miner

Actors

1. Primary Actor(s): Administrator, Teacher, Course Advisor, Student
2. Secondary Actor(s): Database, Graphs API

Use Cases

1. Login
2. Search
3. Add information
4. Delete information
5. Add course data
6. View Relevant reports
7. View Relevant Graphs
8. Logout

Use Case Description

Use Case No 1

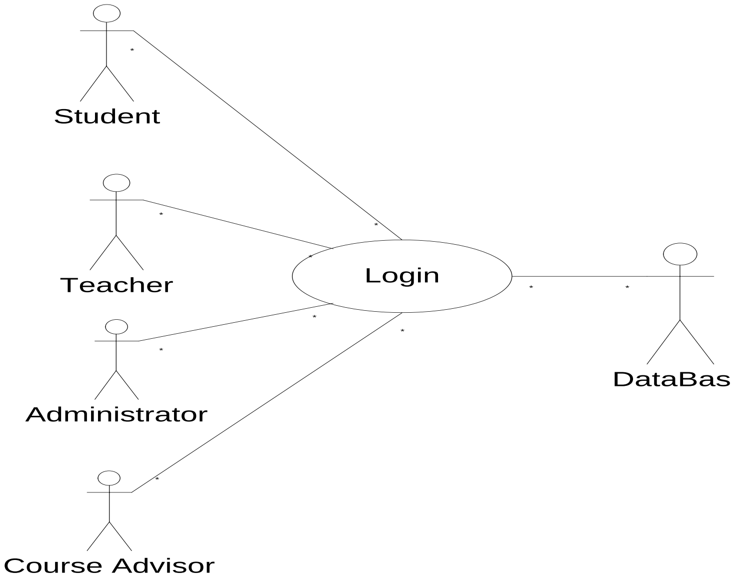


Fig 4-5: Use Case Diagram No 1

Use Case Name	Login
Primary Actor	Administrator, Teacher, Student, Course Advisor
Secondary Actor	System/Database
Normal Course	User enters his username User enters his password User clicks the login button to enter the system
Alternate Course	If the user provides incorrect username or password, the login fails and the system displays an error message
Pre-Condition	The username and password of all users must already be registered

Post Condition	The user successfully logins to the system
Extends	N/A
Include	N/A
Assumptions	The user enters correct username and password

Table 4-1: Use Case No 1

Use Case No 2

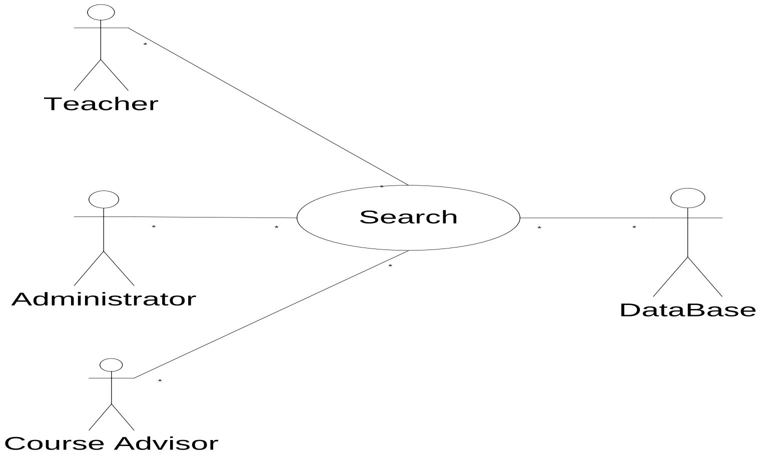


Fig 4-6: Use Case Diagram No 2

Use Case Name	Search
Primary Actor	Administrator, Teacher, Course Advisor
Secondary Actor	Database
Normal Course	user searches a student(from the database) by entering his Student No(GC 607) in the database, or scrolling through a list of registered students

Alternate Course	If the user has enters a Student No that doesn’t occur in the database, then application displays an error message “No Record Found”
Pre-Condition	The user must be logged into the application before searching for a student
Post Condition	student data is retrieved from the database
Extends	N/A
Include	N/A
Assumptions	The user enters name and course of an existing patient

Table 4-2: Use Case No 2

Use Case No 3

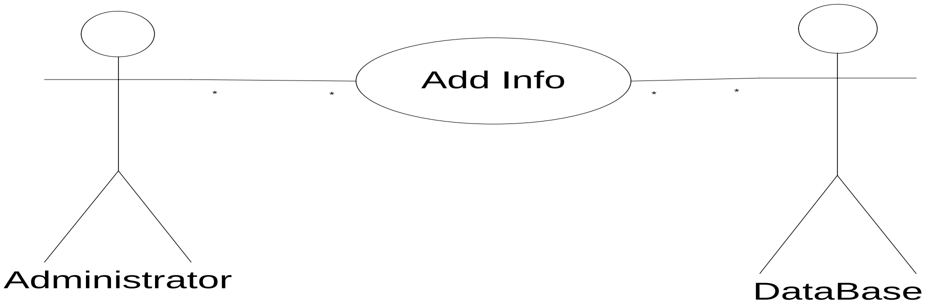


Fig 4-7: Use Case Diagram No 3

Use Case Name	Add info
Primary Actor	Administrator
Secondary Actor	Database
Normal Course	Administrator fills a form with required info and then clicks Add button
Alternate Course	Addition will not be successful if any of the required field(s) in the form will be left blank
Pre-Condition	The administrator must be logged into the system before adding a new subject, course or student.
Post Condition	New data will be successfully entered into the database
Extends	N/A
Include	Course, Subject, Student
Assumptions	The administrator fills all the required fields correctly

Table 4-3: Use Case No 3

Use Case No 4

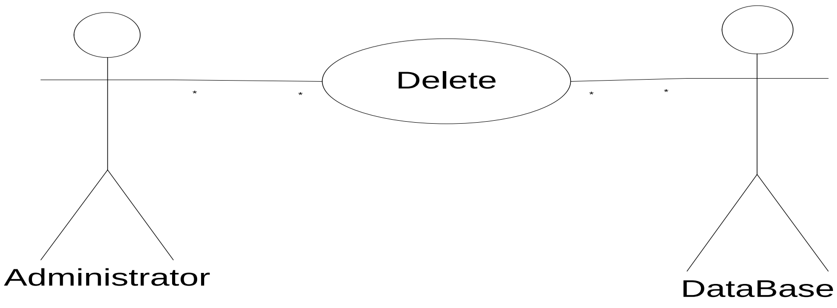


Fig 4-8: Use Case Diagram No 4

Use Case Name	Delete
Primary Actor	Administrator
Secondary Actor	Database
Normal Course	Admin successfully removes the entry and all the records from the database
Alternate Course	Admin tries to erase the data of the patient that doesn't exist in the database
Pre-Condition	The administrator must be logged into the system .
Post Condition	The system displays a " Successfully Removed" message and will navigate administrator back to main screen
Extends	N/A
Include	N/A
Assumptions	Particular Data is successfully removed from the database

Table 4-4: Use Case No 4

<u>Use Case No 5</u>

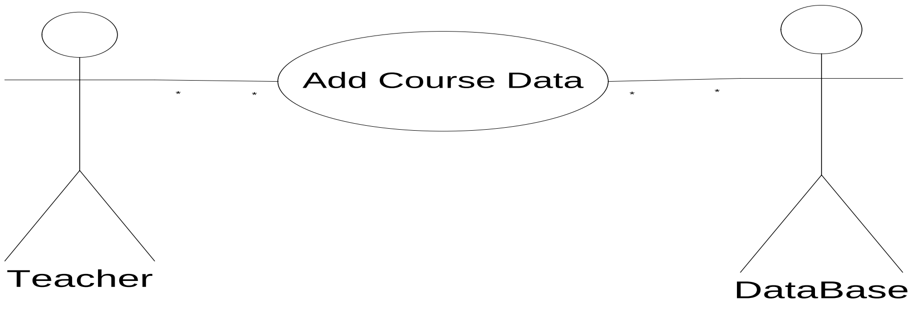


Fig 4-9: Use Case Diagram No 5

Use Case Name	Add Course Data
Primary Actor	Teacher
Secondary Actor	Database
Normal Course	Teacher selects a patient and successfully edits his/her info in the database
Alternate Course	Teacher tries to edit data of the patient that doesn't exist in the database
Pre-Condition	The teacher must be logged into the system and must already select the student.
Post Condition	The student's new info is successfully added in the database and the system then navigates user back to selected student's screen
Extends	N/A
Include	Marks, Attendance
Assumptions	The data of the selected student is successfully added into the database

Table 4-5: Use Case No 5

Use Case No 6

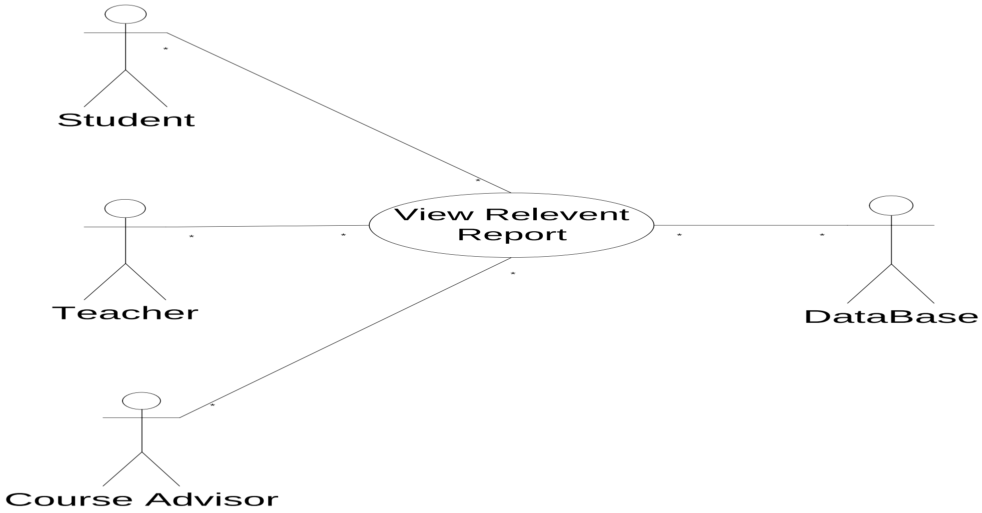


Fig 4-10: Use Case Diagram No 6

Use Case Name	View Relevant reports
Actor	Teacher ,Student, Course Advisor
Normal Course	The user selects a type from the given option on screen
Alternate Course	user tries to select a type that is not allowed
Pre-Condition	The user must be logged into the system and required records should be existing.
Post Condition	Complete detailed information is shown on the screen and options related to the selected patient.
Extends	N/A
Include	Edit/Delete, Perform Exercise and Display Progress
Assumptions	The user selects the intended type.

Table 4-6: Use Case No 6

Use Case No 7

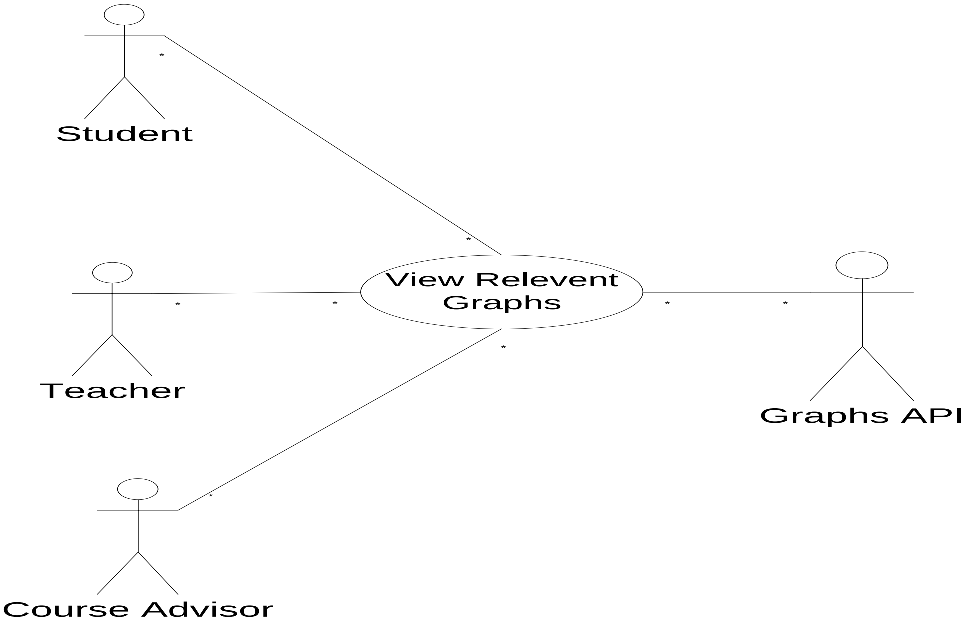


Fig 4-11: Use Case Diagram No 7

Use Case Name	View Relevant Graphs
Actor	Student ,Teacher, Course Advisor
Normal Course	User selects an option from given choices
Pre-Condition	The user must be logged into the system and has already selected a type of result.
Post Condition	The system shows the results in the form of graphs.
Extends	N/A
Include	Perform Exercise
Assumptions	The user selects graph from the available types of results.

Table 4-7: Use Case No 7

Use Case No 8

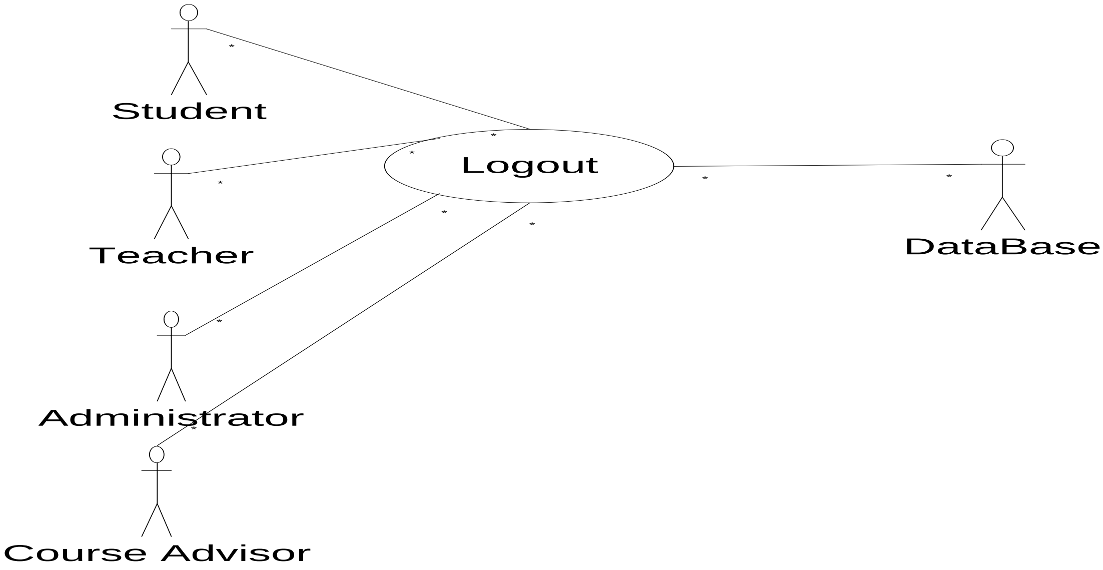


Fig 4-12: Use Case Diagram No 8

Use Case Name	Logout
Actor	Teacher, Student, Course Advisor, Administrator
Normal Course	Doctor selects a patient from main screen and selects to view his/her progress
Alternate Course	The progress will not be shown if the patient doesn’t exist in the database
Pre-Condition	The user must be logged into the system.
Post Condition	The user successfully logs out of the system.
Extends	N/A
Include	N/A
Assumptions	User has successfully performed all the tasks and system is idle

Table 4-8: Use Case No 8

4.2.2.4 Sequence Diagram

Sequence Diagrams of key use cases are mentioned below:

4.2.2.4.1 Login

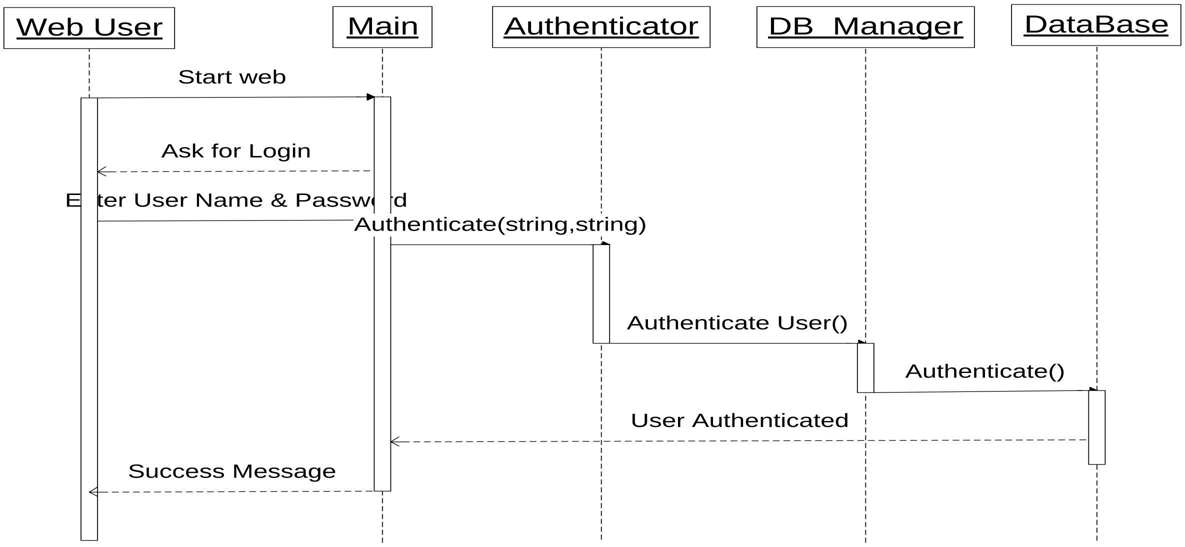


Fig 4-13: Sequence diagram for Login

4.2.2.4.2 Add Information

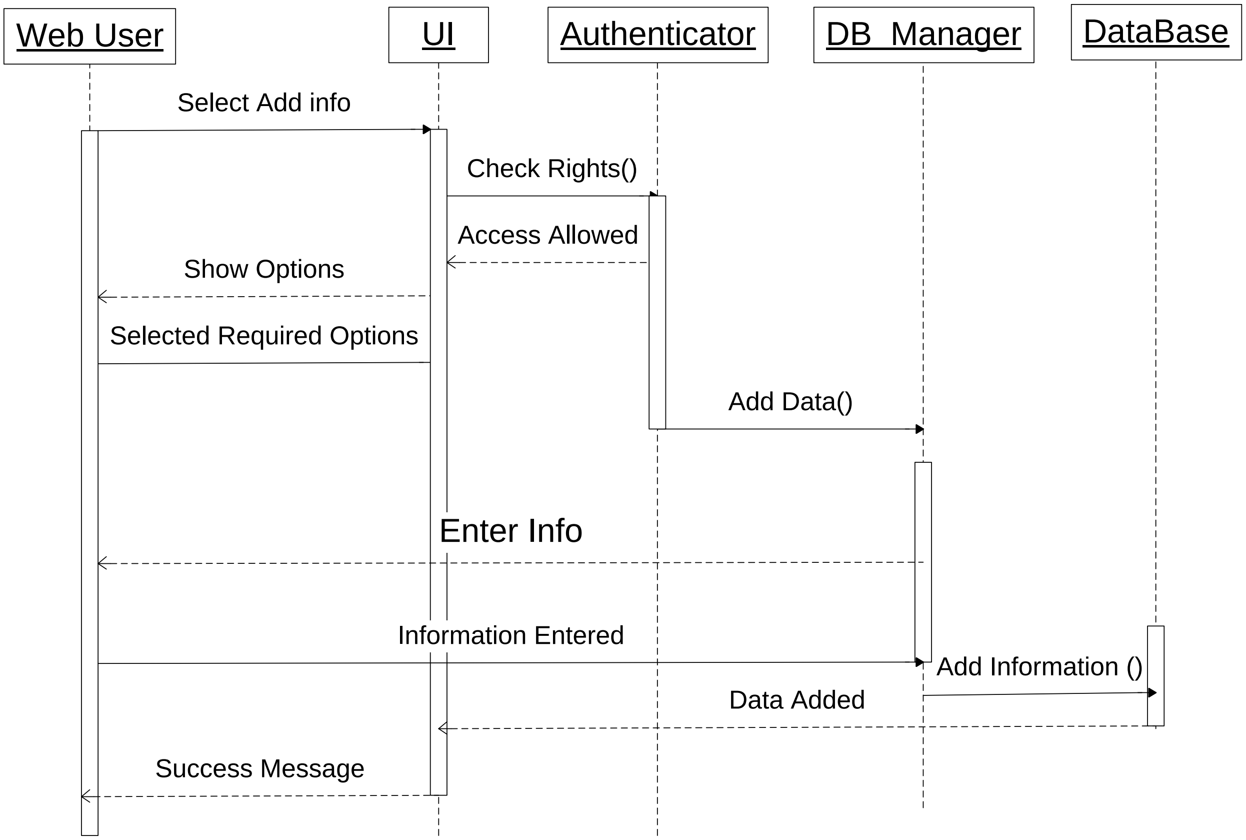


Fig 4-14: Sequence diagram for Adding Information

4.2.2.4.3 Delete Information

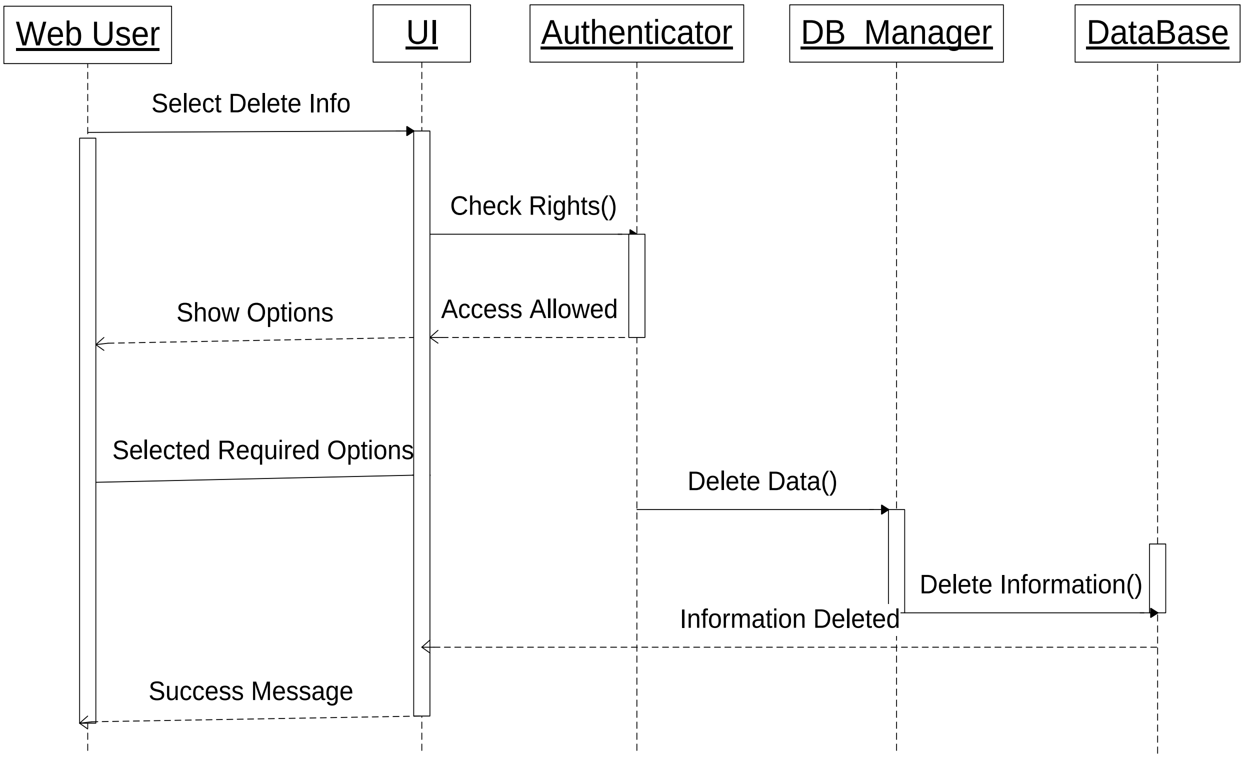


Fig 4-15: Sequence diagram for Deleting Information

4.2.2.4.4 Add Course Data

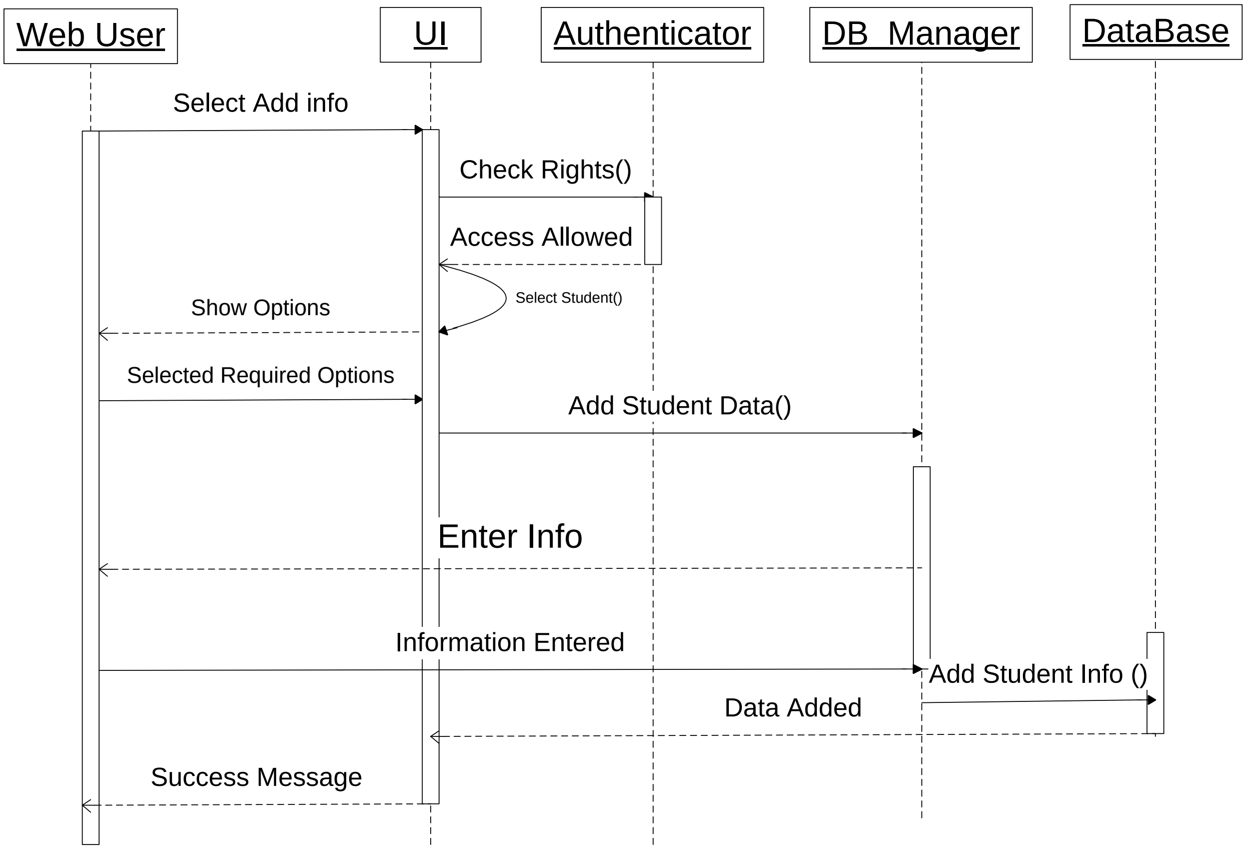


Fig 4-16: Sequence diagram for Adding Course Data

4.2.2.4.5 Show Relevant Graphs

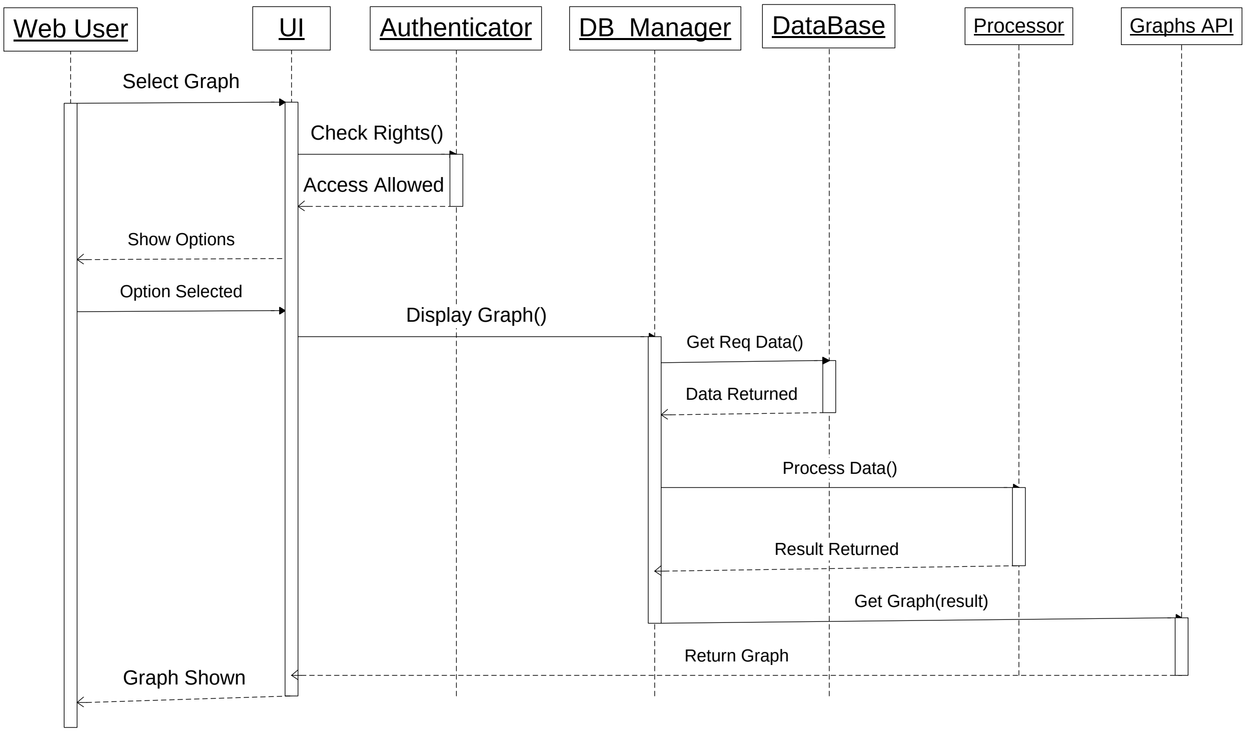


Fig 4-17: Sequence diagram for viewing graphs

4.2.2.4.6 Show Relevant Reports

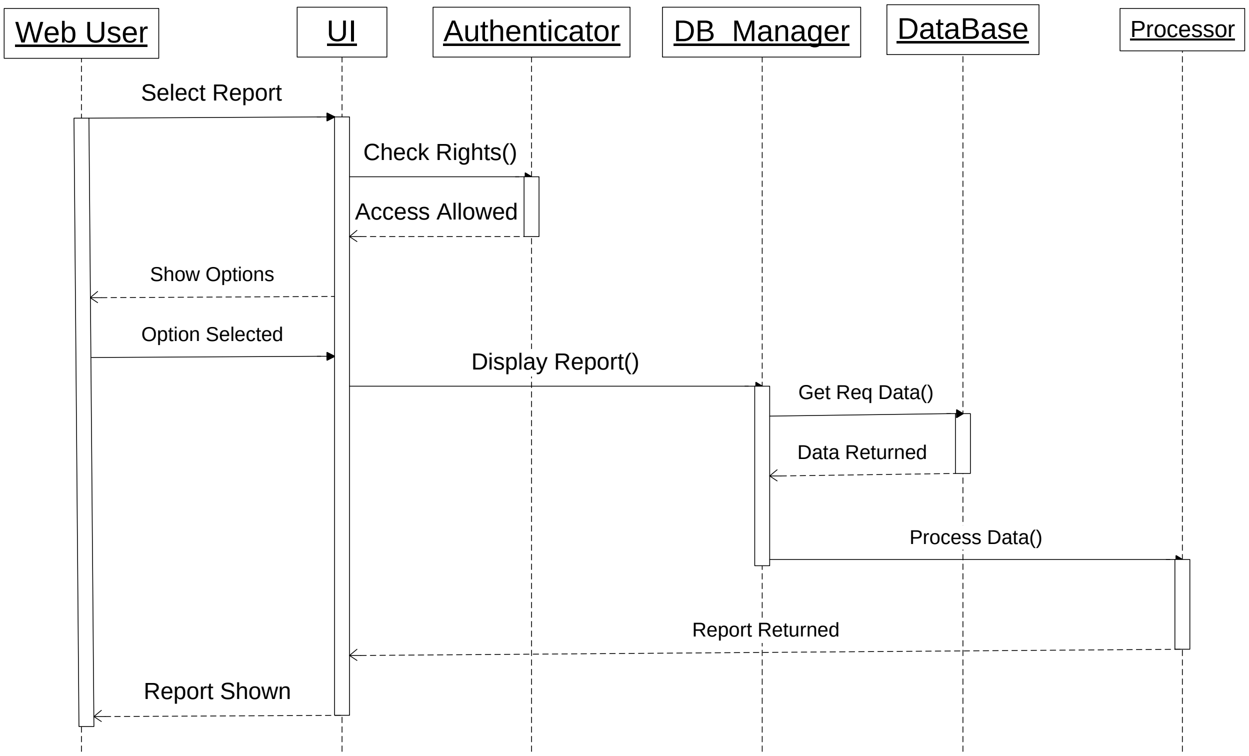


Fig 4-18: Sequence diagram for viewing reports

4.2.2.5 Dynamic View (Activity Diagram)

4.2.2.5.1 Main Activity Diagram

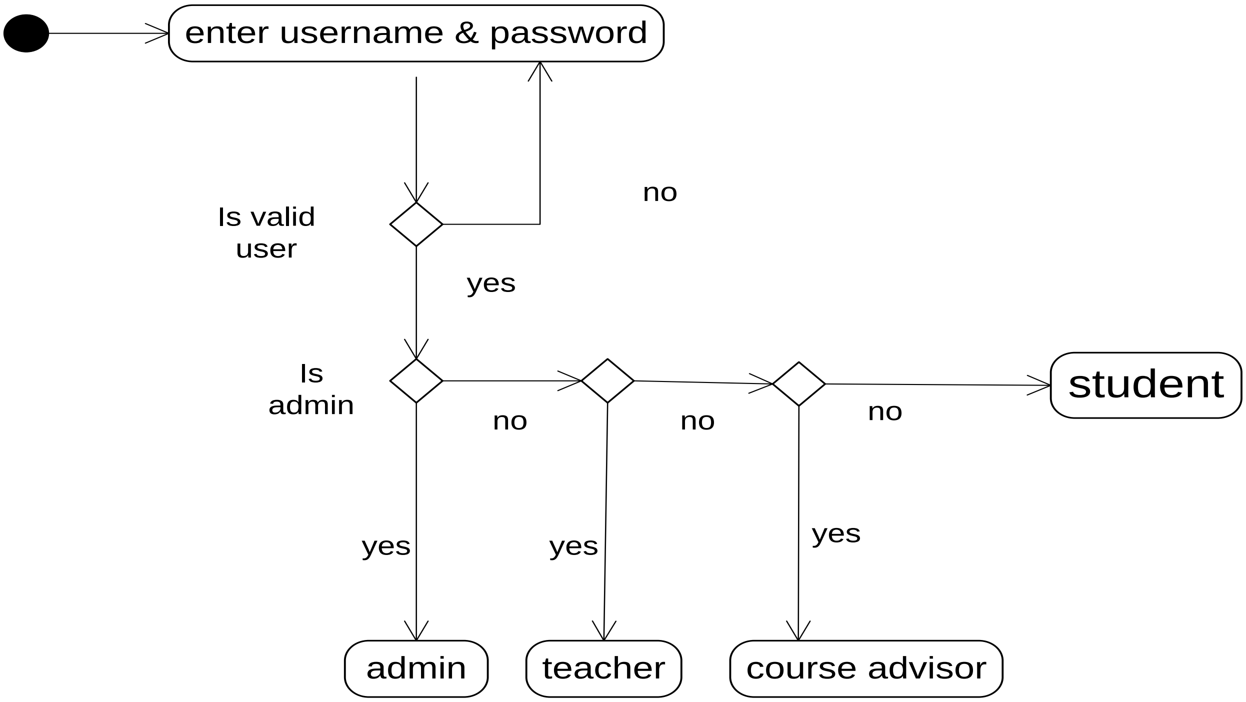


Fig 4-19: Main Activity Diagram

4.2.2.5.2 Activity Diagram for Administrator

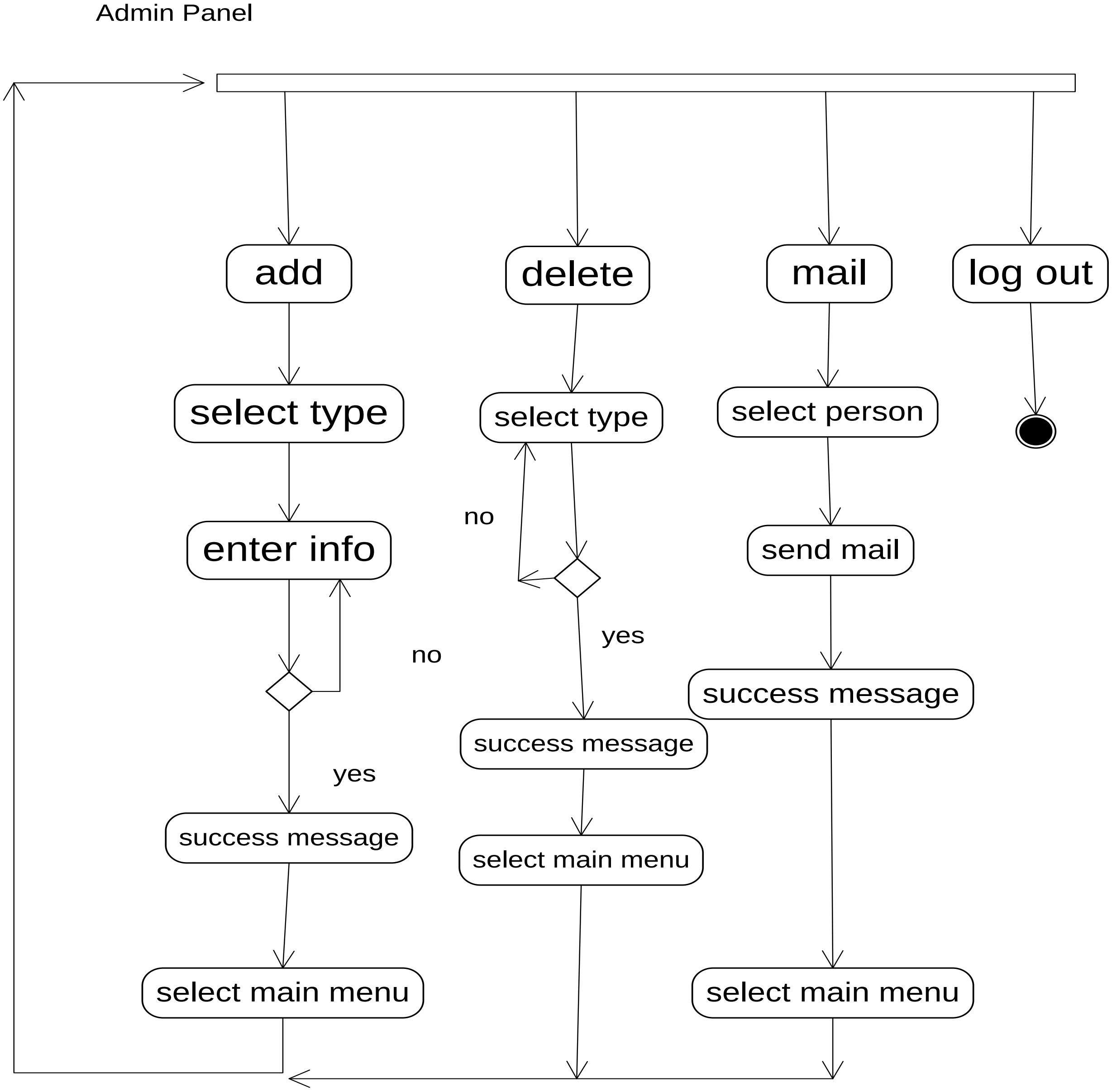


Fig 4-20: Activity Diagram for Administrator

4.2.2.5.3 Activity Diagram for Course Advisor

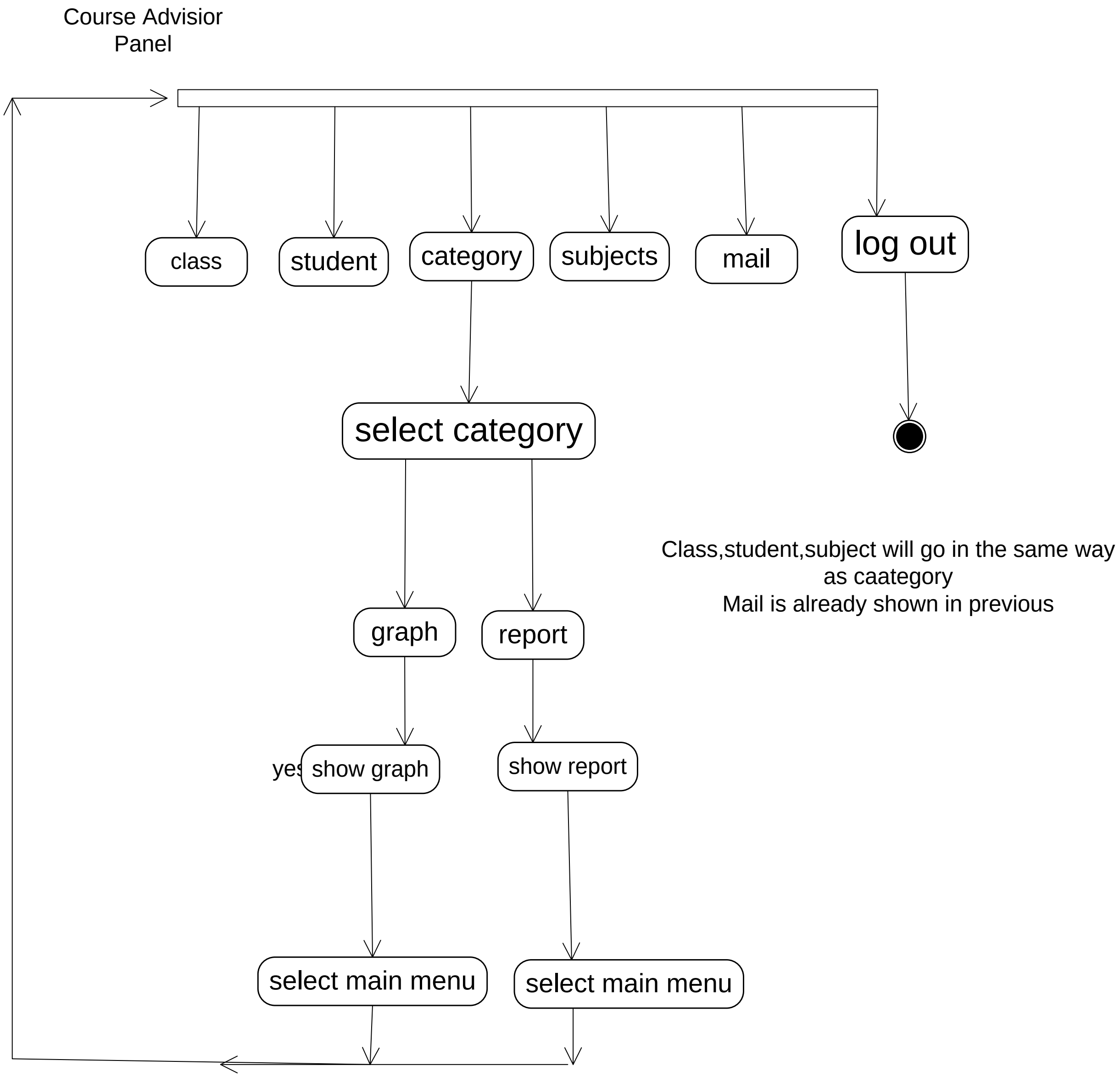


Fig 4-21: Activity diagram for Course Advisor

4.2.2.5.4 Activity Diagram for Teacher

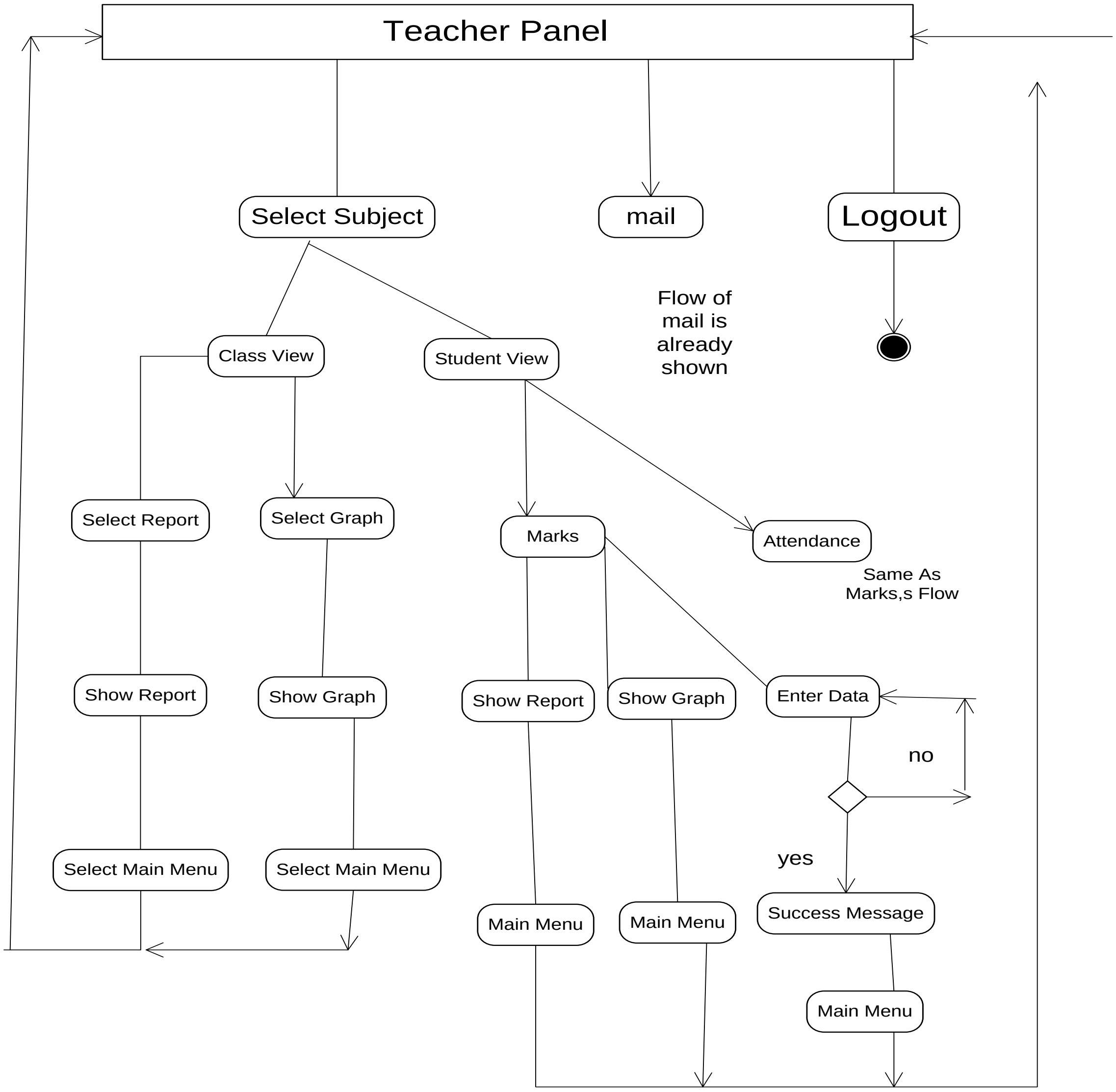


Fig 4-22: Activity diagram for Teacher

4.2.2.5.5 Activity Diagram for Student

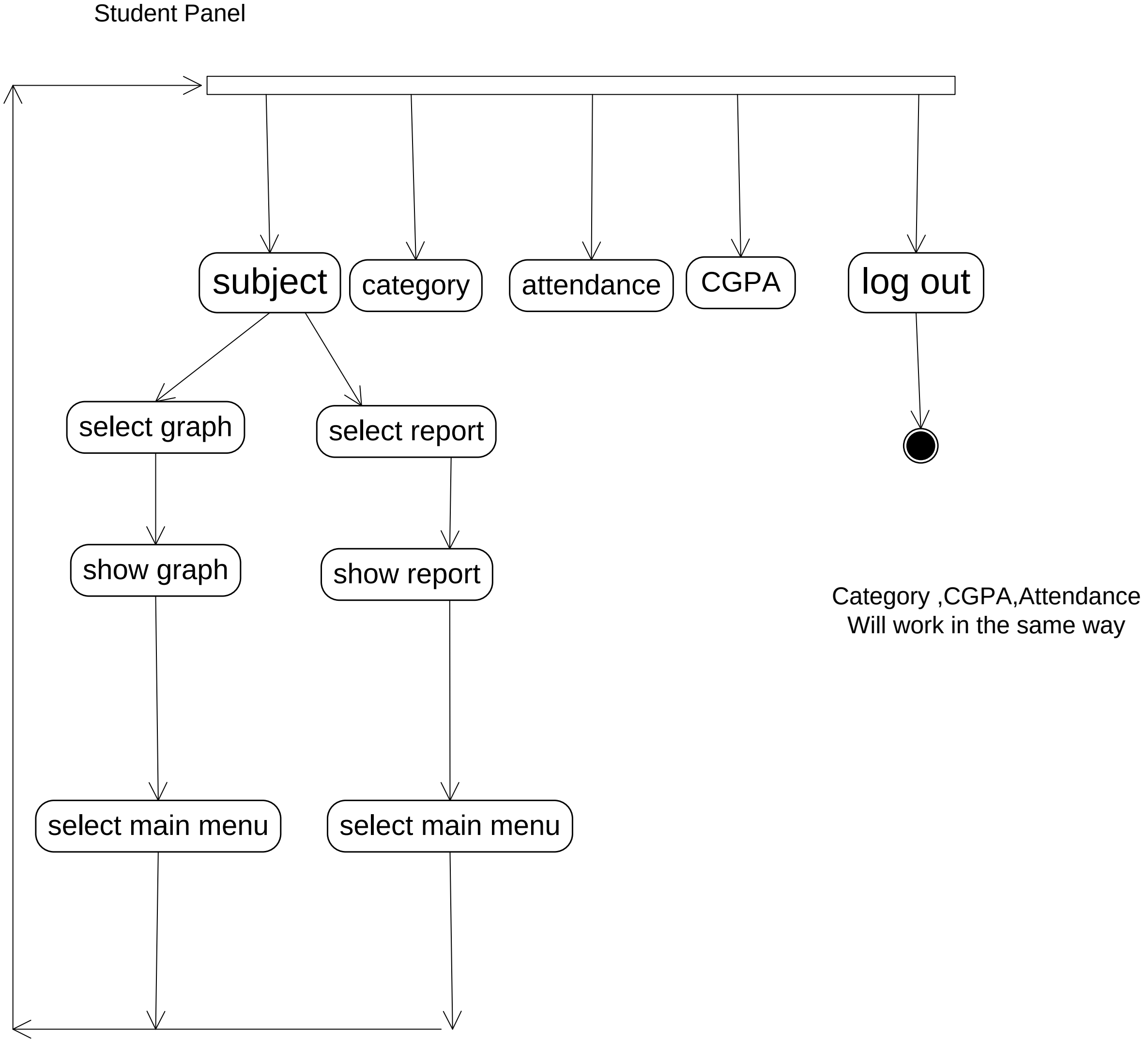


Fig 4-23: Activity diagram for Students

4.2.2.6 Implementation View (Class Diagram)

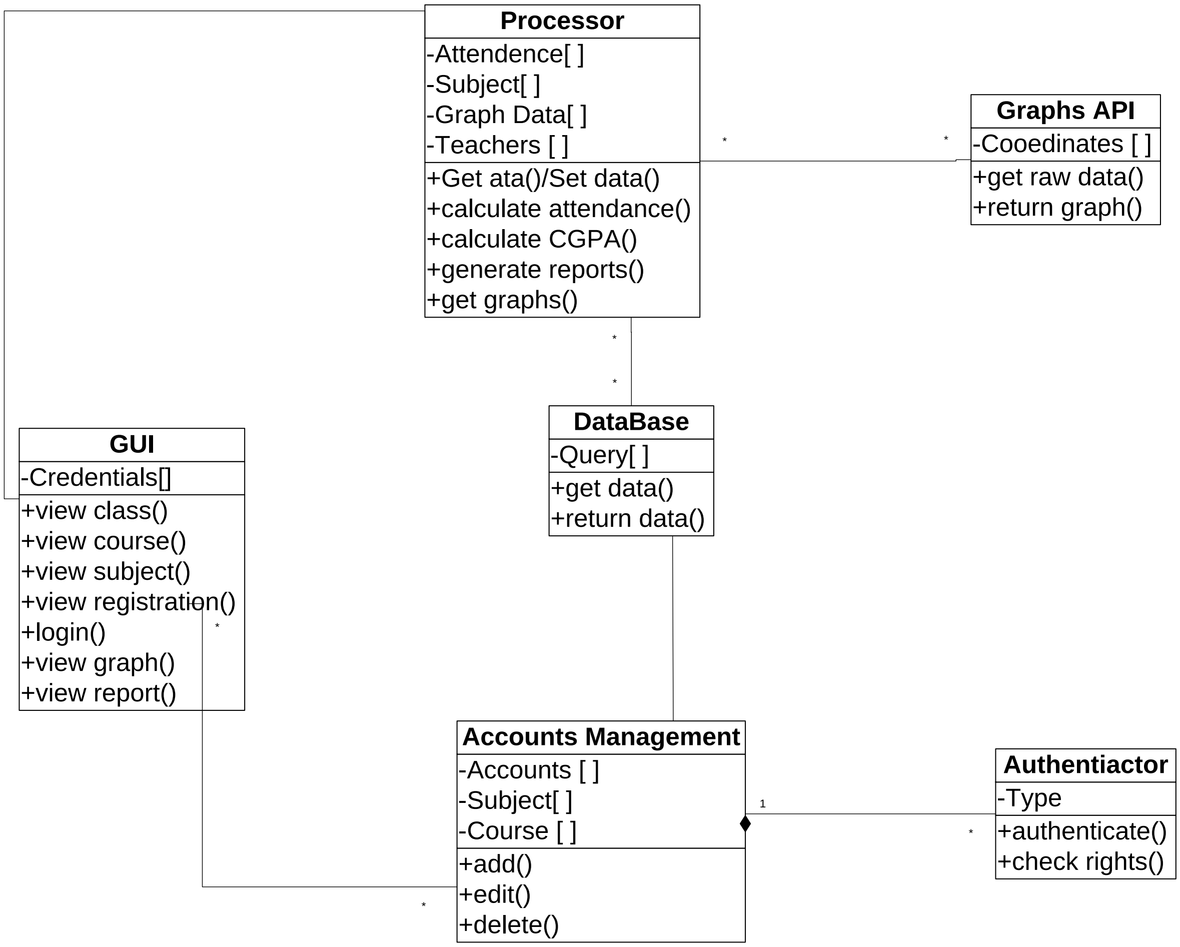


Fig 4-24: Class Diagram for Uni-Miner

Classes	Description
GUI	This is the class of system, which'll be executed first in when the system is run. It controls the view. It will be used for user interaction with system. When user will input it will get data from user and give it to relevant class. Similarly it will be used to show output to user.
Accounts Management	The class that authenticates the user via the Database Manger class. It will compare email and password provided by user with already stored in the system It has sub-class Authenticator, it is used for rights assigning.
Processor	It controls all the major activities of the system. It will get data from database and will perform calculation on that data, then it will send those results to UI for respected user. Calculations will be either regarding CGPA, Subjects, Attendance and Categories. It will interact with GUI, Database Manager and Graphs API.
Database Manager	This class communicates with Accounts Manager, Authenticator and Processor. It maintains database.
Graphs API	The class that's responsible for creating graphs. It will interact with Processor to get results.

Table 4-9: Description of Class Diagram

4.2.2.7 Structure chart

This chart shows the breakdown of the system to its lowest manageable levels. This chart basically shows the structure breakdown of the system starting from main tasks to specific sub-tasks.

4.2.3 User interface

The login screen features a logo of the college with college page, and contains two textboxes; one for username and other for the password.

After a successful login, the main screen shows up, depending on the profile type. Button on top right is always that means a user can logout anytime he wishes to leave.

Each student will only be given access to his/her own report and graphs. Teacher will be given the opportunity to enter student marks and attendance. Teacher will be allowed to check each student's performance in his/her subjects. Course advisor will be allowed to view performance of his/her respective course

Only Administrator will be shown the option to add a new course, new student or new subject. Similarly only admin will be shown the option to edit or delete.
To avoid any redundancy, back button is used for both; canceling and returning back to Main Screen.

The user will be allowed to see the performance of individual student as well of whole class. User will be allowed to see the performance in the form of Reports or Graphs.

Graphs will be further of two types such as pie graph and bar graph.

UI appears much more pleasant to the user, and also make the navigation through system quicker.

4.3 Detailed Description of Components

4.3.1 GUI

Identification	Name: GUI Location: Presentation layer of the system architecture
Type	UI component
Purpose	The user directly interacts with this component. He/she provides an input for the required action (through this component) and it displays its output respectively. This component fulfills following functional requirements (as specified in SRS Document) related to user interaction in the system: REQ-1: System should be able to notify invalid username and password if it is not found in database. REQ-2: If the invalid username or invalid password is entered, system should not load next screen and generate an error message. REQ-3: If both username and password are valid, system should load next screen. REQ-5: The application should display result of entered information in the search bar. REQ-6: If the search item does not exist in the database, system should show empty list. REQ-7: The system should be able to identify blank required fields for addition and should highlight them. REQ-8: The system should be able to open a separate panel for every individual selected student. . REQ-9: The system should be able to display add, delete, etc.

	facilities according to each user's rights. REQ-10: The system should be able to display graphs. REQ-11: The system should be able to display reports. REQ-12: The system should generate error message if the options selected by the user are invalid. REQ-13: The system should notify if student previous data does not exist. REQ-14: The system should display the whole class performance as well as individual performances. REQ-15: The system should give the option to record the remarks of the course advisor, about the progress of the student/course and it should give the option to teacher to add marks/attendance in the database. REQ-16: The system should give the option of logout on different screens so that users can logout of the system at any time.
Function	This component has two major functions; take input from the user and display all system screens. It takes input from user in form of keystrokes or other mouse events, and provides a graphical output accordingly.
Subordinates	This component has two subordinates; one is responsible for input, the other for output. The input subordinate satisfies all functional requirements (mentioned in the SRS Document) that require user input. The output subordinate satisfies all functional requirements (mentioned in the SRS Document) that provide output.
Dependencies	It interacts with Authenticator for login ,logout and for checking rights. It also interacts with processor for Graphs or Reports.

	Whereas no component depends upon it.
Interfaces	All user interfaces are part of it. The user input and output on screens will be shown using these interfaces. It will provide external interface to Authenticator and Processor in form of inputs taken from the user.
Resources	Hardware: Keyboard and mouse enable user to interact with the application. It will require a user screen to display the system. The screens will be run by using internal memory i.e.; RAM, Processor of the computer. Software: CSS and HTML to design interface for display on screen.
Data	Entered Values from User, Information String, Name String, Option Selected Integer etc.

Table 4-10: Graphical User Interface Description

4.3.2 Accounts Management

Identification	Name: Account Management
Type	Software component
Purpose	Following functional requirements mentioned in SRS are fulfilled by this component: REQ-18: The system should be able to manage accounts
Function	This component checks username & password given by user and

	compare it with those stored in the database, after that depending on the type it assigns rights to that user.
Subordinates	It has two subordinates: accounts and authenticator. Both these subordinates fulfill the REQ-18 of SRS mentioned in the purpose section above
Dependencies	This component is dependent on database, and UI is dependent on it.
Interfaces	No interface is part of it.
Resources	Software: JavaScript, PHP, MySQL.
Processing	When accounts are validated, each user is given corresponding rights, it check profile type and provide rights accordingly.

Table 4-11: Accounts Management Description

4.3.3 Processor

Identification	Name: Processor Location: Logic layer of the system architecture
Type	Software component
Purpose	Following functional requirements mentioned in SRS are fulfilled by this sub-component: REQ-4: The system should be able to show class performance REQ-6: The system should be able to show individual performance REQ-7: The system should be able to show subjects performance. REQ-8: The system should be able to show category

	performance REQ-9:The system should be able to generate report
Function	Based on the user action, it gets data from database, perform calculations accordingly. after that it return values to UI. if graphs are required it sends data to GRAPH API ,which returns graph. These graphs are sent back to UI.
Subordinates	It has four subordinates, CGPA, Subject, Attendance and Category,. and satisfy these functional requirements: REQ-6, REQ-7, REQ-8, REQ-10 and REQ-11. Whereas retrieve data will satisfy these requirements: REQ-4, REQ-8, REQ-10 and REQ-11, REQ-13 and REQ-14.
Dependencies	This dependent is dependent on Database and Graph API.
Interfaces	Student/Class/Category data is fetched from database. calculations are performed ,results are given to UI
Resources	Hardware: RAM and Processor of the system will be utilized. Software: JavaScript, MySQL , PHP
Processing	The component acts as the brain, it manages all the activities.
Data	Get Data(), Perform calculation(), Return Results()

Table 4-12: Processor Description

4.3.4 Graphs API

Identification	Name: Graphs API Location: System Logic layer of the system architecture
Type	System Logic Component
Purpose	Following functional requirements mentioned in SRS are fulfilled by this component:

	REQ-17: The system should get co-ordinates from processor. REQ-18: The system should be able to return graphs
Function	The system has to get the data and draw graphs for user to make system user-friendly. On the abstract level.
Subordinates	It has no subordinate.
Dependencies	It is dependent on processor .
Interfaces	It gives results to processor ,and then those results are sent to UI
Resources	Hardware: RAM, Processor of the system will be utilized Software: JQ charts/Microsoft BI tools
Processing	It require input values and tool
Data	x, y-coordinate are used at least .

Table 4-13: Graphs API Description

4.4 Reuse and Relationship to other products

Uni-Miner is not based on any previous systems neither it’s an extension of any other application. But it can be evolved into a bigger and more complex system with more features and functionality. Experienced developers can reuse some of the modules of the system.

The system can also be enhanced to further include observe trends in other areas, such as admission library and extra-curricular activities. In future it can also be integrated with LMS or CMS.

4.5 Design and Tradeoffs

The Uni-Miner is an interactive system which requires multiple types of user interface. Developing such systems require thorough consideration on the design factors as it might result in complexity problem. A poorly-designed system results in a system consuming more resources with very little efficiency and a slower response time which directly affects the experience of the target user. Besides this the poor designs make testing and maintenance activities difficult.

Interface of the system is distinct from the system logic. Layered architecture is used to isolate system logic from the user interface. It can be modeled using Multitier Layered Architecture consisting of three layers i.e.; presentation, system logic and database. Presentation layer corresponds to elements of the user interface such as text, checkbox item etc., and system logic layer controls the communication of data between the presentation and the database layer, and is the part where the main logic, user actions and working of the system is defined. In general, it controls the complete behavior of the system, while the database layer is responsible for handling and storing of the processed data.

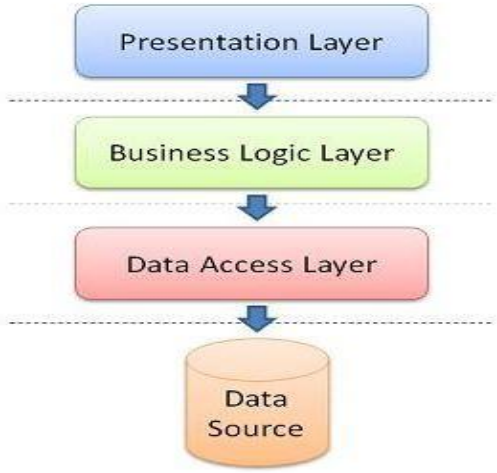


Fig 4-25: Layer Diagram

Presentation layer consists of user interfaces of the system, with which the users will be interacting. All reports, graphs, buttons and form elements (textbox, radio buttons etc.) used in the application will come under view.

System logic layer includes all the algorithms that will be implemented, handles whatever action is done on the presentation layer. It will process the information gathered from the presentation and the database layer.

It provides a communication channel between both these layers and also controls the data flow between them. It also controls the information that is to be displayed and the data that is to be stored.

Database layer is only responsible for storing the processed information into the database. It can contain multiple databases in which data is stored in form of tables. It returns some result to the system logic layer upon request.

4.5.1 For Adding Information

```
If Admin is Login
begin
        Show Add new course
        Show Add new subject
        Show Add new student
end
else
        print ‘Invalid Username/Password’
```

4.5.2 For Deleting Information

```
If Admin is already logged in
begin
        Remove student
        Remove course
      Remove subject
end
else
        print ‘Invalid Username or Password’
```

4.5.3 For Viewing Graphs

```
If user is already logged in
begin
        select student/course/category
        select graph type
        Get required data from database
        Perform calculation
        Show result in graphic form
end
else
        print ‘Sorry you are not allowed’
```

4.5.4 For Viewing Reports

```
If user is already logged in
begin
        select course/student/subject
        select report
        get data from database
        perform calculation
        show report
end
else
        print ‘Sorry you are not allowed’
```

4.5.5 For Adding Course Data

```
if teacher is already logged in
begin
        select course/student/subject
        select marks/attendance
        enter data
        show “success message’’
end
else
print ‘Please Recheck values’
```

Chapter 5: Project Analysis and Evaluation

The following test case table checks the login of the administrator.

Test Case Number	1
Test Case Name	Testing admin login
Description	In this test, we will test that whether the admin login is working or not. It will depend on the credentials of the admin user.
Preconditions	Web application should be in running process Admin should have account
Input	Click on Sign In button
Steps	Input username and password for admin Click Sign In button
Expected output	Admin gets logged in
Results	Admin gets logged in

Table 5-1: Testing Admin Login

The following test case table checks the tab button of adding new course.

Test Case Number	2
Test Case Name	Testing the tab for addition of new course
Description	This test will prove that does the web application allows us to add any new course.
Preconditions	Web application should be in running process Admin should be logged in
Input	Click on Add New Course Button
Steps	Open the tab on right side under logged on user Click on Add New Course Button

Expected output	Add new course page will be shown
Results	Add new course page will be shown

Table 5-2: Testing the tab for addition of new course

The following test case table checks the addition of new course.

Test Case Number	3
Test Case Name	Testing the addition of new course
Description	In this test we will test that does the web application adds the new course to database.
Preconditions	Web application should be in running process Admin should be logged in Page for add new course should be open
Input	Input the details
Steps	Enter the details mentioned on the form Click the register button
Expected output	The course will be added to the database
Results	The course is added to the database

Table 5-3: Testing the addition of new course

The following test case table checks the tab button of adding new subject.

Test Case Number	4
Test Case Name	Testing the tab for addition of new subject
Description	In this test we will test that does the web application allows us to add any new subject.

Preconditions	Web application should be in running process Admin should be logged in
Input	Click on Add New Subject Button
Steps	Open the tab on right side under logged on user Click on Add New Subject Button
Expected output	Add new subject page will be shown
Results	Add new subject page will be shown

Table 5-4: Testing the tab for addition of new subject

The following test case table checks the addition of new subject.

Test Case Number	5
Test Case Name	Testing the addition of new subject
Description	In this test we will test that does the web application add the new subject to the database.
Preconditions	Web application should be in running process Admin should be logged in Page for add new subject should be open
Input	Click plus sign at upper right corner Input the details
Steps	Enter the details mentioned on the form Click the save button on upper right corner
Expected output	The subject will be added in the mentioned course
Results	The subject will be added in the mentioned course

Table 5-5: Testing the addition of new subject

The following test case table checks the tab button of adding new teacher.

Test Case Number	6
Test Case Name	Testing the tab for addition of new teacher
Description	In this test we will test that does the web application allows us to add any new teacher.
Preconditions	Web application should be in running process Admin should be logged in
Input	Click on Register New Teacher Button
Steps	Open the tab on right side under logged on user Click on Register New Teacher Button
Expected output	Register new teacher page will be shown
Results	Register new teacher page will be shown

Table 5-6: Testing the tab for addition of new teacher

The following test case table checks the addition of new teacher.

Test Case Number	7
Test Case Name	Testing the addition of new teacher
Description	In this test we will test that does the web application register the teacher in the database.
Preconditions	Web application should be in running process Admin should be logged in Page for register new teacher should be open
Input	Input the details
Steps	Enter the details mentioned on the form Click the register button

Expected output	The teacher will be added
Results	The teacher will be added

Table 5-7: Testing the addition of new teacher

The following test case table checks the tab button of assigning new subjects.

Test Case Number	8
Test Case Name	Testing the tab for assigning of subjects
Description	In this test we will test that does the web application allows us to assign subjects to the teachers.
Preconditions	Web application should be in running process Admin should be logged in
Input	Click on Assign Subject Button
Steps	Open the tab on right side under logged on user Click on Assign Subject Button
Expected output	Assign subject page will be shown
Results	Assign subject page will be shown

Table 5-8: Testing the tab for assigning of subjects

The following test case table checks the assigning of new subjects.

Test Case Number	9
Test Case Name	Testing the assigning of subjects
Description	In this test we will test that does the web application assigns the subjects to the teachers.
Preconditions	Web application should be in running process

	Admin should be logged in Page for assign subject should be open
Input	Input the course details
Steps	Enter the details mentioned on the form Click the Submit button
Expected output	The subject will be assigned
Results	The subject will be assigned

Table 5-9: Testing the assigning of subjects

The following test case table checks the teacher login.

Test Case Number	10
Test Case Name	Testing teacher login
Description	In this test, we will test that whether the teacher login is working or not. It will depend on the credentials assigned to the teacher.
Preconditions	Web application should be in running process Teacher should have account
Input	Click on Sign In button
Steps	Input username and password assigned to the teacher Click Sign In button
Expected output	Teacher gets logged in
Results	Teacher gets logged in

Table 5-10: Testing teacher login

The following test case table checks the tab button of assigned subjects to the teachers.

Test Case Number	11
Test Case Name	Testing the tab for assigned subjects of the teacher
Description	In this test, we will test that whether the web application shows all the subjects assigned to any particular teacher
Preconditions	Web application should be in running process Teacher should be logged in
Steps	Click the tab on left side named as Subjects
Expected output	Subjects assigned to a teacher will be shown
Results	Subjects assigned to a teacher will be shown

Table 5-11: Testing the tab for assigned subjects of the teacher

The following test case table checks the addition/updating of marks..

Test Case Number	12
Test Case Name	Testing the addition of marks
Description	In this test, we will test the addition of marks of the subjects that is assigned to any teacher.
Preconditions	Web application should be in progress Teacher should be logged in Teacher should have selected any subject
Input	Click the subject for which the marks have to be entered
Steps	Select the subject Enter details for example marks (category wise) and attendance Click save button on upper left side

Expected output	The marks for the subject will be added and saved
Results	The marks for the subject will be added and saved

Table 5-12: Testing the addition of marks

The following test case table checks the predictions based on provided results.

Test Case Number	13
Test Case Name	Testing the overall prediction based on current results
Description	Here we will test that whether the predictions shown are based on current results.
Preconditions	Web application should be in progress User should be logged in User should have selected any subject
Input	Click the subject for which the prediction is to be shown
Steps	Select the subject Click the Prediction based on current results tab
Expected output	Prediction based on current results will be shown
Results	Prediction based on current results will be shown

Table 5-13: Testing the overall prediction based on current results

The following test case table checks the predictions based on subject history.

Test Case Number	14
Test Case Name	Testing the overall prediction based on subject history
Description	In this, we will test that whether the predictions are based on subject history.

Preconditions	Web application should be in progress User should be logged in User should have selected any subject
Input	Click the subject for which the prediction is to be shown
Steps	Select the subject Click the Prediction based on subject history tab
Expected output	Prediction based on subject history will be shown
Results	Prediction based on subject history will be shown

Table 5-14: Testing the overall prediction based on subject history

The following test case table checks the students' login.

Test Case Number	15
Test Case Name	Testing student login
Description	In this test, we will test that whether the student login is working or not. It will depend on the credentials of the student user.
Preconditions	Web application should be in running process Student should have an account
Input	Click on Sign In button
Steps	Input username and password assigned to the students Click Sign In button
Expected output	Student gets logged in
Results	Student gets logged in

Table 5-15: Testing student login

The following test case table checks the tab button of overviewing CGPA and Attendance.

Test Case Number	16
Test Case Name	Testing the tab for overview of CGPA and Attendance
Description	In this test, we will test that whether the web application shows the CGPA and attendance options for the student.
Preconditions	Web application should be in running process Student should be logged in
Steps	Click the tab on left side named as overview
Expected output	CGPA and Attendance options will be shown
Results	CGPA and Attendance options will be shown

Table 5-16: Testing the tab for overview of CGPA and Attendance

The following test case table checks the options for CGPA and Attendance.

Test Case Number	17
Test Case Name	Testing the CGPA and Attendance options
Description	In this test, we will test that whether the web application shows the CGPA and attendance of the student.
Preconditions	Web application should be in running process Student should be logged in
Steps	Click the CGPA/Attendance tab
Expected output	CGPA and Attendance options will be shown
Results	CGPA and Attendance are not shown still.

Table 5-17: Testing the CGPA and Attendance options

The following test case table checks the tab button of subjects.

Test Case Number	18
Test Case Name	Testing the tab named Subjects
Description	In this test, we will test that whether the web application shows all the subjects previous/current of the student.
Preconditions	Web application should be in running process Student should be logged in
Steps	Click the tab on left side named as subjects
Expected output	All the subjects will be shown (current/previous)
Results	All the subjects will be shown (current/previous)

Table 5-18: Testing the tab named Subjects

The following test case table checks the details for any particular subject.

Test Case Number	19
Test Case Name	Testing the subject details
Description	In this test, we will test that whether the web application shows the details of the subject selected by the student.
Preconditions	Web application should be in running process Student should be logged in Subjects should be already opened by the Subjects tab
Steps	Click the subject for which you want to see the details.
Expected output	Details for the selected subject will be shown
Results	Details for the selected subject will be shown

Table 5-19: Testing the subject details

The following test case table checks the prediction based on current results for any particular selected subjects.

Test Case Number	20
Test Case Name	Testing the prediction based on current results for selected subjects
Description	Here we will test that whether the predictions are shown for the selected subject based on current results.
Preconditions	Web application should be in progress User should be logged in User should have selected any subject
Input	Click the subject for which the prediction is to be shown
Steps	Select the subject Click the Prediction based on current results tab
Expected output	Bar chart of the selected subjects' result will be shown
Results	Bar chart of the selected subjects' result will be shown

Table 5-20: Testing the prediction based on current results for selected

The following test case table checks the predictions for any selected subject with reference to the subject history.

Test Case Number	21
Test Case Name	Testing the prediction based on subject history for selected subjects
Description	Here we will test that whether the predictions are shown for the selected subject based on subject history
Preconditions	Web application should be in progress User should be logged in

	User should have selected any subject
Input	Click the subject for which the prediction is to be shown
Steps	Select the subject Click the Prediction based on subject history tab
Expected output	Bar chart of the selected subjects’ result will be shown
Results	Bar chart of the selected subjects’ result will be shown

Table 5-21: Testing the prediction based on subject history for selected subjects

The following test case table checks the system logout.

Test Case Number	22
Test Case Name	Testing logout
Description	In this test, we will test that whether the logout is working or not.
Preconditions	Web application should be in running process User should be logged in already.
Input	Click Log out
Steps	Open users tab Click Logout button
Expected output	User gets logged out
Results	User gets logged out

Table 5-22: Testing logout

Chapter 6: Future work

The same technique can be used in future to evaluate the results in a broader manner. In future the predications might be done not only for students but for any institute and help users check the performance of employees also.

Not only performance in academics but discipline, personality and physical analysis may also be done. By this the organizations and the employees themselves can be well aware of the loop holes in their performance and can improve themselves.

Chapter 7: Conclusion

Uni-Miner can be a valuable resource for all university users that need information to provide support for:

1. Performance Management
2. Decision Support at strategic, tactical and operational level
3. Strategic Planning
4. Compliance Reporting
5. Day to day operations

Glossary

DB	Database
Mediator Pattern	Defines an object that encapsulates how a set of objects interact
WBS	Work Breakdown Structure
GUI	Graphical User Interface
UML	Unified Modelling Language

References

[1] Microsoft BI suite
https://powerbi.microsoft.com/en-us/

[2] Interactive charts for browsers, Google Charts, Controls and Dashboards
https://developers.google.com/chart/?hl=en

[3] Mediator Pattern
www.dofactory.com/net/design-patterns

[4] Common charts, Advanced Charts
http://www.arcplan.com/en/solutions/enterprise/common-charts/

[5] BI Graphs: Graph 1, Visualizing Data: Para 3
http://www.altova.com/mobiletogether/charts.html

[6] "Margaret Rouse", Business Intelligence and Data analyzing
http://searchdatamanagement.techtarget.com/definition/business-intelligence

[7] "David Euwings", BI Dashboards: Quickly and Effectively Communicating Actionable Information
http://www.arcplan.com/en/blog/2011/08/dashboard-charts-graphs-which-is-right for-your-data/

Appendix-A

User Manual

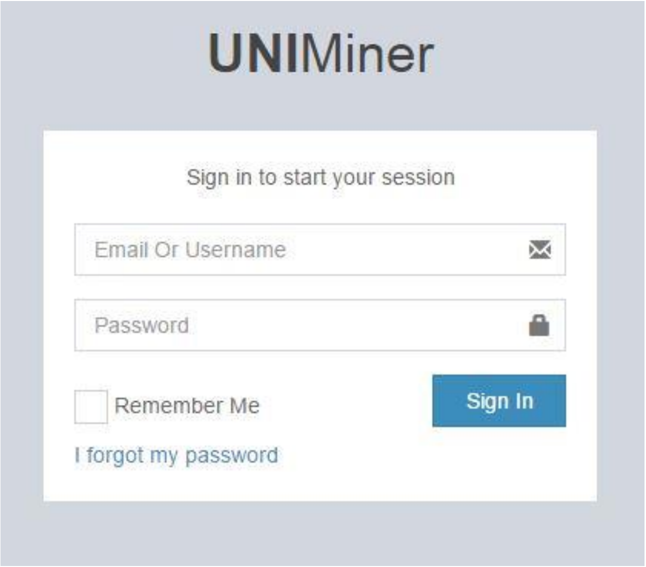
This Software User Manual applies to Uni-Miner Software. The purpose of this user manual is to assist the user in using the Uni-Miner software. Please read this manual thoroughly before using the software to ensure safe and proper use. Descriptions are based on the default settings.

Login

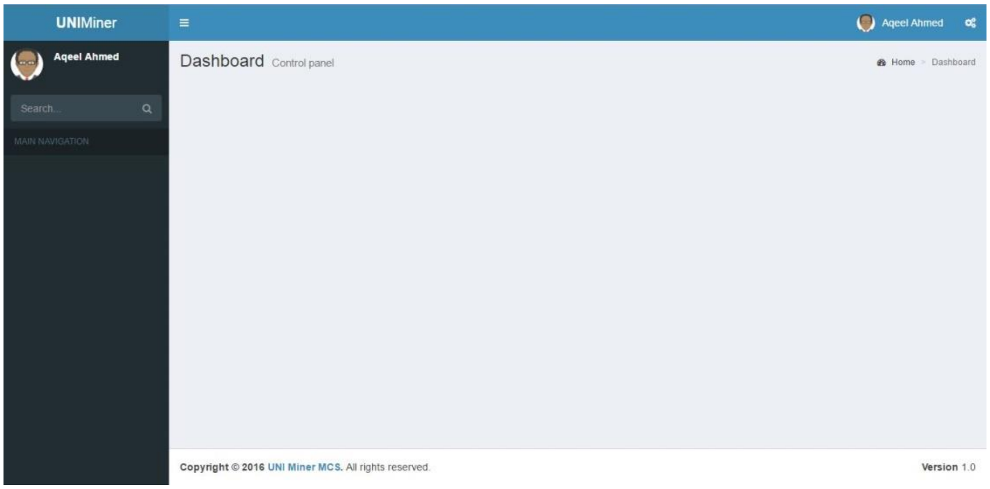
The user can have four (4) types of login.

1. Administrator
2. Head of Department/Course Advisor
3. Teacher
4. Student

Only the administrator has the privilege to set the credentials for all the categories. The user can only login through the credentials provided by the administrator.

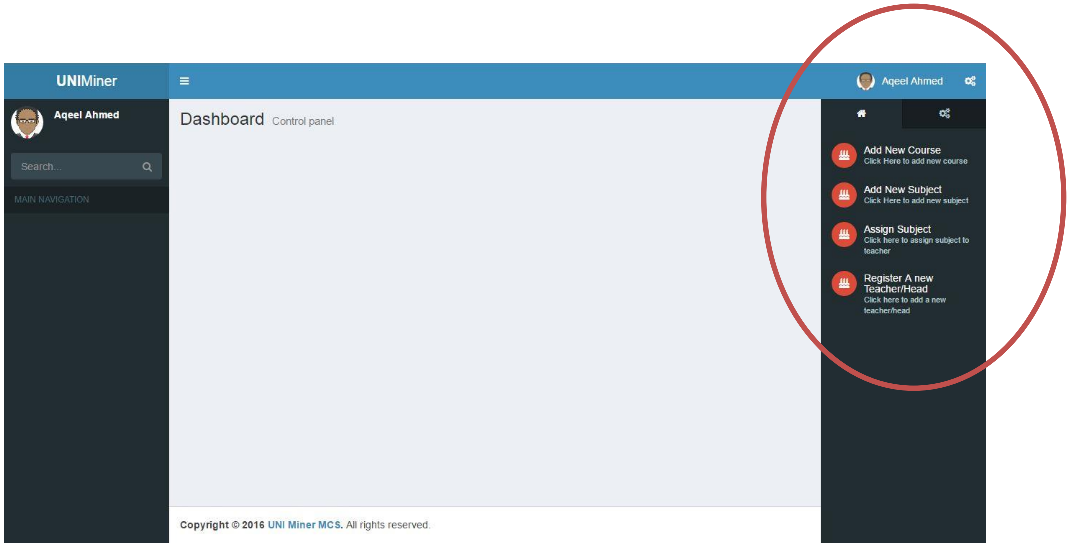


Getting Started (Administrator)



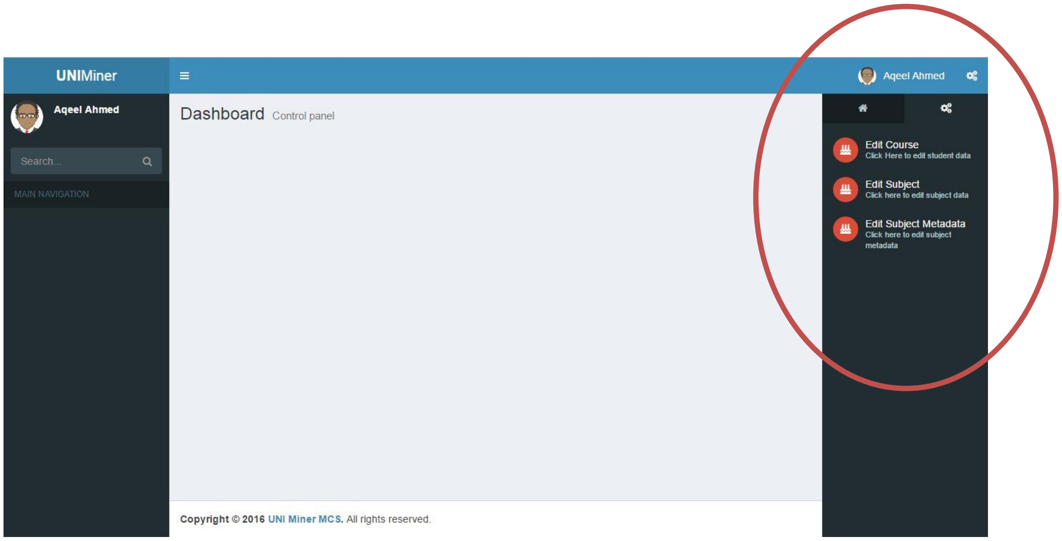
Admin Rights include:

1. Add New Course
2. Add New Subject
3. Assign Subject
4. Add Teacher

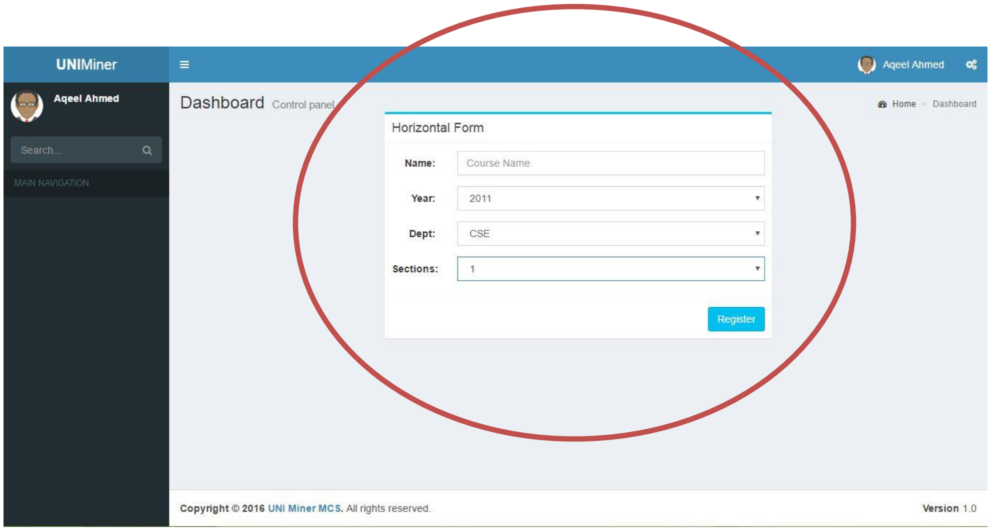

UNIMiner
Aqeel Ahmed
Search...
MAIN NAVIGATION
Dashboard Control panel
Aqeel Ahmed
Add New Course
Click Here to add new course
Add New Subject
Click Here to add new subject
Assign Subject
Click here to assign subject to teacher
Register A new Teacher/Head
Click here to add a new teacher/head
Copyright © 2016 UNI Miner MCS. All rights reserved.

Admin can also:

1. Edit Course
2. Edit Subject
3. Edit Subject Metadata

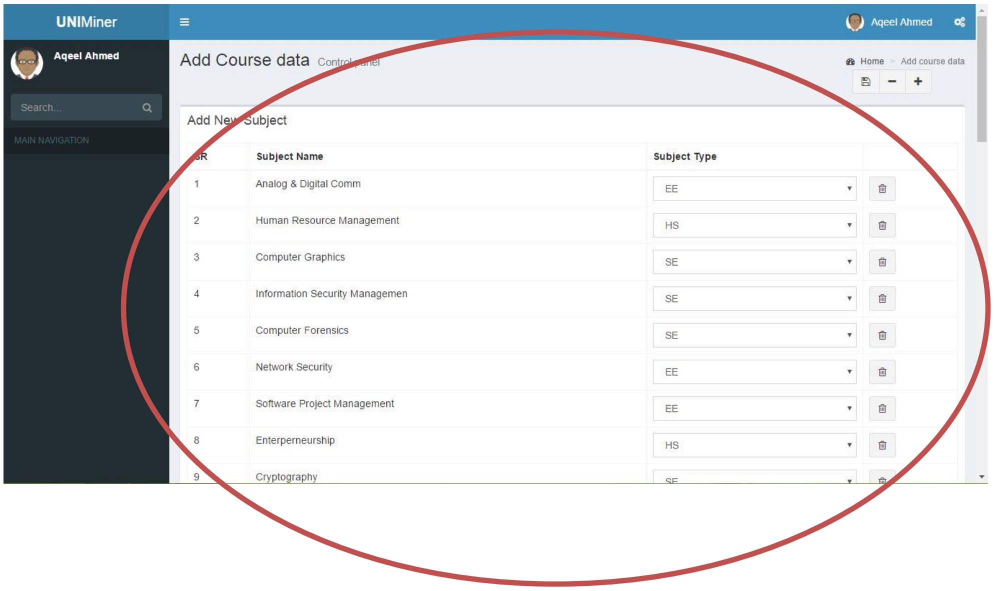

UNIMiner
≡
Aqeel Ahmed
Aqeel Ahmed
Search...
MAIN NAVIGATION
Dashboard Control panel
Edit Course
Click Here to edit student data
Edit Subject
Click here to edit subject data
Edit Subject Metadata
Click here to edit subject metadata
Copyright © 2016 UNI Miner MCS. All rights reserved.

Add New Course



Name:	Course Name
Year:	2011
Dept:	CSE
Sections:	1

Add New Subject

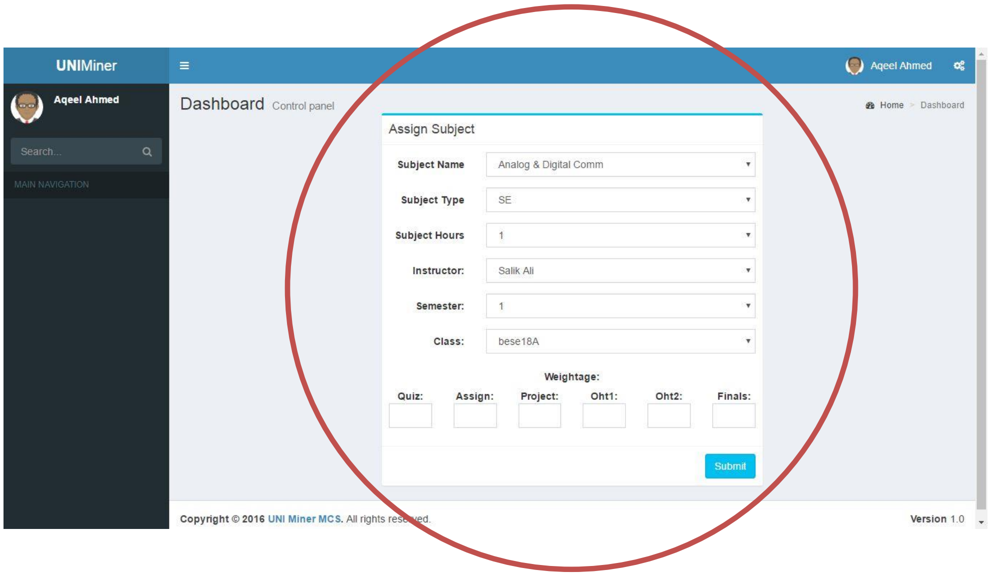


Add Course data Control panel

Add New Subject

SR	Subject Name	Subject Type
1	Analog & Digital Comm	EE
2	Human Resource Management	HS
3	Computer Graphics	SE
4	Information Security Managemen	SE
5	Computer Forensics	SE
6	Network Security	EE
7	Software Project Management	EE
8	Enterperneurship	HS
9	Cryptography	SE

Assign Subject



Dashboard Control panel

Assign Subject

Subject Name: Analog & Digital Comm

Subject Type: SE

Subject Hours: 1

Instructor: Salik Ali

Semester: 1

Class: bese18A

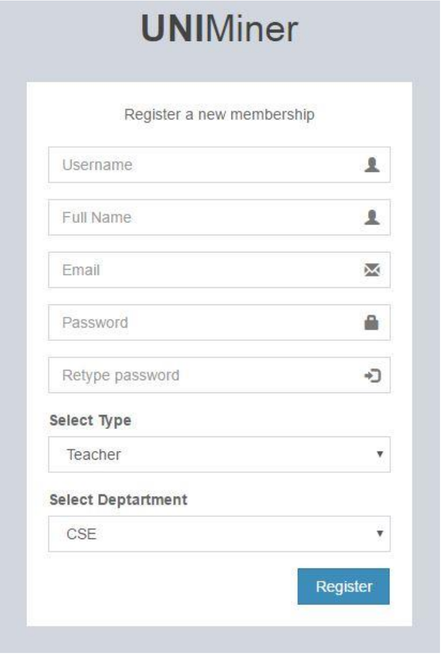
Weightage:

Quiz: Assign: Project: Oht1: Oht2: Finals:

Copyright © 2016 UNI Miner MCS. All rights reserved.

Version 1.0

Add New Teacher



UNIMiner

Register a new membership

Username

Full Name

Email

Password

Retype password

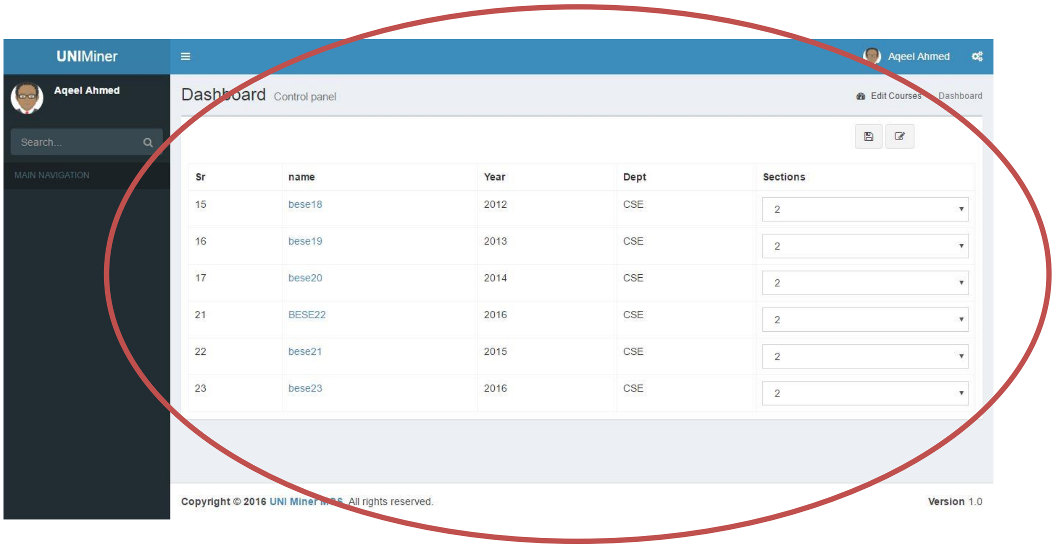
Select Type

Teacher

Select Deptartment

CSE

Edit Courses



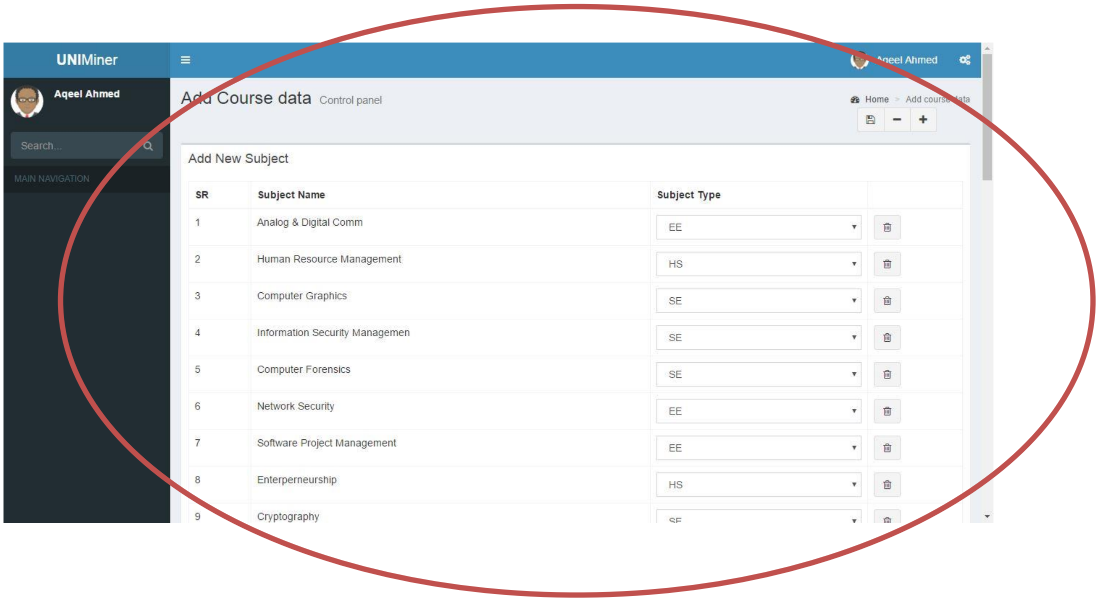
Dashboard Control panel

Sr	name	Year	Dept	Sections
15	bese18	2012	CSE	2
16	bese19	2013	CSE	2
17	bese20	2014	CSE	2
21	BESE22	2016	CSE	2
22	bese21	2015	CSE	2
23	bese23	2016	CSE	2

Copyright © 2016 UNI Miner MCS. All rights reserved.

Version 1.0

Edit Subjects

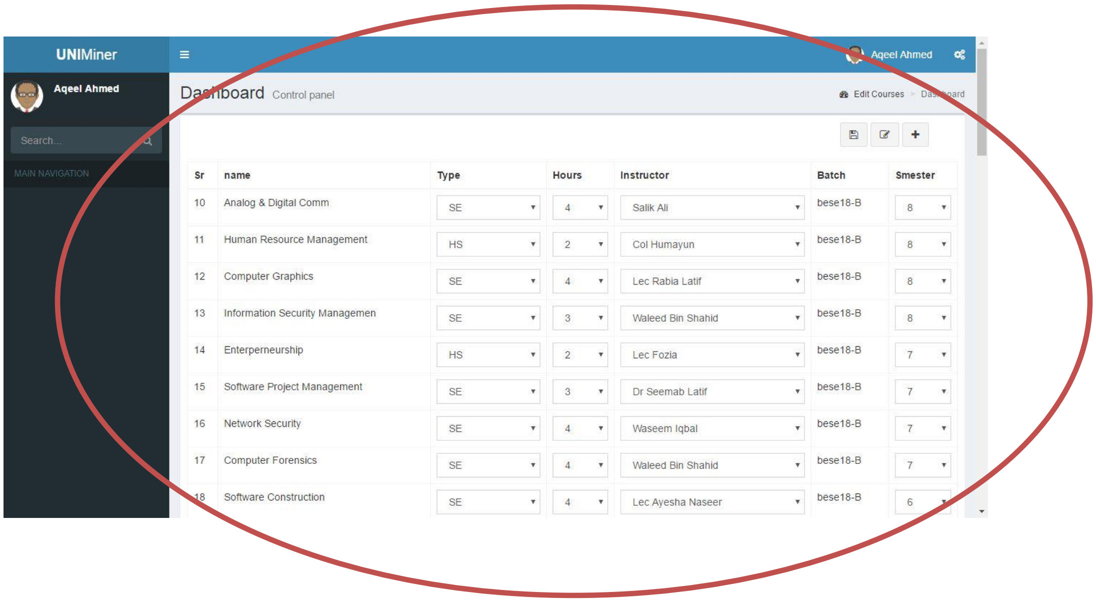


Add Course data Control panel

Add New Subject

SR	Subject Name	Subject Type
1	Analog & Digital Comm	EE
2	Human Resource Management	HS
3	Computer Graphics	SE
4	Information Security Managemen	SE
5	Computer Forensics	SE
6	Network Security	EE
7	Software Project Management	EE
8	Enterperneurship	HS
9	Cryptography	SE

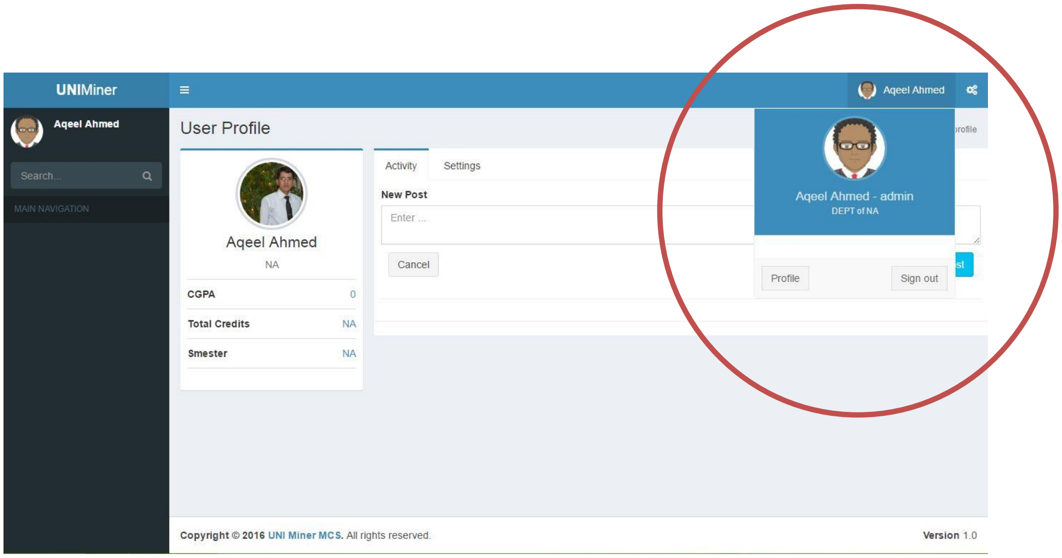
Edit Subject Metadata



Dashboard Control panel

Sr	name	Type	Hours	Instructor	Batch	Smester
10	Analog & Digital Comm	SE	4	Salik Ali	bese18-B	8
11	Human Resource Management	HS	2	Col Humayun	bese18-B	8
12	Computer Graphics	SE	4	Lec Rabia Latif	bese18-B	8
13	Information Security Managemen	SE	3	Waleed Bin Shahid	bese18-B	8
14	Enterperneurship	HS	2	Lec Fozia	bese18-B	7
15	Software Project Management	SE	3	Dr Seemab Latif	bese18-B	7
16	Network Security	SE	4	Waseem Iqbal	bese18-B	7
17	Computer Forensics	SE	4	Waleed Bin Shahid	bese18-B	7
18	Software Construction	SE	4	Lec Ayesha Naseer	bese18-B	6

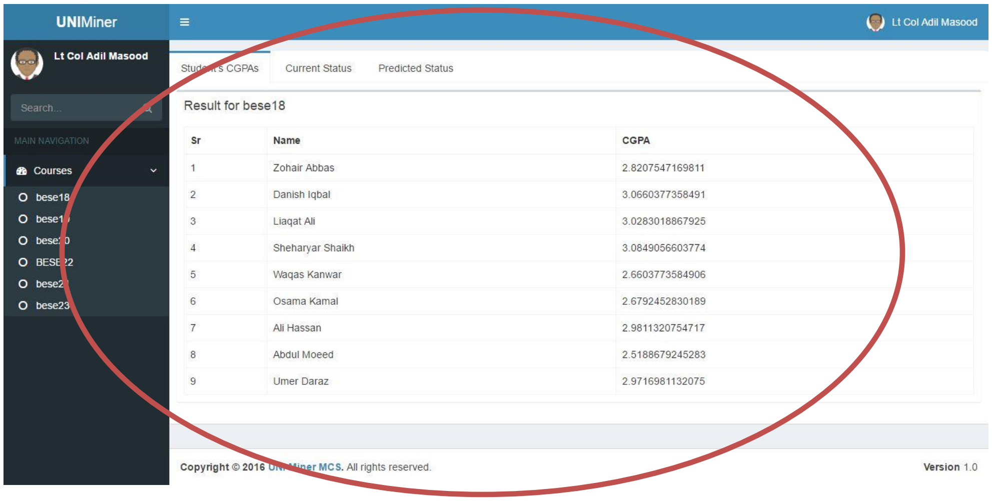
Logging out Administrator



CGPA	0
Total Credits	NA
Smester	NA

Copyright © 2016 UNI Miner MCS. All rights reserved. Version 1.0

Getting Started (Head of Department/Course Advisor)

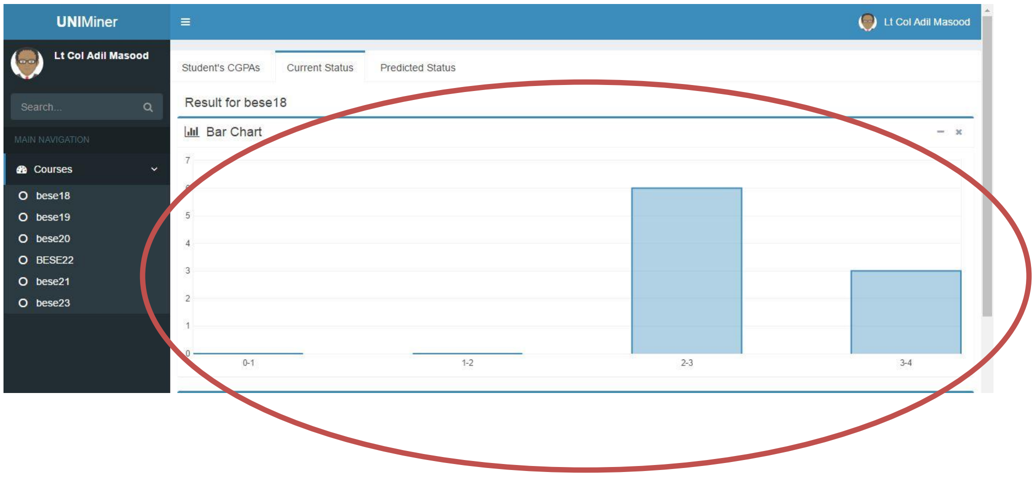


Result for bese18

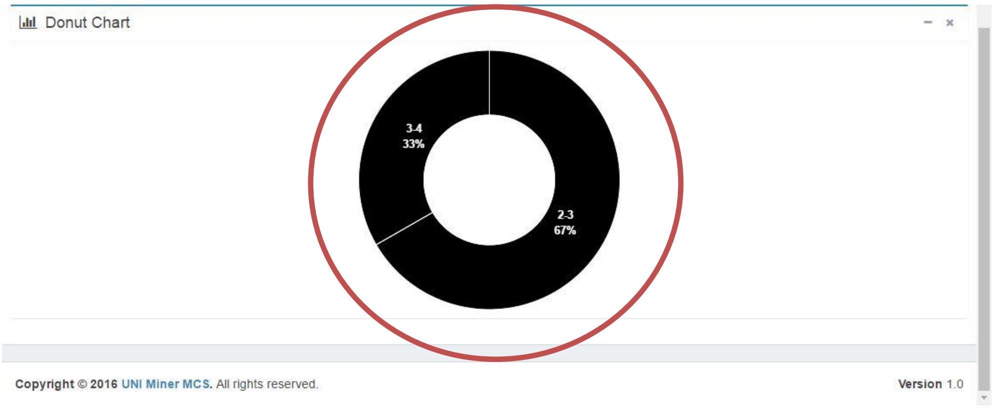
Sr	Name	CGPA
1	Zohair Abbas	2.8207547169811
2	Danish Iqbal	3.0660377358491
3	Liaqat Ali	3.0283018867925
4	Sheharyar Shaikh	3.0849056603774
5	Waqas Kanwar	2.6603773584906
6	Osama Kamal	2.6792452830189
7	Ali Hassan	2.9811320754717
8	Abdul Moeed	2.5188679245283
9	Umer Daraz	2.9716981132075

Copyright © 2016 UNI Miner MCS. All rights reserved. Version 1.0

View Current Results (Bar Graphs)



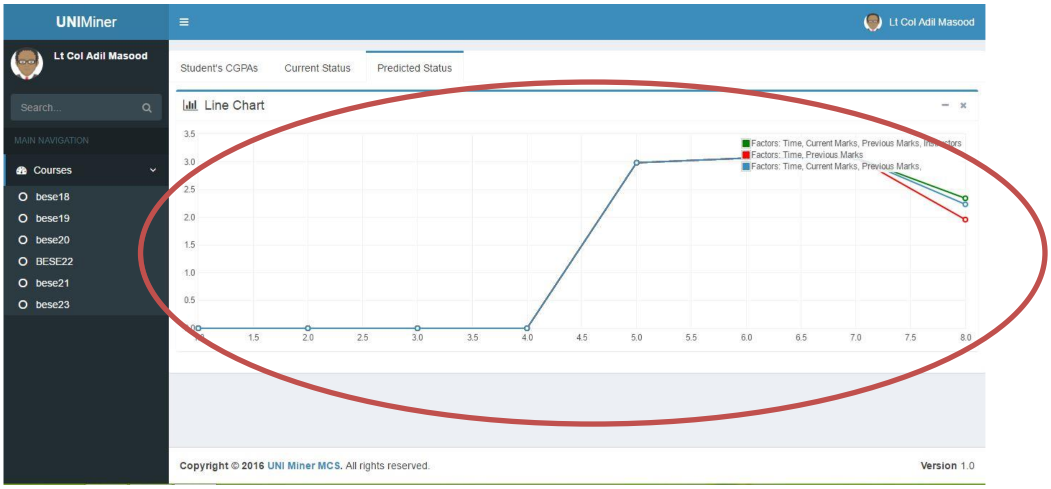
View Current Results (Donut Charts)



Copyright © 2016 UNI Miner MCS. All rights reserved.

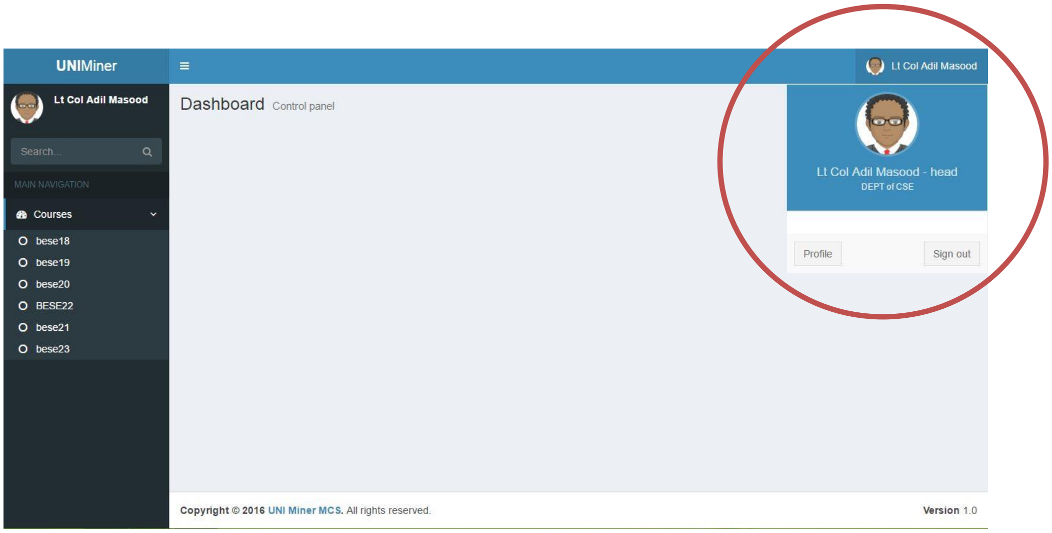
Version 1.0

View Predicted Results



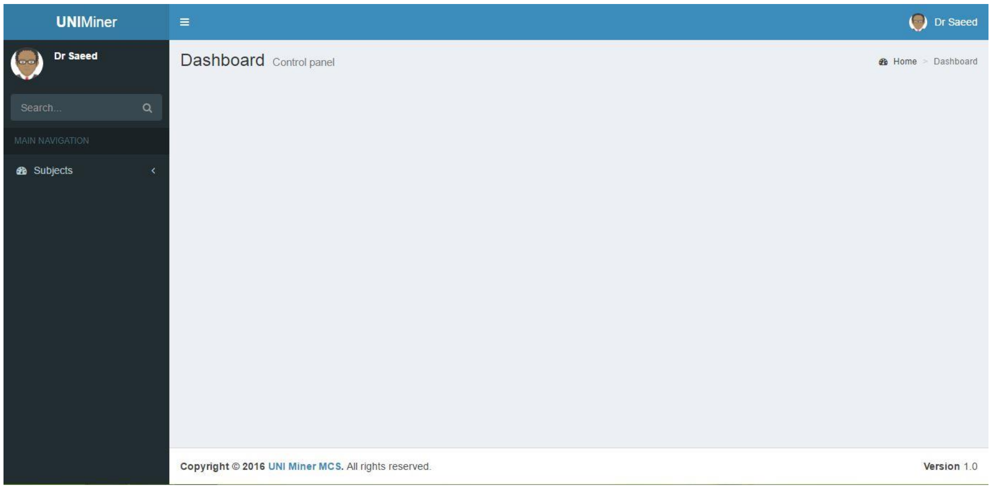
Copyright © 2016 UNI Miner MCS. All rights reserved.

Logging out Head of Department/Course Advisor



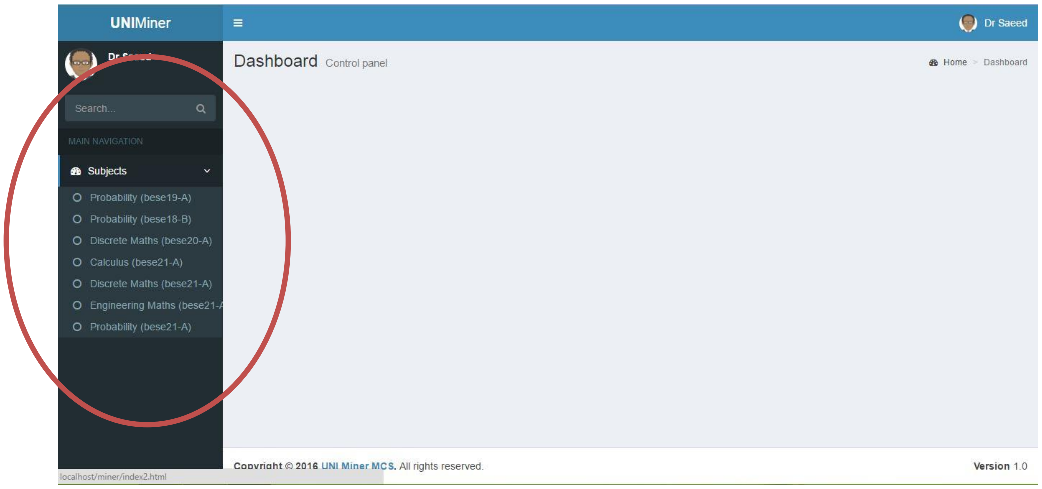
Copyright © 2016 UNI Miner MCS. All rights reserved.

Getting Started (Teacher)



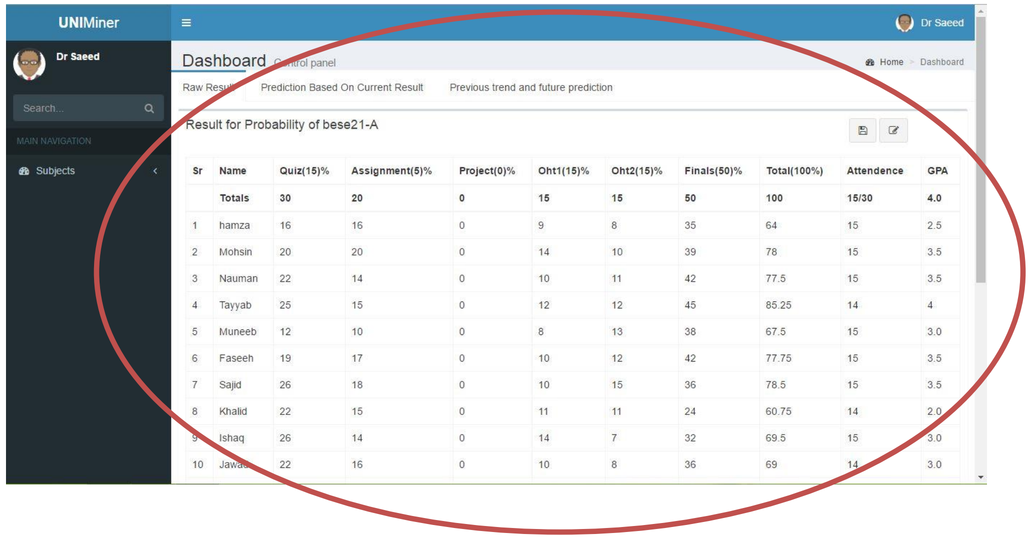
Copyright © 2016 UNI Miner MCS. All rights reserved.

View Subjects



Copyright © 2016 UNI Miner MCS. All rights reserved.

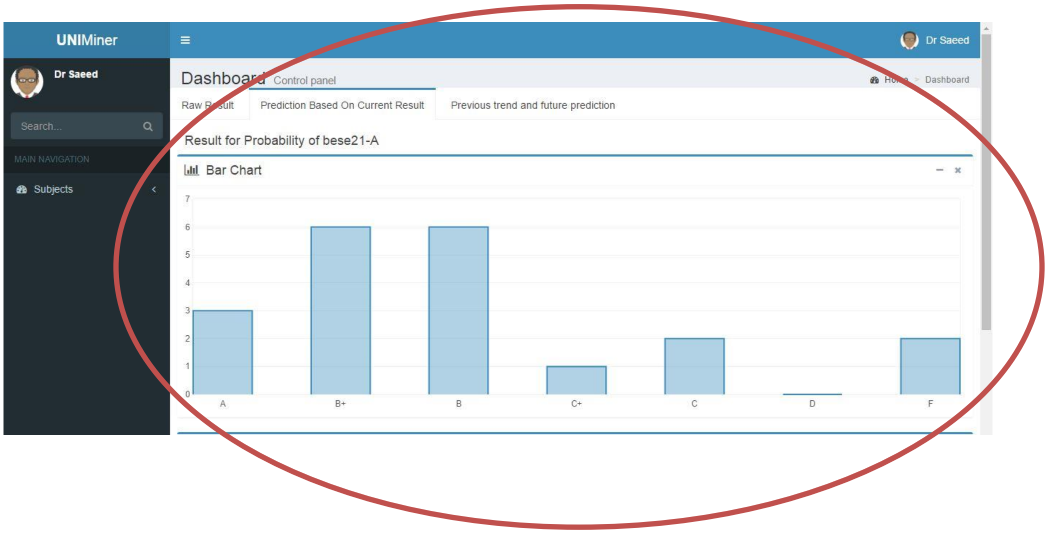
View Results (Selected Subject)



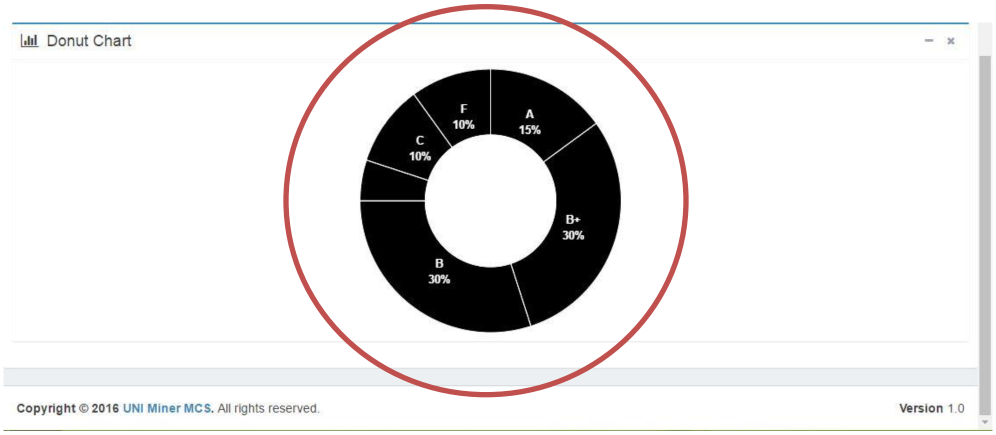
Result for Probability of bese21-A

Sr	Name	Quiz(15)%	Assignment(5)%	Project(0)%	Oht1(15)%	Oht2(15)%	Finals(50)%	Total(100%)	Attendence	GPA
	Totals	**30**	**20**	**0**	**15**	**15**	**50**	**100**	**15/30**	**4.0**
1	hamza	16	16	0	9	8	35	64	15	2.5
2	Mohsin	20	20	0	14	10	39	78	15	3.5
3	Nauman	22	14	0	10	11	42	77.5	15	3.5
4	Tayyab	25	15	0	12	12	45	85.25	14	4
5	Muneeb	12	10	0	8	13	38	67.5	15	3.0
6	Faseeh	19	17	0	10	12	42	77.75	15	3.5
7	Sajid	26	18	0	10	15	36	78.5	15	3.5
8	Khalid	22	15	0	11	11	24	60.75	14	2.0
9	Ishaq	26	14	0	14	7	32	69.5	15	3.0
10	Jawad	22	16	0	10	8	36	69	14	3.0

View Results (Selected Subjects-Bar Graphs)

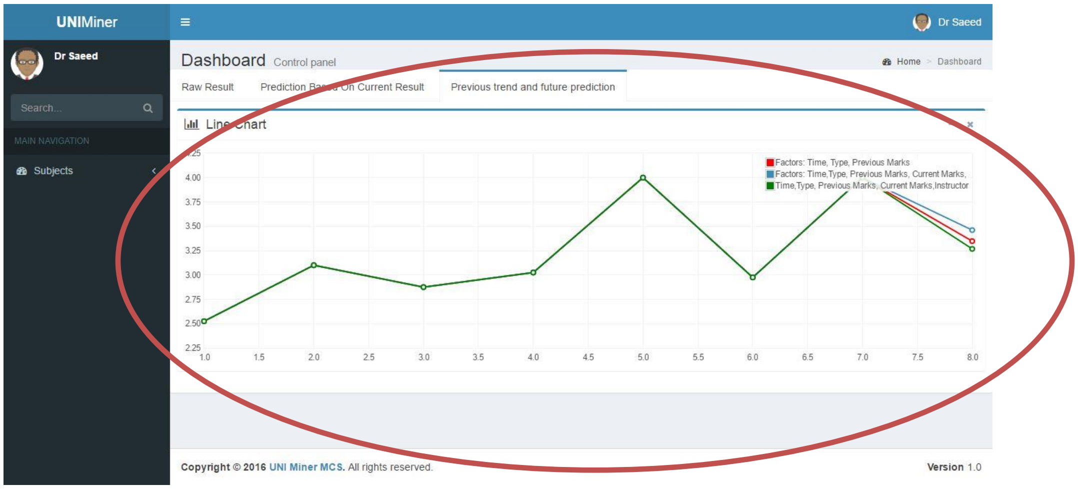


View Results (Selected Subjects-Donut Charts)



Copyright © 2016 UNI Miner MCS. All rights reserved.

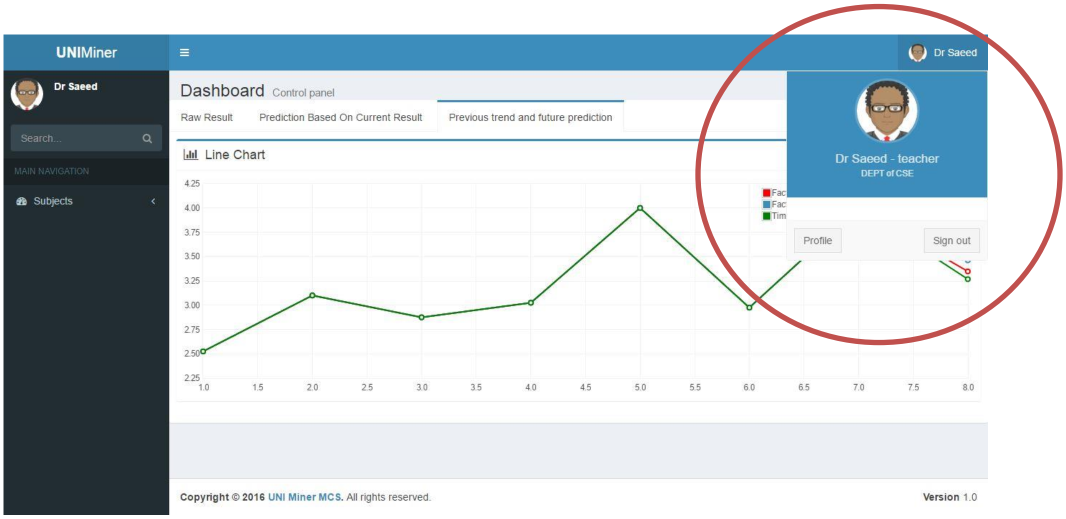
View Predicted Results (Selected Subject)



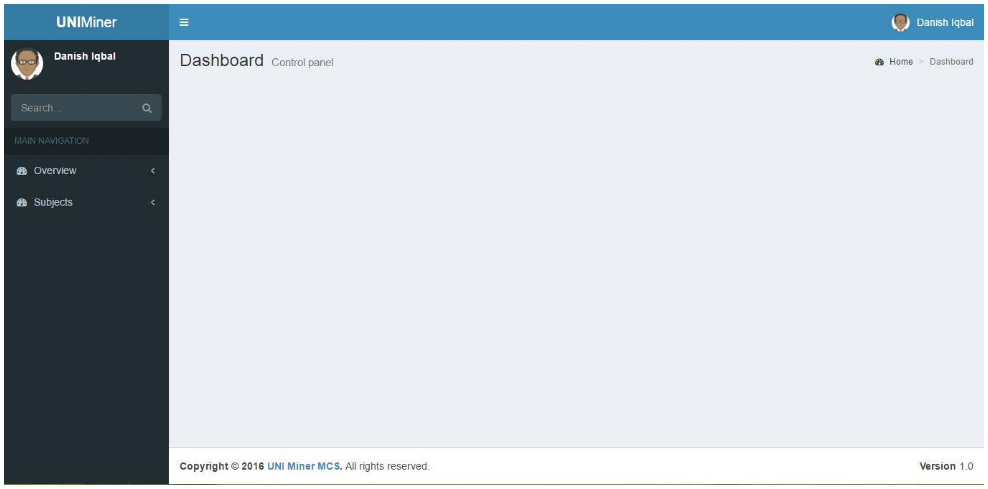
UNIMiner
≡
Dr Saeed
Dr Saeed
Search...
MAIN NAVIGATION
Subjects
Dashboard Control panel
Home > Dashboard
Raw Result
Prediction Based On Current Result
Previous trend and future prediction
Line Chart
4.25
4.00
3.75
3.50
3.25
3.00
2.75
2.50
2.25
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
🟥 Factors: Time, Type, Previous Marks
🟦 Factors: Time,Type, Previous Marks, Current Marks,
🟩 Time,Type, Previous Marks, Current Marks,Instructor
Version 1.0

Copyright © 2016 UNI Miner MCS. All rights reserved.

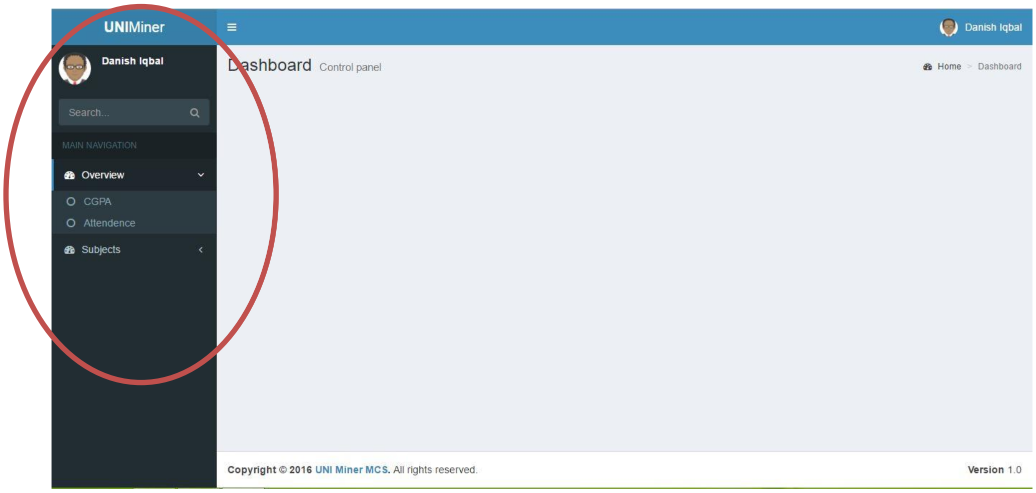
Logging Out Teacher



Getting Started (Student)



View Side Bar Options



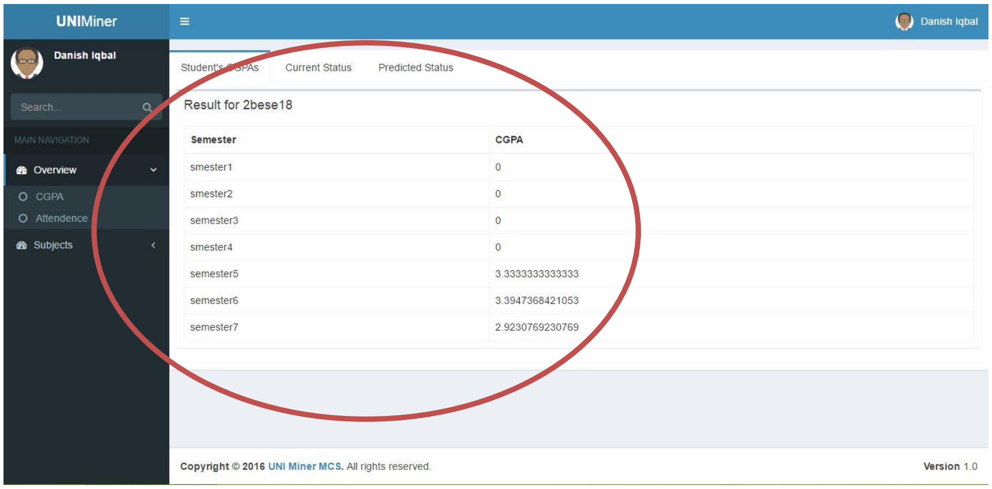
Dashboard Control panel

Home > Dashboard

Copyright © 2016 UNI Miner MCS. All rights reserved.

Version 1.0

View Results (CGPA)



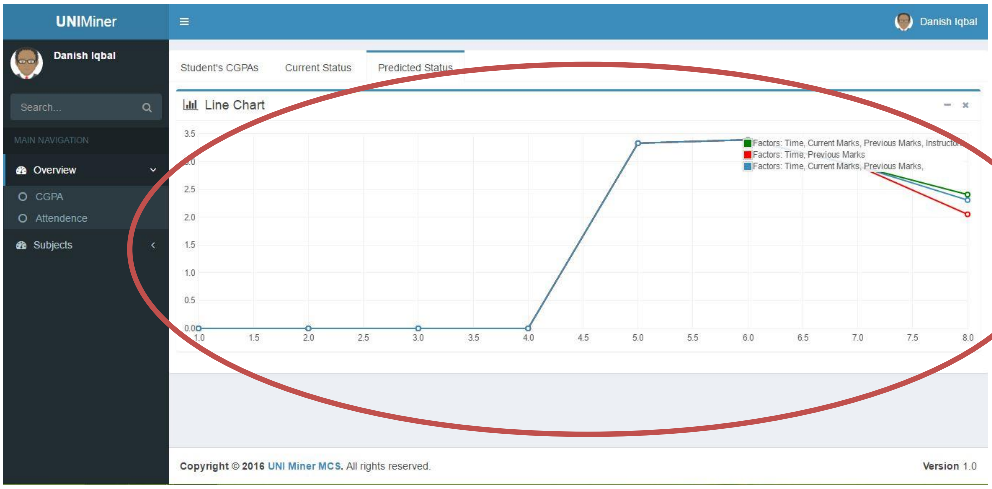
Result for 2bese18

Semester	CGPA
smester1	0
smester2	0
semester3	0
smester4	0
semester5	3.3333333333333
semester6	3.3947368421053
semester7	2.9230769230769

Copyright © 2016 UNI Miner MCS. All rights reserved.

Version 1.0

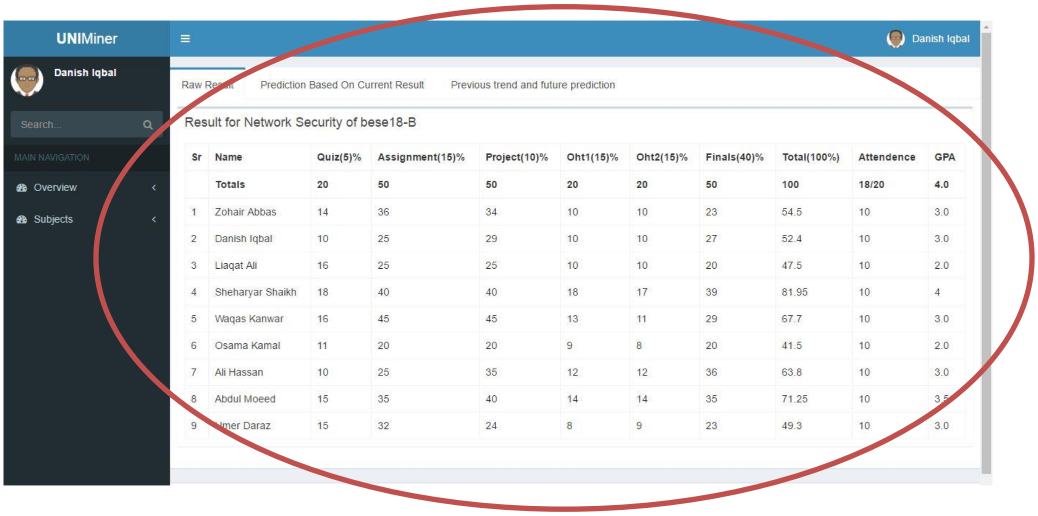
View Predicted CGPA



Copyright © 2016 UNI Miner MCS. All rights reserved.

Version 1.0

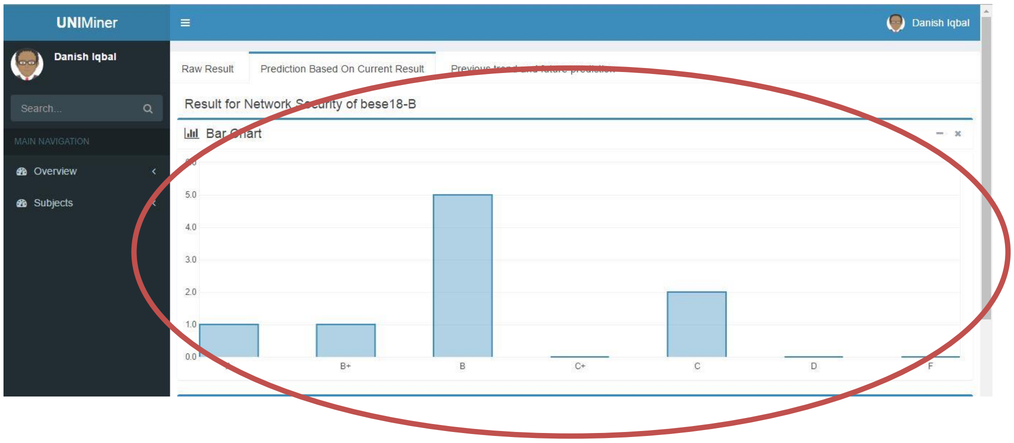
View Results (Selected Subjects)



Result for Network Security of bese18-B

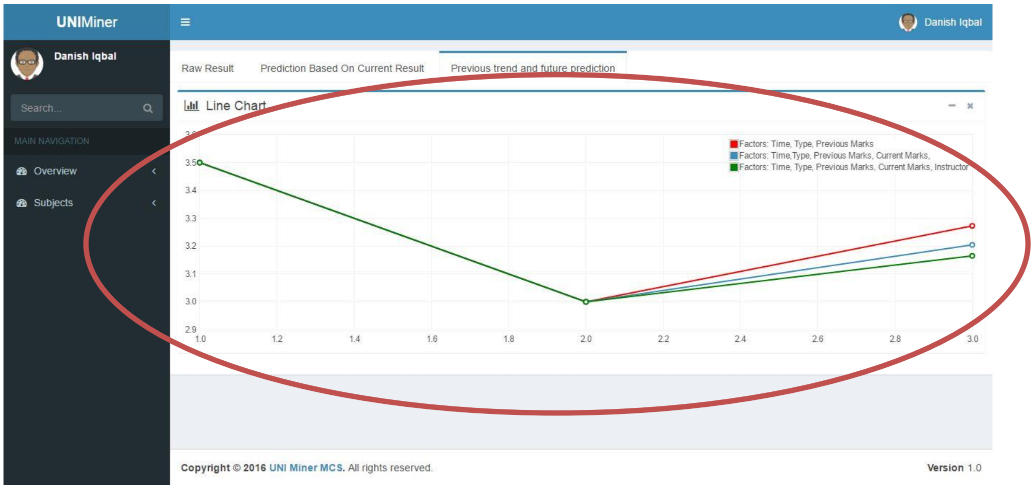
Sr	Name	Quiz(5)%	Assignment(15)%	Project(10)%	Oht1(15)%	Oht2(15)%	Finals(40)%	Total(100%)	Attendence	GPA
	Totals	**20**	**50**	**50**	**20**	**20**	**50**	**100**	**18/20**	**4.0**
1	Zohair Abbas	14	36	34	10	10	23	54.5	10	3.0
2	Danish Iqbal	10	25	29	10	10	27	52.4	10	3.0
3	Liaqat Ali	16	25	25	10	10	20	47.5	10	2.0
4	Sheharyar Shaikh	18	40	40	18	17	39	81.95	10	4
5	Waqas Kanwar	16	45	45	13	11	29	67.7	10	3.0
6	Osama Kamal	11	20	20	9	8	20	41.5	10	2.0
7	Ali Hassan	10	25	35	12	12	36	63.8	10	3.0
8	Abdul Moeed	15	35	40	14	14	35	71.25	10	3.5
9	Umer Daraz	15	32	24	8	9	23	49.3	10	3.0

View Results (Selected Subjects-Bar Graphs)



Result for Network Security of bese18-B

View Predicted Results (Selected Subjects)

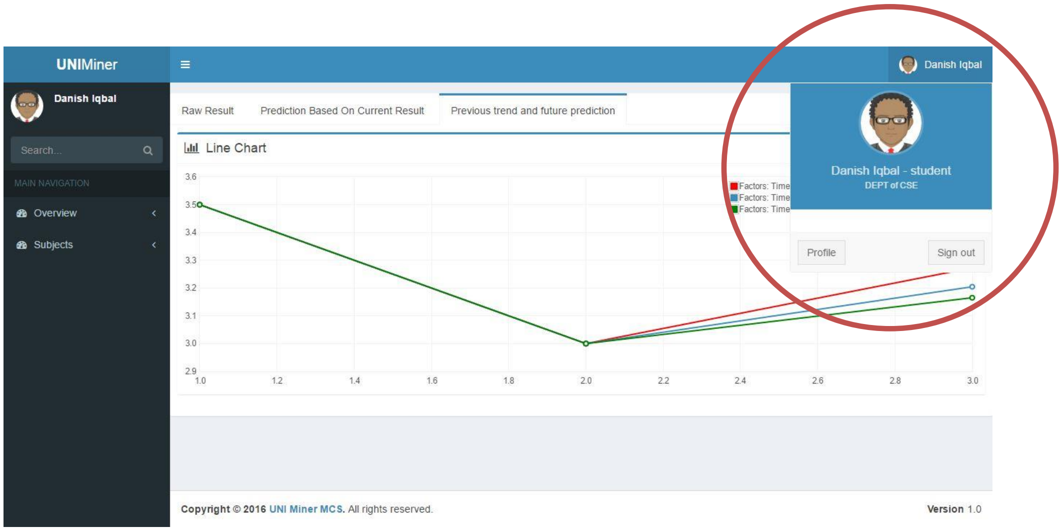


UNIMiner
Danish Iqbal
Danish Iqbal
Search...
MAIN NAVIGATION
Overview
Subjects
Raw Result
Prediction Based On Current Result
Previous trend and future prediction
Line Chart
🟥 Factors: Time, Type, Previous Marks
🟦 Factors: Time,Type, Previous Marks, Current Marks,
🟩 Factors: Time, Type, Previous Marks, Current Marks, Instructor
3.5
3.4
3.3
3.2
3.1
3.0
2.9
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0

Copyright © 2016 UNI Miner MCS. All rights reserved.

Version 1.0

Logging Out Students



Copyright © 2016 UNI Miner MCS. All rights reserved.