

School Mangement System

Assessment 2- Group Project Report, ICT206 – Software Engineering



Submitted on:

**Submitted by Group Members:**

Pratik Sapkota (CIHE22387)

Anish Ghimire (CIHE22584)

Jhalak Dhakal (CIHE22238)

Subham Sharma (CIHE22519)

Surekha Baral (CIHE22242)

Table of Contents

[1. Introduction 2](#_Toc147416714)

[2. Aim and Scope 2](#_Toc147416715)

[3. Methodology 3](#_Toc147416716)

[4. High level Requirements 4](#_Toc147416717)

[4.1 Functional Requirements 4](#_Toc147416718)

[**Testing Box**: 5](#_Toc147416719)

[4.2 Non-Functional Requirements 5](#_Toc147416720)

[5. Requirement Prioritization Table 5](#_Toc147416721)

[6.Analytic Hierarchy Process 6](#_Toc147416722)

[7. Design 10](#_Toc147416723)

[7.1 Architecture 11](#_Toc147416724)

[7.2 Event Driven 11](#_Toc147416725)

[7.3 Functional Decomposition 12](#_Toc147416726)

[7.4 Entity Relationship Diagram 13](#_Toc147416727)

[7.5 Use Case Diagram 14](#_Toc147416728)

[7.6 DFD Diagram 15](#_Toc147416729)

[8. Class Design 17](#_Toc147416730)

[8.1 UML Class Design 18](#_Toc147416731)

[**Inheritance in Class Diagram:** 20](#_Toc147416732)

[8.2 UML State Design for school Management System 20](#_Toc147416733)

[8.3 UML Sequence Diagram 23](#_Toc147416734)

[9. Budget Breakdown 26](#_Toc147416735)

[10. Budget Breakdown 26](#_Toc147416736)

[11.Detail Timeline 27](#_Toc147416737)

[12. User Interface 28](#_Toc147416738)

[**AI Recommendation:** 29](#_Toc147416739)

[13. Conclusion 29](#_Toc147416740)

[14.Reference 30](#_Toc147416741)

# 1. Introduction

The School Management System (SMS) is a comprehensive software solution that aims to improve the efficiency and effectiveness of school operations. The SMS will streamline various administrative and academic processes, enhance communication between stakeholders, and optimize resource allocation.

A school management system (SMS) is a software application that helps schools to automate and manage their day-to-day operations. SMSs can be used to track student attendance, grades, and transcripts; manage staff schedules and payroll; and communicate with parents and students. The rapidly growing demands of education forces schools to adapt quickly and efficiently to the change in system according to the demand. SMS is just a tool that every school needs.

# 2. Aim and Scope

The primary purpose of this document is to provide a full overview of the capabilities of the school Management System. The school management system aims to the effectiveness and efficiency of the school administration. It automatically updates the information about the student registration, attendance, Schedules, and report generation.

The school management system also aims to give school members access to online course registration, academic and administrative records, wide range of extra services and copying the actual school experience on digital devices of users. In this system, we have included the following features as a scope of school management system:

**i)Students Accounts:** The system will relieve the creation of student accounts, accessible through unique login credentials.

**ii)Course Schedule Management:** Students can use this system to examine the available courses, examine course descriptions, and enroll students in their respective chosen courses. This system will also handle course scheduling by granting students the chance to manage and organize their academic calendars.

**iii)User-Friendly Interface:** This system will simplify many parts of school administration and the system will provide a native user interface with a collection of features.

**iv)Students and professor:** They will be able to submit assignments, class materials, monitor grades, cooperate with classmates, and participate and contribute to group projects.

**v)Administrative Tools:** School administrators will have access to administrative tools for managing student records, faculty information, academic programs, and generating relevant reports.

**vi)Parental Access:** The system also includes a feature that allows parents to monitor their academic progress of children, attendance, and communicate with teachers.

By observing the requirements bound in this document, the School Management System can be developed to successfully provide to the requirements of its users and guarantee capable school administration. This incorporates the user attendance monitoring, record-keeping, engagement, academic planning, attendance monitoring, and record-keeping.

# 3. Methodology

Methodology in school management systems (SMS) refers to the systematic approach used to design, develop, implement, and maintain the system. It provides a framework for ensuring that the SMS meets the specific needs of the school and its stakeholders. It involves the action that identifies the work tasks that are to be completed, the quality assurance points that will be required, and the milestones that will be used to indicate progress. In addition, it includes activities like project tracking and control along with risk management, configuration management, technical reviews, and other activities throughout the process.

The different types of following methodology can be used for the school management system.

**i) Waterfall model**

Each step of development is finished before moving on to the next in this sequential technique. It works effectively for projects with clear needs and little chance of modification.

**ii)Agile methodology**

The system is created using this incremental and iterative technique over the course of several short sprints. This makes it possible to include user feedback throughout the whole development process.

**iii)Object-oriented technology**

The object-oriented technique entails disassembling the system into smaller, reusable objects. The system is now simpler to create, manage, and understand.

The approach of SMS development that is chosen depends on the requirements of the institution. The size and complexity of the school, the accessibility of resources, and the desired timeline are all factors to take into consideration.

**Requirements gathering**: The first step is to gather the requirements of the system from the stakeholders. This includes the school administration, teachers, parents, and students.

**System analysis**: The next step is to analyze the requirements and identify the functional and non-functional requirements of the system.

**System design**: The system is then designed based on the functional and nonfunctional requirements. This includes designing the system architecture, database schema, and user interfaces.

**System development:** The system is then developed using a programming language.

**System testing**: The system is tested to ensure that it meets the requirements and is free of defects.

**System maintenance**: The system is maintained to fix bugs and add new features.

# 4. High level Requirements

## 4.1 Functional Requirements

The functional requirements of a school management system (SMS) are the features and functionalities that the system should provide. These requirements can be gathered from the stakeholders, including the school administration, teachers, parents, and students.

**1. Student Management:** The system should allow the registration of students with details like name, date of birth, contact information, and class enrollment.

**2. Authorization and User Authentication:** Users should be allowed to log in using their own passwords, including administrators, instructors, students, and parents. Role-based access control should be used to make sure that users can only access features that are relevant to them.

**3. Teacher management:** This should include the ability to add, edit, and delete teacher records, as well as assign teachers to classes and courses.

**4. Grade management:** Teachers must have the ability to input and modify grades for coursework, examinations, and assignments. It should be possible for parents and children to view grades and development.

**5. Tools for communication and collaboration:** This should allow teachers, parents, and students to communicate with each other.

**6. User Assistance:** Provide user assistance and help desk tools to resolve questions and problems. In addition to these essential functionalities, several SMS software packages provide extra functionality such as:

* **Parents Portal:** Parents can see their child's grades, attendance, and other information through the parent portal.
* **Student Portal:** Students can access their grades, timetable, and other information through the student portal.
* **Mobile app:** This allows users to use their mobile devices to access the SMS program.
* **Integration with other systems:** This enables SMS software to communicate with other systems such as student information systems (SIS) and learning management systems (LMS).

## **Testing Box**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| 1 | Test Case ID | Test Case Name | Steps | Expected Outcome |
| 2 | 1 | Positive test case: Pass result | Student with 85%, call Pass function. | Success message,  Result = Passed. |
| 3 | 2 | Negative test case:  Pass result | Student with 33 %, call Pass function. | Failure message,  Result = failed. |

## 4.2 Non-Functional Requirements

* In the system only authorized or relevant persons who intend to open the program can access the system by using the username and password.
* The performance of the system in which the data can be tracked and accessed.
* In the system update can be done easily and update.
* System user-friendly: it is easy to learn and use and a very interactive environment to use.
* Maintenance: backup for the database is available for the update which helps the system to be up to date.

# 5. Requirement Prioritization Table

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement id | Description | level | Notes |
| REQ101 | Student enrollment | high | It involves certain process such as, admit, verifying the data of student and get approved [4] |
| REQ102 | Manage student fee | high | It includes payment of fees, transaction, and approval. |
| REQ103 | Student detail management | medium | It involves keeping the student detail for the records and further use. |
| REQ104 | View student record | medium | It helps to review the student progress, academic and so on. |
| REQ105 | Update student detail | medium | It involves editing, managing, and changing the details of the student. |
| REQ106 | Generate student records | medium | It involves making the records of the student in other word student portfolio. |
| REQ107 | Manage courses and subjects | high | It involves the managing the specific subject for the student. |
| REQ108 | Mark attendance | medium | It shows the daily presence or absence of the student. |
| REQ109 | Assign student | medium | This process helps the student for the daily task which improve student learning and improvement in student. |
| REQ110 | Mark grade | medium | It allocates the marks obtain by the student in their portfolio. |
| REQ111 | View grade and attendance | medium | It allows the student and the teacher to observe the mark of the student. |

# 6.Analytic Hierarchy Process

**Pairwise comparison:** In the pairwise comparison table, the first step is to put the objective of the school management in the table. The second object is to compare the objective inside the table which depends upon which objective is more or less important. The third step is to determine the total and fourth step is to rank all the objectives to determine which one is more important, and which one is less. Eventually, reflect the result.

**Analytical hierarchical table:**

|  |  |  |
| --- | --- | --- |
| Requirements | Significant i.e(1-10) | Complicated I.e(1-10) |
| Student enrollment | 10 | 8 |
| Manage student fees and payment | 8 | 6 |
| Student detail management | 8 | 6 |
| View student records | 8 | 6 |
| Update student -detail management | 8 | 6 |
| View student grade and attendance | 8 | 6 |
| Update student detail | 8 | 6 |
| Generate student reports | 6 | 6 |
| Manage course and subjects | 8 | 8 |
| Mark attendance | 6 | 6 |
| Assign student | 6 | 6 |
| Mark grade | 6 | 6 |
| View grade and attendance | 8 | 6 |

In this above table the highest number indicates the higher significant and the lowest number indicates lower the importance or significance. On the other hand, in terms of complication or difficulty the higher number indicates the higher complication meanwhile, the lowest number indicates lower complication. In the above table, student enrollment is very significant for the further process or for function. Management of the student fees and payment is important but a little bit less important than student.

Before we are heading towards the normalized pairwise comparison, I would like to highlight the MoSCoW technique.

The MoSCoW technique is used in a large range of business field.it helps everyone who is involving in a project that gives a depth knowledge to understand which work needs to be done first according to the priority and how that work increases the performance, productivity, lower the operating cost and satisfy the customer. When choosing a software provider, it can help the stakeholder talk about the relevance of the product feature. Moscow technique has four selections which indicate the priority of the requirement and show the returns of the investment form the selection. The given below table will show the school management system requirement and priority.

|  |  |  |
| --- | --- | --- |
| Requirement | selection | Priority |
| Student enrollment | Must have | High |
| Manage student fees and payment | Should have | medium |
| Student detail management | Must have | High |
| View student records | Must have | High |
| Update student -detail management | Must have | High |
| View student grade and attendance | Could have | low |
| Update student detail | Must have | High |
| Generate student reports | Should have | medium |
| Manage course and subjects | Must have | High |
| Mark attendance | Must have | High |
| Assign student | Must have | High |
| Mark grade | Must have | High |

**Normalized Pair-wise Comparison:** Normalized pairwise comparison is like a framework of the matrix PVE (or w) is obtained by normalizing the vectors in each column of  the matrix, and then by computing the average of the resulting matrix rows.

Apparently, the diagonal element in the table is set to be 1 as we know the corresponding elements are equally important and equally prioritized compared between themselves.

For example, in first row

* "Enrollment" is considered 1 time more important than itself.
* "Fees & Payment" is considered 4 times more important than "Enrollment".
* "Student Detail" is considered 3 times more important than "Enrollment".
* "View Records" is considered 2 times more important than "Enrollment".
* And so on for the rest of the columns in the first row.

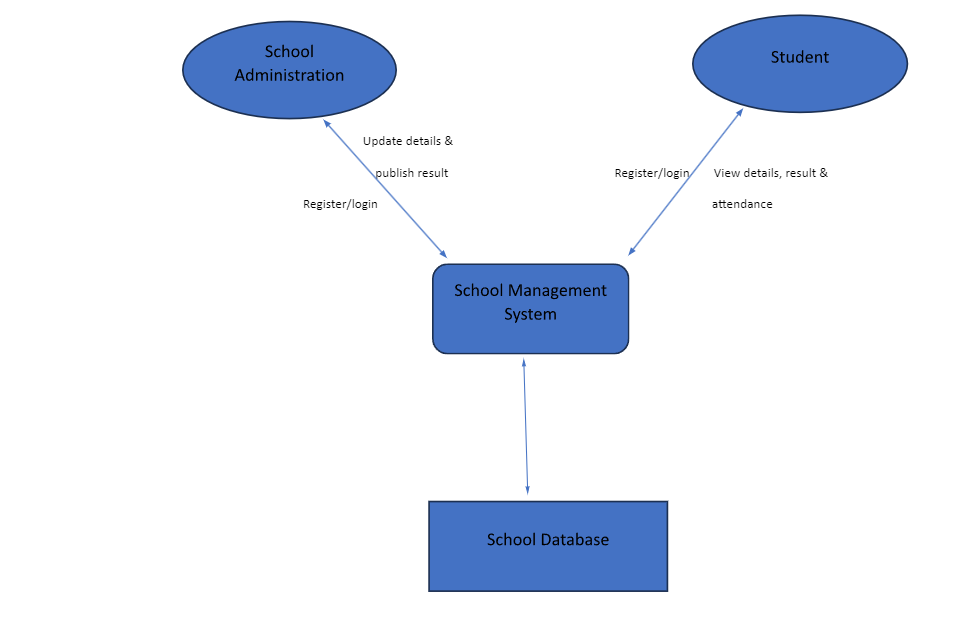
Table for normalized pairwise comparison.

A grid of black and white squares

Description automatically generated

# 7. Design

Good software design is very necessary for making every project successful. Here, we have made several designs and diagrams which show the process of how the School Management project works and implementation of this design in SMS. School administrator, school management, staff, parents, and students are the main figures for SMS.

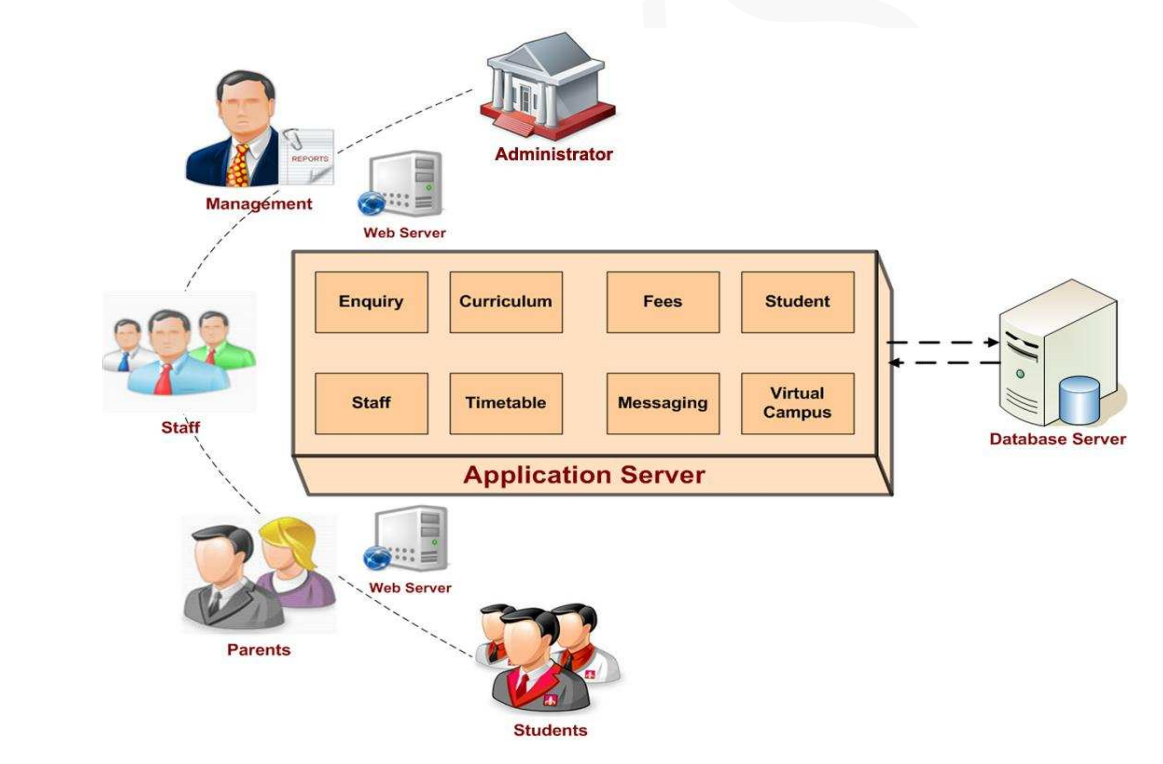


## 7.1 Architecture

Architecture means organizing software systems in a project and how it is related with other structural components. In architecture, multiple figures can work in a single database at the same time from different locations. It increases the flow of information between all the departments which also helps in increasing communication and system analysis.

School management system can use MVC architectural pattern (Model-View-Controller) for better effective results of each components (the model, the view, and the controller) which makes project more extensible and scalable.

The diagram below is the System Architecture of SMS.



Staff

AI Module

(Singhal, 2018)

## 7.2 Event Driven

Event-driven decomposition is a strategy for breaking down a system into smaller, more manageable components depending on the system's events or triggers. We can detect numerous events that influence the behaviors of the Library Management System. Here are some examples of probable occurrences and their components:

* **User Registration Event:**

User Registration Component: In charge of managing user registration, including verifying user details, establishing user accounts, and storing user information.

* **User Login Event:**

User Login Event: Handles user authentication and authorization, verifies user credentials, and allows access to system functionality.

* **Enrollment event:** Enrollment component: Enrolled new students and add information of student’s details in their database which includes their phone number, address, academic details, etc.
* **Manage course event:** Course component: It shows in which subjects’ students are enrolled in. It will contain subject name, unit code, lecture name and timetable as well.
* **Update student Details Event:** Student details component: Sometimes students change their mail, phone number and address. It is necessary to update this information in the system so that it won’t create any problem with communication.
* **Allocate grade event:** Grade component: It contains results of student’s quiz, assignment, and exams. Teachers are the main people who allocate grades to the student’s report and exams.
* **Assign student’s event:** The teacher teaches students and to check students’ knowledge of subject, they assign students some tasks which may be some project, assignment, tutorials, and workshop.
* **Generate report event:** When the teacher allocates grades to student exams, they send it to the coordinator, and they generate report cards which later were viewed by students.

## 7.3 Functional Decomposition

* **Management of user account:** It manages user (teacher and student) account, registration, and login details.
* **Course and subject management:** Manages the whole curriculum activities and assign subjects to the teacher which they are going to teach. They also manage subjects for students as well.
* **Enrollment process:** Handles the process of enrolling new students and manages their academic and personal details.
* **Grading process:** After students submit their assignments and quizzes, the teacher checks their paper and allocates grades which are later published by the coordinator. And students can view their results.
* **Manage fee:** Manages students overall payment procedure. They check whether students have paid their fees on time or not.
* **Fine and penalties:** It manages fines and penalties for students. If students fail to pass some subjects, they must pay some fine and penalties. They can charge it on late submission as well.

## 7.4 Entity Relationship Diagram

An Entity-Relationship Diagram (ERD) is a model and graphical representation that shows relationships between entities which may be persons, objects, places, or any kind of events. It highlights each detail and attributes of every entity and information is stored in tables. The ERD diagram shows how these tables are related to each other. It contains three types of relation. They are one to one, one too many, many too many.

The diagram below is the ERD of the school management system.



1. One to one:

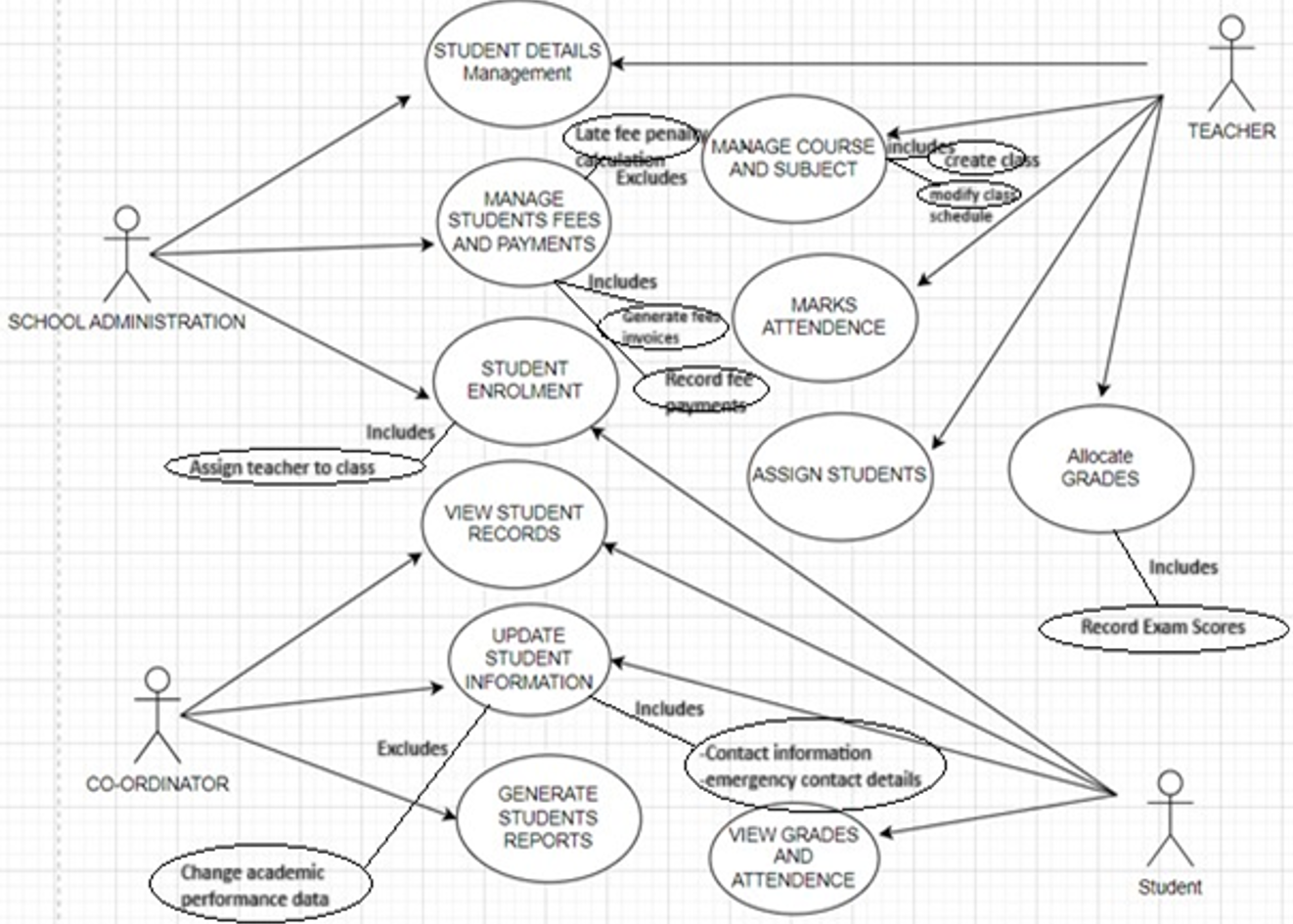
Here, in this ERD school administration and coordinator has a one-one relation. School administration contains only one coordinator, and a coordinator has only one school administration.

1. One too many: Here, coordinator and teacher, school administration and students, both have one too many relations. Only one coordinator supervises all the teachers and administration can enroll numerous students, but students can have only one administration.

1. Many too many: In the above diagram, teachers and students have many too many relations. A student can have many teachers and a teacher can teach multiple students.

## 7.5 Use Case Diagram

Use case diagrams contain actors, system how this project works, relationship between the actors and actor’s role in the given system. Use case diagram of SMS is drawn below:



A close-up of a document

Description automatically generated

## 7.6 DFD Diagram

DFD shows the way data and information flows throughout the process. This diagram does not go into details but shows the whole process of the project. Here, in school management system we have 3 levels of dataflow which are shown below:

**Zero Level Data Flow Diagram**

This is the first level of the system which shows the whole process of the project and analyzes the process.

A diagram of school management system

Description automatically generated

**First Level Dataflow Diagram**

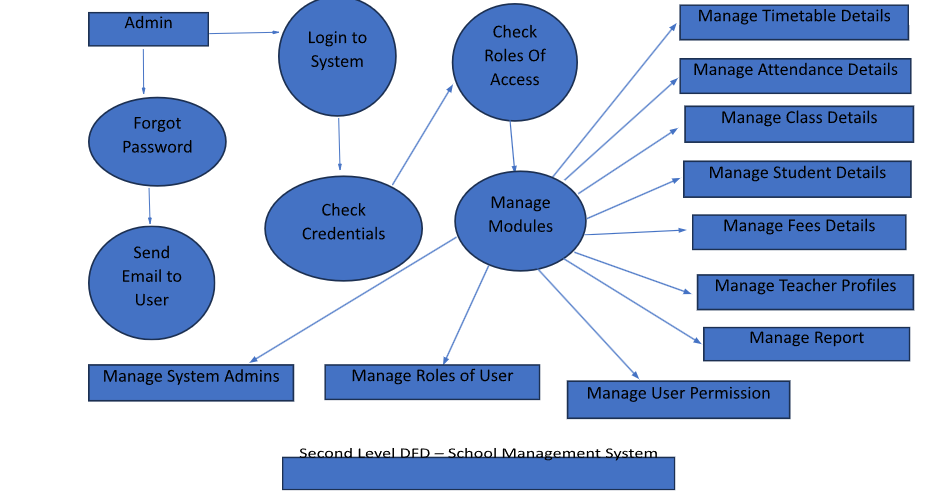
This level diagram shows how this system is divided into further parts, and it will also highlight the main function of School management system.

A diagram of a school management system

Description automatically generated

**Second Level Dataflow Diagram**

This level of diagram goes deep in Level 1 parts details. It shows each function of the level 1 sub-part.



# 8. Class Design

Class Design is a set of classes and a set of relationships between different classes which shows the static structure of the classifiers in a system. Class diagram offers an essential notation for further UML-required structural diagrams which is very useful to both designers and the rest of the team. Class diagrams are an instrument on which the company or any business’s researchers can use to model systems from a business point of view. Here we have created class Design for the school Management system.

The School Management System Class Diagram explains the structure of a class, also with its properties, functions, and relationships to other objects. The main components of the school management system are Schools, Courses, Students, Classes, Teachers, and Registrations/Enrollment.

**Diagram of the Classes of School Management Systems are:**

* **Schools Management:** Manage every aspect of school operations.
* **Students Management:** Manage all student activities and actions.
* **Classes class:**  Manage all a class's activities.
* **Teachers Management:** Manage all teacher activities.
* **Courses Management:** Manage all the Courses activities.
* **Registrations/Enrollment Management**: Control or manage all the enrollment-related activities.

A diagram of a class

Description automatically generated

## 8.1 UML Class Design

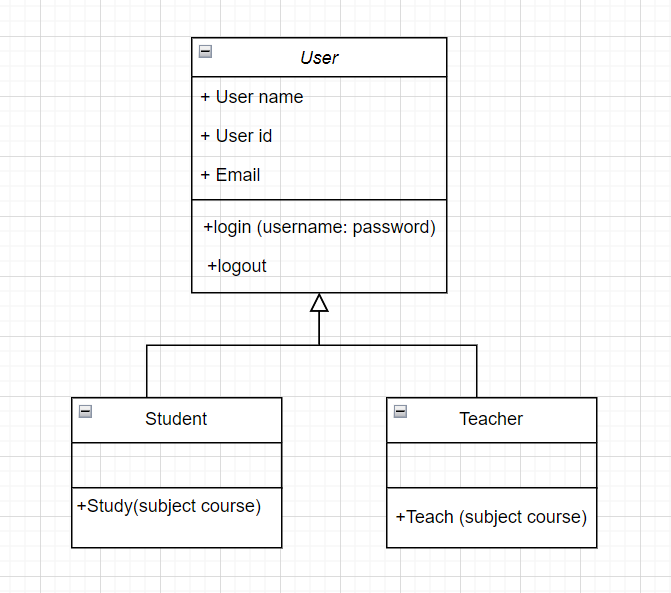
A diagram of a school management

Description automatically generated

The main components of this UML Class design with short description are given below:

* **School Management:** This signifies the main entity in this Design, which covers information about the school's name, location, and contact information and relationships with other classes, like the Coordinator, Teacher, and Student classes. It also shows the connections between the school and these objects. Furthermore, it provides the required tools to manage the school's faculty and students, such as addStudent, removeStudent, addTeacher, and removeTeacher.
* **School Administration:** It signifies the administrative team of the school like the administrators or principals. It stored the Information like adminName, adminID, email, and password inside it. It also provides the Logging in and logging out options.
* **Coordinator:** It signifies the coordinators or principals in charge of managing school activities. It includes the coordinatorName, coordinatorID, email, and password belongings. It also provides ways to sign in, sign out, and assign the students to their courses.
* **Teachers Management:** This mainly represents the Teaching staff of school. It includes the subjects that they teach, along with information like teacherName, teacherID, email, and password.  It also gives methods for controlling the subjects that they teach and for logging in and out.
* **Student Management:** It represents the enrolled students at school. It contains components like studentName, studentID, email, password, enrolled courses, grades, attendance, and fees. It also provides the ways to sign in, sign out, view grades, check the attendance, update student information, and generate students reports.
* **Students Details:** It contains further details on students, like contact information, address, and date of birth.
* **Enrollment/ Registration Management**: It maintains track of student’s details such as enrollment Date and status linked to a student's enrolment.
* **Marks:** It signifies grades or marks of students in a topic, along with the qualities for the grade and subject.
* **Course Management:** It manages and controls the courses and fields of study of the school. Also, it includes the procedures to add and delete properties for courses and subjects.

## **Inheritance in Class Diagram:**

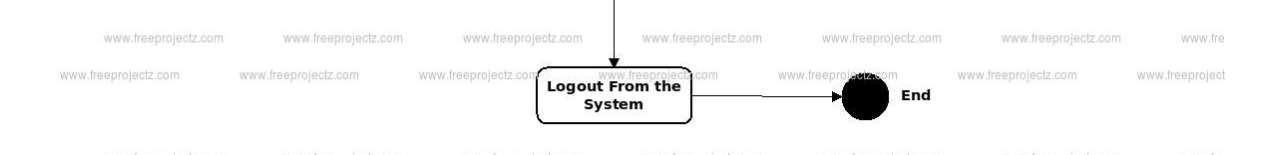


## 8.2 UML State Design for school Management System

This is the UML state diagram for School Management System which demonstrates the interactions between the activities of Course, School, Teacher, Enrollment and Classes. The following are the essential activities shown in this UML State Diagram of the School Management System:

* Course Activity
* School Activity
* Teacher Activity
* Classes Activity
* Registration/Enrollment Activity

****

****

**Features of this above School Management System state UML Diagram are given below in points:**

* A course can be added, informed, or deleted by an admin user. They can also search for a course and check its category.
* It also displays the activity or state flow of school's editing, adding, and updating.
* The user can look for teachers, enroll for classes, and produce reports.
* All objects are connected, including courses, schools, and classes in the above diagram.
* It provides a complete explanation of how the course, classes, enrolment, teachers, students, and schoolwork.

1. **Login state Diagram for Student and teachers for Searching course, generate student report and so on.**

A diagram of a computer program

Description automatically generated

A diagram of a student

Description automatically generated

**b. State Diagram for Student’s Attendance and Timetable**

A diagram of a student

Description automatically generated

## 8.3 UML Sequence Diagram

Sequence diagrams are those useful diagrams which explain how several things interact within a certain use case. Sequence diagrams are useful in locating the unknown missing items from the analysis object model. The sequence diagrams of each of the identified use cases are described below so that we can see how the items interact with each other. Here we have created a UML Sequence diagram for the school Management system.

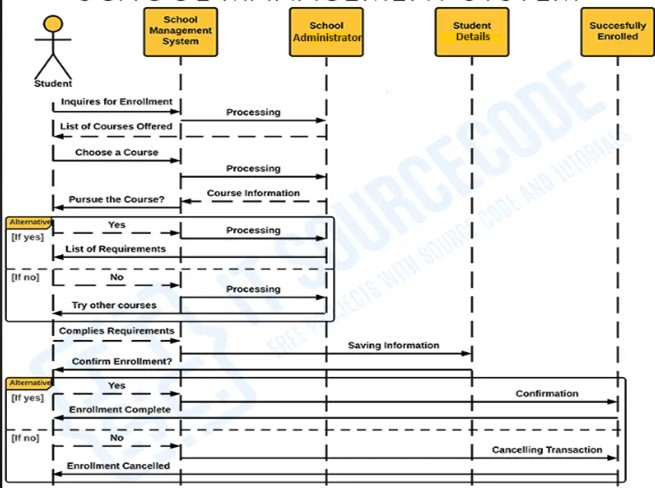
This is the UML sequence diagram of School Management System, which displays the relationships between the objects of Registration, Teacher, Course, Student, and Classes. Following are examples of the class objects used in this school management system UML sequence diagram:

* Registration/Enrollment Management
* Teacher Management
* Course Management
* Student Management
* School Management

A diagram of a school management system

Description automatically generated

1. **Sequence Diagram for student enrollment**



1. **Sequence diagram for recording student attendance**

A screenshot of a computer

Description automatically generated

1. **Sequence diagram for generating class timetable for student.**

**A diagram of a process

Description automatically generated**

# 9. Budget Breakdown

The estimated Price to develop an application or a website for the school management system: Costs for creating a website or web application for the school management system vary from $60,000 to $100,000. A team of 10-12 developers will be necessary to conduct and handle the responsibilities for creating websites or applications. It includes planning, deploying, creating, testing, debugging, and maintaining the school system web application. Here we have created a table for Project stage, Required Task, and their estimated project.

|  |  |  |
| --- | --- | --- |
| **Project Stage** | **Task** | **Estimated Budget** |
| Initiation | Initial meetings and arranging requirements | AUS$40000 |
| Planning | Designing | AUS$80000 |
| Implementation | Set up database | AUS$ 150000 |
|  | System programming |  |
|  | System Analysis |  |
| Control | Analyze | AUS$90000 |
| Closing | Final re-examination | AUS$30000 |
|  | The launch |  |
| **Total Budget Estimate** | **AUS$390000** |  |

**Explanation for Budget estimation:**

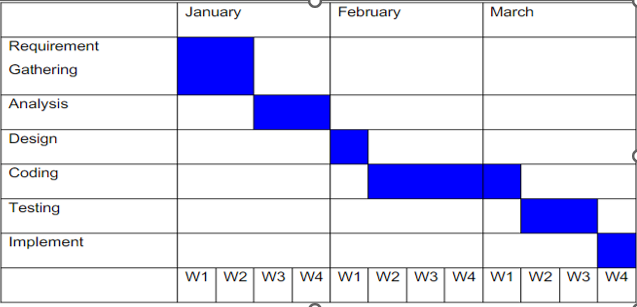
* **Initiation:** The first discussions with the main stakeholders are held on this first phase to gather the required needs and establish a project schedule. This stage is moderately estimated. It was not very expensive, around AUS$40000.
* **Planning:** The design development and requirement orientation are the main topics of the planning phase. There are extra expenditures, but they are still within a fair range around AUD$80000.
* **Implementation:** The significant economic expenditures are required during this stage, which also involves installation and creation of system components themselves. It costs around AUD$150000.
* **Control:** To make sure the smooth progress of project, this phase also involves threat analysis and risk assessment. Probably AUD$90,000 will be spent on this phase.
* **Closing:** This phase involves the official launch of the system and the final inspections and re-examinations. Australian dollars’ worth of $30,000 will be spent on this stage.

# 10. Budget Breakdown

It is important to keep in mind that the budget breakdown may change based on the needs of the specific project and variables like resource allocation and scalability. Once the budget has been created, the anticipated gain from the new system can be calculated by deducting the projected revenue from the budgeted expenses. Please be aware that these figures are only a rough guide and could change depending on the size and complexity of the project, makeup of team, and the location.

# 11.Detail Timeline

During the first two weeks of Jan, we collected all the data and information we needed to create a system from students, teachers, parents, school management and the rest of all the other staff of school. Then, we checked the data to see if it was relevant to our system or not. We analyzed whether our system could meet our expectations or not by analyzing the obtained info. Then we designed the wireframes and user interface. It took us a couple of months to do coding that fits the expectation from the data we had obtained. Then we did functional, performance and security testing. And finally, we rolled out our system to school and gathered feedback for improvements.



|  |  |  |  |
| --- | --- | --- | --- |
| **Project Stage** | **Task** | **Estimated Time** | **Timeline** |
| Initiation | Collect all Requirements | 1 weeks | 2 weeks |
|  | Meetings | 1 weeks |  |
| Planning | Design Prototypes | 3 Weeks | 3 Weeks |
| Implementation | Configuring the database System | 3 Months | Around 5months |
|  | System Programming | 1 Month |  |
|  | System analysis | 3 weeks |  |
| Control | System Evaluation | 2 Months | 2.5 Months |
|  | Threat Assessment | 2 weeks |  |
| Close Out | Re- examination Final | 1 Weeks | 1 week 1 day |
|  | Launch | 1 Days |  |

# 12. User Interface

This is the user/student’s interface that students can login to access the functionalities of our system. Both teachers or lecturers and students have access to our system. This system is specially for course structure, viewing student grades, visit profile and so on. Users can view and open their enrollment course from this system in our school management system.

A close-up of a notebook

Description automatically generated

After logging in the system, the user interface opens like this. From here, users can view and open their enrolled course, view their grades, assigned materials and examination timeline, and so on.

A close-up of a notebook

Description automatically generated

From this interface, students can access their assigned tasks, their grades, unit outline, their lecture materials, and weekly workshops. Students can view their courses they’re reading by entering my courses section. Students also have the option to log out just by going to the setting option as per their requirement.

Students can view their payment information by entering a fee payment option and can see upcoming fee payment dates, and paid fees.



This is the complete software solution for school management systems which enhances the use of technology which helps students, teachers, and administration to make things easier and more productive.

## **AI Recommendation:**



SMS can take advantages of AI algorithms to examine the style and movement of student’s activities, their learning pattern, performance, and behavior. With the help of this data-driven methodology, teachers have a stronger ability to spot at-risk learners, make wise choices, and act quickly to enhance student performance.

This approach is very useful in the school management system as there is a high risk of student dropout. It will help teachers and course coordinator to predict who is on the verge of dropout. In this case, AI will play a vital role by giving recommendation to the teachers, that some of the students are going to ruled out from the course/school. After that, teachers can give their time and effort to those students on time and protect them from begin dropout. It will directly help in their academic career.

# 13. Conclusion

A school management system is a complete software solution that significantly improves the productivity and coordination of educational institutions like schools and universities. It offers features including teacher and student profiles, attendance monitoring, grade management, and communication tools, as well as centralizes administrative operations, automates record-keeping, and centralizes administrative tasks. The system guarantees secure user authentication, enabling authorized workers to access data and features. The streamlining of communication through messaging and notification capabilities is advantageous for parent-teacher relationships and event announcements. It is appropriate for institutions with a range of sizes and unique requirements thanks to scalability and customization options. While integration capabilities with other systems enhance overall productivity, security safeguards, safeguard critical data. It may be greatly improved with the right instruction and continued support, making it an essential tool in modern education for improving operations, data management, and communication.

# 14.Reference

1. Activity Diagram for Student Attendance Management System. (n.d.). <https://meeraacademy.com/activity-diagram-for-student-attendance-management-system/>
2. Activity Diagram for Student Attendance Management System | Activity diagram, Student attendance, State diagram. (n.d.). Pinterest. Retrieved September 21, 2023, from <https://www.pinterest.co.uk/pin/activity-diagram-for-student-attendance-management-system--635148353705202005/>
3. School Management System UML Diagram | FreeProjectz. (n.d.). Www.freeprojectz.com. <https://www.freeprojectz.com/uml-diagram/school-management-system-uml-diagram>
4. Nym. (2022, January 5). School Management System Sequence Diagram | UML. Itsourcecode.com. <https://itsourcecode.com/uml/school-management-system-sequence-diagram/>
5. School Management System Sequence UML Diagram | FreeProjectz. (n.d.). Www.freeprojectz.com. Retrieved September 22, 2023, from <https://www.freeprojectz.com/uml-diagram/school-management-system-sequence-diagram>
6. Singhal, A. (2018, January 22). School Management System (School ERP) School Management System (school ERP) paper Less Education School Management System (SCHOOL ERP. Academia.edu. <https://www.academia.edu/35727461/School_Management_System_School_ERP_School_Management_System_School_ERP_Paper_less_Education_School_Management_System_School_ERP>
7. *Freeproject*. (DFD) FreeProjectz. (n.d.). <https://www.freeprojectz.com/dfd/school-management-system-dataflow-diagram>
8. Sydle. (2023). Enrollment process: How does it work? How can you organize it? *Blog SYDLE*. <https://www.sydle.com/blog/enrollment-process-63e27c29a128f13e2e83c1a9#:~:text=Traditionally%2C%20the%20enrollment%20process%20is,an>
9. Brush, K. (2023). MoSCoW method. *Software Quality*. <https://www.techtarget.com/searchsoftwarequality/definition/MoSCoW-method>

‌