

**A PROJECT REPORT  
ON  
School Management System  
Presented and Submitted By**

**Mohammad Zeeshan  
PRN NO. 04420002045**

**Towards The Partial Fulfilment of the  
Bachelor Computer Application**



**Vikas College Study Center,  
Vikhroli (East).**

**Tilak Maharashtra University, Pune  
Faculty of Regular Education  
[2020 – 2023]**



## **Tilak Maharashtra Vidyapeeth**

('Deemed University' under section 3 of UGC Act 1956 vide 9-notification NO.F 19/85-U3 dated 24h April 1987 by the Government of India) Vidyapeeth Bhavan, Gultekdi, Pune- 411 037

# **CERTIFICATE**

This is to certify that project  
**School Management System**  
Has been satisfactorily completed  
**BY**

**Mr. Mohammad Zeeshan (Prn - 04420002045 & Seat no - 1999)**

Towards The Partial Fulfilment of the 'Bachelor of Computer Application', For the Academic year 2022-2023 at Tilak Maharashtra University, Pune (Faculty of Regular Education) And  
It is approved.

---

Examiner

Head of Department

[Vikas college Vikhroli]



Vidya Vikas Education Society's  
**VIKAS COLLEGE STUDY CENTRE**

Vikas High School Marg, Kannamwar Nagar No.2, Vikhroli (East), Mumbai - 400 083.

---

**Prof. Vikas P. Raut**

Study Centre Co-ordinator

**Shri .P. M. Raut**

Chairman, V.V.E. Society

---

**Certificate**

This is to certify that the project  
“College Management System”  
Has been satisfactory completed by

Mohammad Zeeshan  
Prn No.04420002045

Towards The Partial Fulfilment of the 'Bachelor of Computer Application',  
For the Academic Year [2022-2023] at Vikas College Study Centre,  
Vikhroli (East),  
Tilak Maharashtra Vidyapeeth, Pune (Faculty of Distance Education),  
And it is approved.

---

**Project Guide**

Ms. Asma Qazi

**Examiner**

**Head of  
Department**

Mr. Vikas Raut

## **ACKNOWLEDGEMENT**

With immense pleasure we are presenting 'COLLEGE MANAGEMENT SYSTEM' Project report as part of the curriculum of 'Bachelor of Computing Application'. We wish to thank all the people who gave us unending support.

We express our profound thanks to our head of department Mr. Vikas Raut", project guide and project in charge Ms. Asma Qazi" and all those who have indirectly guided and helped us in preparation of this project.

### **School Management System**

**Mohammad Zeeshan  
04420002045**

# **School Management System**

# PROJECT SYNOPSIS

## OBJECTIVE OF THE PROJECT

The main objective of the Project on School Management System is to manage the details of Schools, Students, Teachers, Admin, Student Attendance View Fees, Pending Fees, and Publish Notices. It manages all the information about Schools, Registrations, and Login. The project is totally built at the administrative end and thus only the administrator is guaranteed access. The purpose of the project is to build an application program to reduce the manual work of managing the Schools.

### **The objectives of the system are -**

- No paperwork is required & can be implemented further
- Reduced operational time.
- Increased accuracy and reliability.
- Make all the systems computerize
- Reduce error scope

<b>Name of Component</b>	<b>Specification</b>
Operating System	Microsoft Windows 10 (OS)
Language Used	Python
Database	SQLite
Development Platform	Vs code, Google Chrome
Hardware Requirements	Ram 2GB, Hard Drive 10GB

# **INDEX**

<b>Chapter No.</b>	<b>Topics</b>
1	Introduction
2	Need for New System
3	Feasibility Study
4	Technology Review
5	SDLC Life Cycle
6	Software Development Model
7	Stakeholders
8	Gantt Chart
9	UML Diagrams  9.1 E-R Diagrams 9.2 Use Case Diagrams 9.3 Class Diagrams 9.4 Object Diagrams 9.5 Data Flow Diagrams 9.6 Sequence Diagram 9.7 Activity Diagram 9.8 Context-Level Diagrams
10	Data Structures Screenshots
11	Project Screen Shots
12	Testing and Validations

13	System Maintenance
14	Limitations and Future Enhancements
15	Conclusion
16	Bibliography



# **CHAPTER 1**

## **INTRODUCTION**

The "School Management System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and sometimes reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides an error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by this all, it proves it is user-friendly. School Management System, as described above, can lead to an error-free, secure, reliable, and fast management system.

# **CHAPTER 2**

## **Need for New System**

In the offline system, it is an overhead to keep the records related to faculty, student, parents, and other school staff on the papers. Everything related to their progress in the system is marked manually. E.g. A report of a student's attendance is generated monthly and is shown to his/her parents. Now, a regular student, going to school every day, is marked absent for a day by mistake.

It is a burden to take out the register and view the records. As you can see, it is a very time-consuming process and it costs much. So, I thought why I should not help these young guns of the nations to help them to have a bright future and to make an online centralized platform that can be accessed from anywhere in the world.

My other aim is to minimize the paperwork as minimum as I can so that there is no need to cut more and more trees. Indirectly, I will be helping Mother Nature.

## **2.1 Features of New System**

- Student Panel
- Teacher Panel
- Admin Panel
- Manage Requests
- Publish Notice
- View Fees, Pending Fees
- Student Attendance Management
- Manage the information of Schools
- Editing, Adding, and Updating Records
- Manage the information of Teachers
- Manage the information of Students
- Accuracy in work
- Easy & fast retrieval of information
- Access to any information individually

## **2.2 Advantages of Proposed System**

A school management system helps you manage all the student-related data in a well-organized manner. A school management system gives you a unique ID for every student. And using that ID, you can easily track the fee status, assignments, exam results, grades, and parent info within seconds.

# **CHAPTER 3**

# **Feasibility Study**

## **2.2 Technical Feasibility**

The minimum hardware required to implement the system is an AMD A6 processor with 1 GB RAM. The whole system is being developed on Html, CSS, and Bootstrap for GUI (Front End) and uses SQLite for Database.

## **3.2 Economical Feasibility**

It is a cost-effective system. No extra efforts are needed to train for using the system. This system is quite beneficial with respect to its cost. The user shall be conversant with computer technology. The system is user-friendly and thus easy to use and operate.

## **3.3 Operational Feasibility**

This system will reduce the workload and loss of vital information. It is time-saving and will aid the Customer to be system generated rather than person dependent. Data retrieval and data presentation will be done by the system so the report generation will be automated.

# **CHAPTER 4**

## **Technology Review**



## 4.1 Programming Language

### 4.2 Python

Python is a [high-level, general-purpose programming language](#).

[Guido van Rossum](#) began working on Python in the late 1980s as a successor to the [ABC programming language](#) and first released it in 1991 as Python 0.9.0.<sup>[37]</sup> Python 2.0 was released in 2000. Python 3.0, released in 2008, was a major revision not completely [backward-compatible](#) with earlier versions. Python 2.7.18, released in 2020, was the last release of Python 2.

Python consistently ranks as one of the most popular programming languages

It is used for:

Python is commonly used for developing websites and software, task automation, data analysis, and data visualization. Since it's relatively easy to learn, Python has been adopted by many non-programmers such as accountants and scientists, for a variety of everyday tasks, like organizing finances.

## 4.3 Framework

### 4.4 [Django](#)

Django is a free and open-source, Python-based web framework that follows the model–template–views architectural pattern. It is maintained by the Django Software Foundation, an independent organization established in the US as a 501 non-profit.

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

### [Bootstrap](#)

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS, and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

## 4.5 Database\_

### SQLite

SQLite is a database engine written in the C programming language. It is not a standalone app; rather, it is a library that software developers embed in their apps. As such, it belongs to the family of embedded databases.

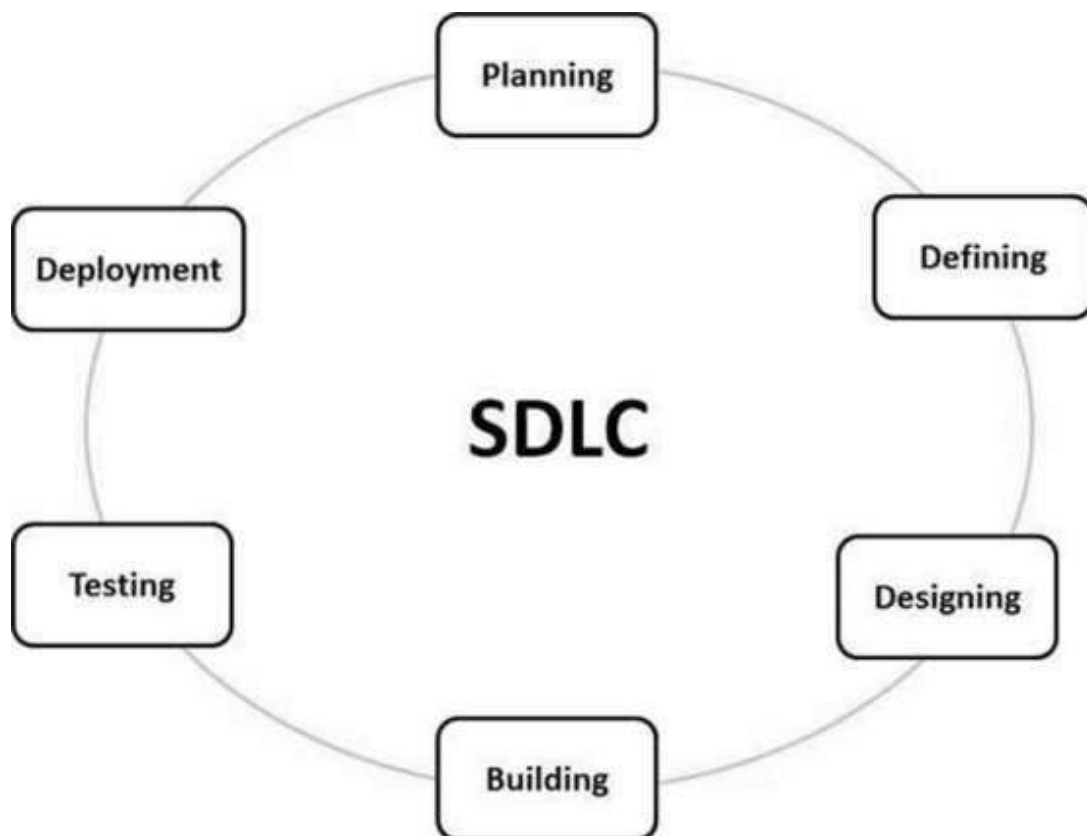
SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is the most widely deployed SQL database engine in the world. The source code for SQLite is in the public domain. This tutorial will give you a quick start with SQLite and make you comfortable with SQLite programming.

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database [file format](#) is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between [big-endian](#) and [little-endian](#) architectures. These features make SQLite a popular choice as an [Application File Format](#). SQLite database files are a [recommended storage format](#) by the US Library of Congress.

# **CHAPTER 5**

## **SDLC Life Cycle**

**SDLC** is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace, and alter or enhance specific software. The Software Development Life Cycle defines a methodology for improving the quality of software and the overall development process. The following figure is a graphical representation of the various stages of a typical SDLC.



# **CHAPTER 6**

## **Software Development Mode**

To solve actual problems in an industry, it is imperative to adopt a development strategy that encompasses the process, methods, and tool layers. The strategy is incorporated by the software engineer and his team and is called Software Development Life Cycle (SDLC). It aims to produce high-quality software that meets or exceeds customer expectations and reaches completion within times and cost estimates. Software Development Process Models are defined and designed and are meant to be followed during the software development process. The following are the most important and popular SDLC models followed in the industry.

## **Waterfall Model**

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

## **Iterative Model**

In the Iterative model, the iterative process starts with a simple implementation of a small set of software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed

An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

## **Spiral Model**

In the Iterative model, the iterative process starts with a simple implementation of a small set of software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.



## **RAD Model**

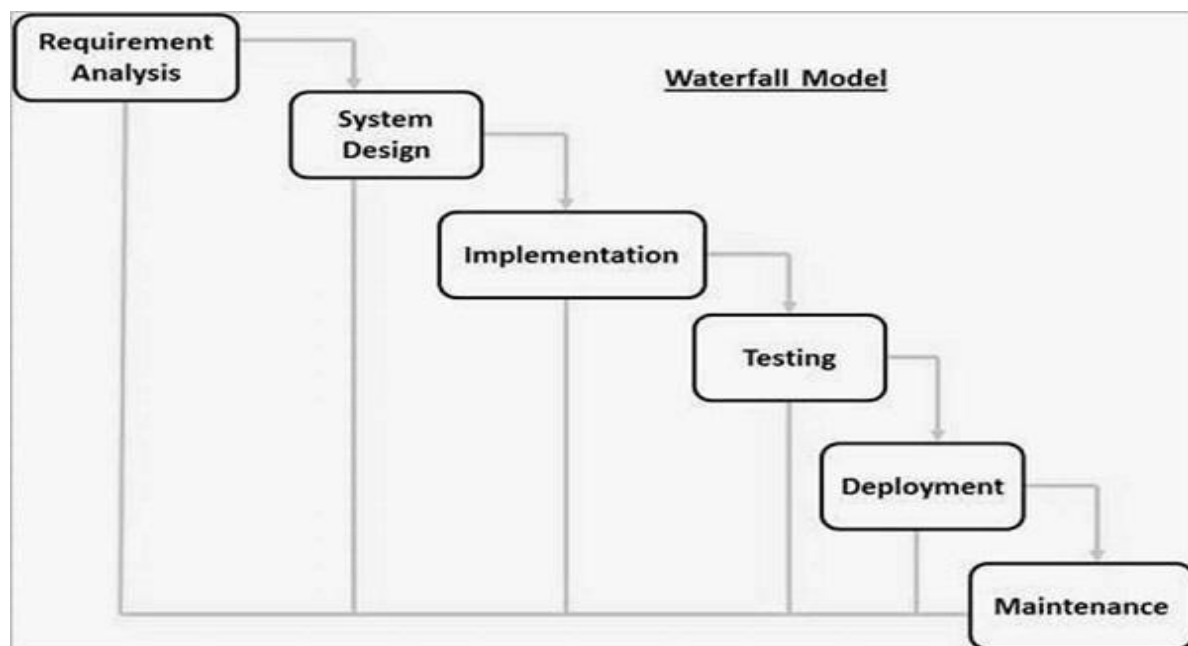
The RAD (Rapid Application Development) model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product. Rapid Application Development focuses on gathering customer requirements through workshops or focus groups, early testing of the prototypes by the customer using iterative concepts, reuse of the existing prototypes (components), continuous integration, and rapid delivery.

## **V - Model**

The V-model is an SDLC model where the execution of processes happens in a sequential manner in a V-shape. It is also known as Verification and Validation model. The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.

## 6.1 Structure of Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The Waterfall model is the earliest SDLC approach that was used for software development. The waterfall Model illustrates the software development process in a linear sequential flow.



### **6.2.1 Requirement Gathering and Analysis**

All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

### **6.2.2 System Design**

The requirement specifications from the first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

### **6.2.3 Implementation**

With inputs from the system design, the system is first developed in small programs called units, which are integrated into the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

### **6.2.4 Integration and Testing**

All the units developed in the implementation phase are integrated into a system after testing each unit. Post integration the entire system is tested for any faults and failures.

## **6.2.5 Deployment of system**

Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

## **6.2.6 Maintenance**

There are some issues that come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment. All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for the previous phase and it is signed off, so the name " Waterfall Model ". In this model, phases do not overlap.

## **6.3 Advantages of the Waterfall Model**

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Development moves from concept, through design, implementation, testing, installation, and troubleshooting, and ends up with operation and maintenance. Each phase of development proceeds in strict order.

## 6.4 Disadvantages of the Waterfall Model

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The major disadvantages of the Waterfall Model are as follows –

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty are high with this process model.
- It is difficult to measure progress within stages. • Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big bang". at the very end, which doesn't allow identifying any technological or business bottlenecks or challenges early.

# **CHAPTER 7**

## **Stakeholders**

It is software for the Project on School Management System to manage the details of Schools, Students, Teachers, Admin, Student Attendance View Fees, and Pending Fees, and Publish Notices. It manages all the information about Schools, Registrations, and Login. The project is totally built at the administrative end and thus only the administrator is guaranteed access. The purpose of the project is to build an application program to reduce the manual work of managing the Schools.

The following are the important stakeholders in our project:

- 1) Admin:** Admin is the owner and the manager of software. Admin is allowed to update and modify.
- 2) Students:** They are the ones who use the software as per their requirements.
- 3) Developer:** Builds the whole application.
- 4) Tester:** This is the one who tests the software. In our project, we developed as well as tested our own application.

# **CHAPTER 8**

## **Gantt Chart**



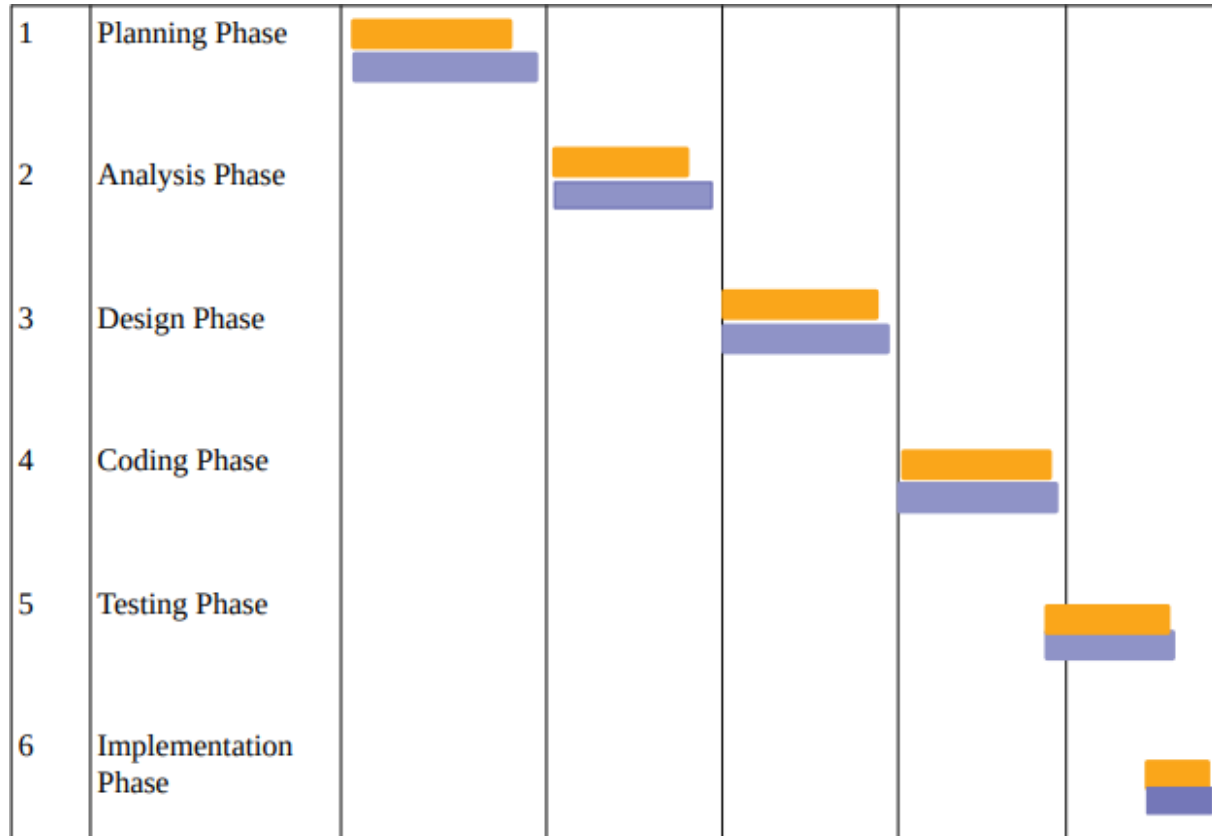
Gantt chart provides a graphical illustration of a schedule that helps to plan, coordinate, and track specific tasks in a project.

A Gantt chart is constructed with a horizontal axis representing the total time span of the project, broken down into increments (months), and a vertical axis representing the tasks that make up the project's School Management System.

It depicts our Project Development Road Map based on task scheduling right from Preliminary Research up to Project Submission. Timelines are indicated that show the time required for the completion of each and every module. It gives a clear idea as to how our project development is phased out on modules to summarize the entire Project Assessment.

# Gantt Chart

Sr No.                      Oct-2023   Nav-2022   Dec-2033   Jan-2023   Jun-2023   May-2023



# **CHAPTER 9**

# **UML DIAGRAM**

The Unified Modeling Language is a standard visual modeling language intended to be used for modeling business and similar processes, analysis, design, and implementation of software-based systems.

UML is a common language for business analysts, software architects, and developers used to describe, specify, design, and document existing or new business processes, structure, and behavior of artifacts of software systems.

UML defines various kinds of diagrams to cover most of the aspects of a system. There are two broad categories of diagrams and they are again divided into subcategories –

## **Structural Diagrams:**

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram, which form the main structure and are therefore stable.

1. Class diagram
2. Object diagram
3. Component diagram
4. Deployment diagram

## **Behavioral Diagrams:**

Behavioral diagrams basically capture the dynamic aspect of a system. The dynamic aspect can be further described as the changing/moving parts of a system.

UML has the following five types of behavioral diagrams:

1. Use - Case Diagram
2. Sequence Diagram
3. Collaboration Diagram
4. State - Chart Diagram
5. Activity Diagram

## **9.1 E-R Diagram**

The ER (Entity Relational Model) is a high-level conceptual data model diagram. Entity- Relation model is based on the notion of real-world entities and the relationship between them.

ER, modeling helps to analyze data requirements systematically to produce a well-designed database. So, it is ideal to complete ER modeling before implementing your database.

The entity relationship diagram displays the relationships of entity sets stored in a database. This model is based on three basic concepts:

### **1 Entity**

A real-world thing either living or nonliving that is easily recognizable and unrecognizable.

### **2 Attributes**

It is a single-valued property of either an entity type or a relationship- type. However, there exist multi-valued attributes too.

### **3 Relationships**

Relationship is nothing but an association among two or more entities.

### 9.1.1. Cardinalities:

Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of other set via a relationship set .

**One-to-One** -.One entity from entity set A can be associated with at most one entity of entity set B and vice-versa .

**One – to - Many** -. One entity from entity set A can be associated with more than one entity of entity set B however an entity from entity set B, can be associated with at most one entity.

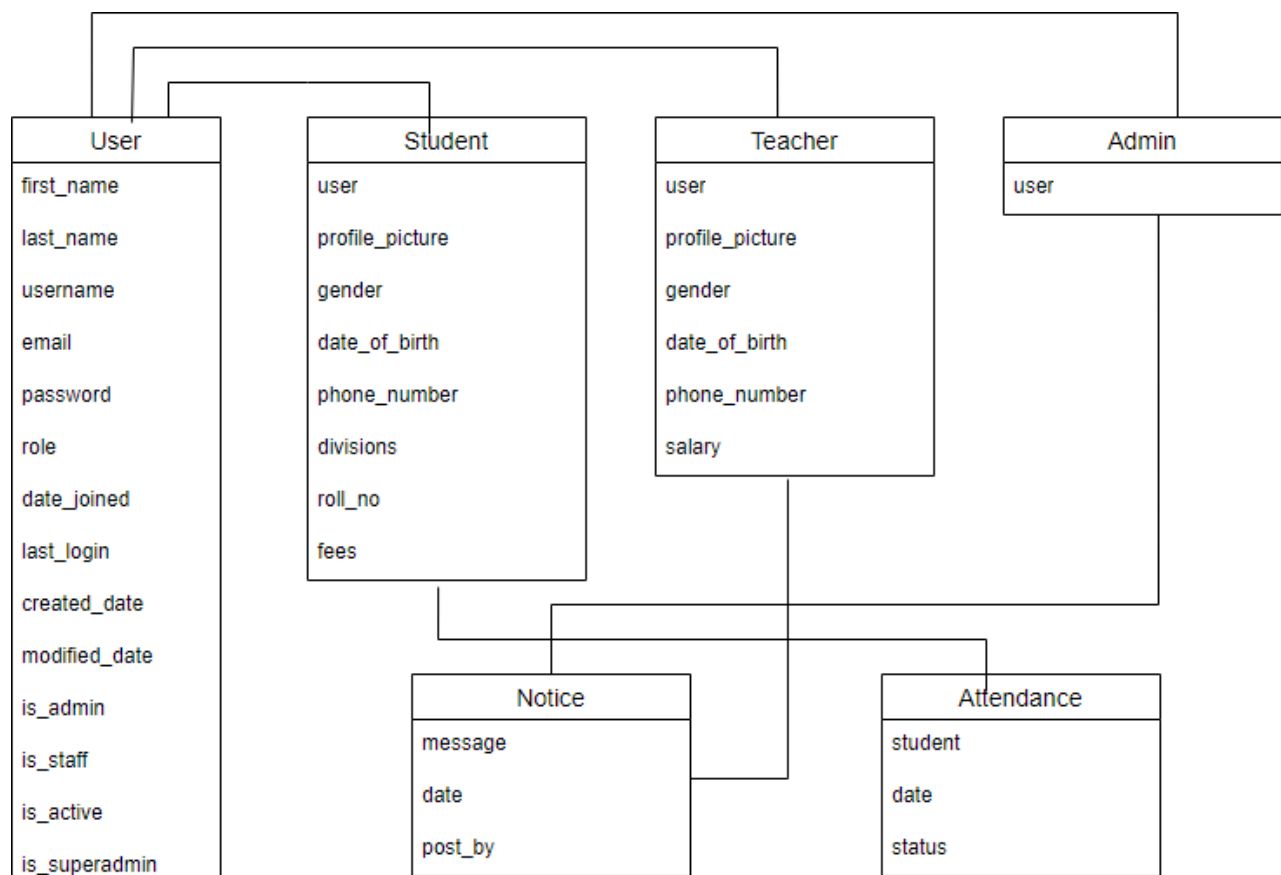
**Many – to - One** -.One entity from entity set A can be associated with more than one entity of entity set B however an entity from entity set B, can be associated with at most one entity.

**Many – to - Many** - One entity from A can be associated with more than one entity from B and vice versa.

# E-R Diagram

An entity relationship diagram ( ERD ) shows the relationships of entity sets stored in a database . An entity in this context is an object, a component of data . An entity set is a collection of similar entities . These entities can have attributes that define its properties .

By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases .





## **Use - Case Diagram**

A use case is a methodology used in system analysis to identify, clarify and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. The method creates a document that describes all the steps taken by a user to complete an activity.

Use cases are typically written by business analysts and can be employed during several stages of software development, such as planning system requirements, validating design, testing software and creating an outline for online help and user manuals. A use case document can help the development team identify and understand where errors may occur during a transaction so they can resolve them.

Every use case contains three essential elements:

The actor. The system user can be a single person or a group of people interacting with the process .

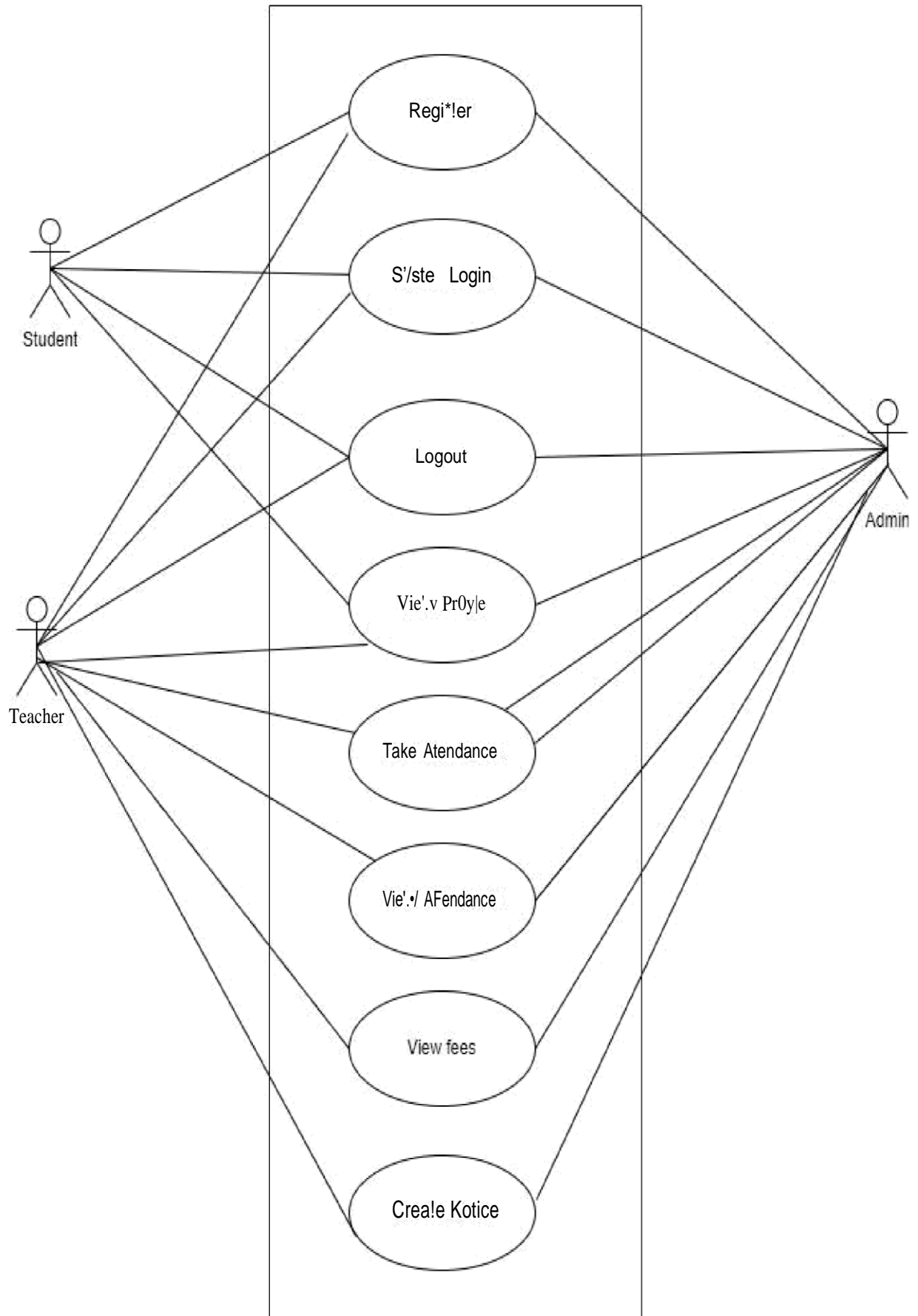
The goal. The final successful outcome that completes the process .

The system. The process and steps taken to reach the end goal, including the necessary functional requirements and their anticipated behaviors .

A use case should display the following characteristics :

Organizes functional requirements.

Models the goals of system/actor interactions .



# Class Diagram

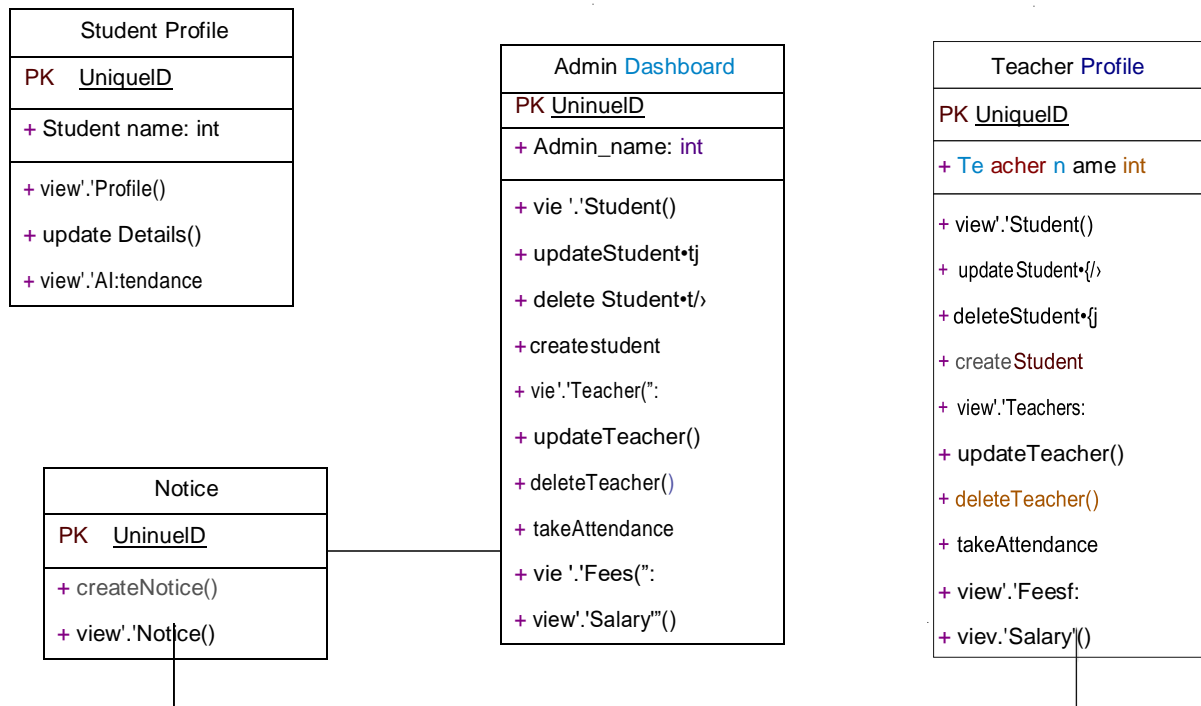
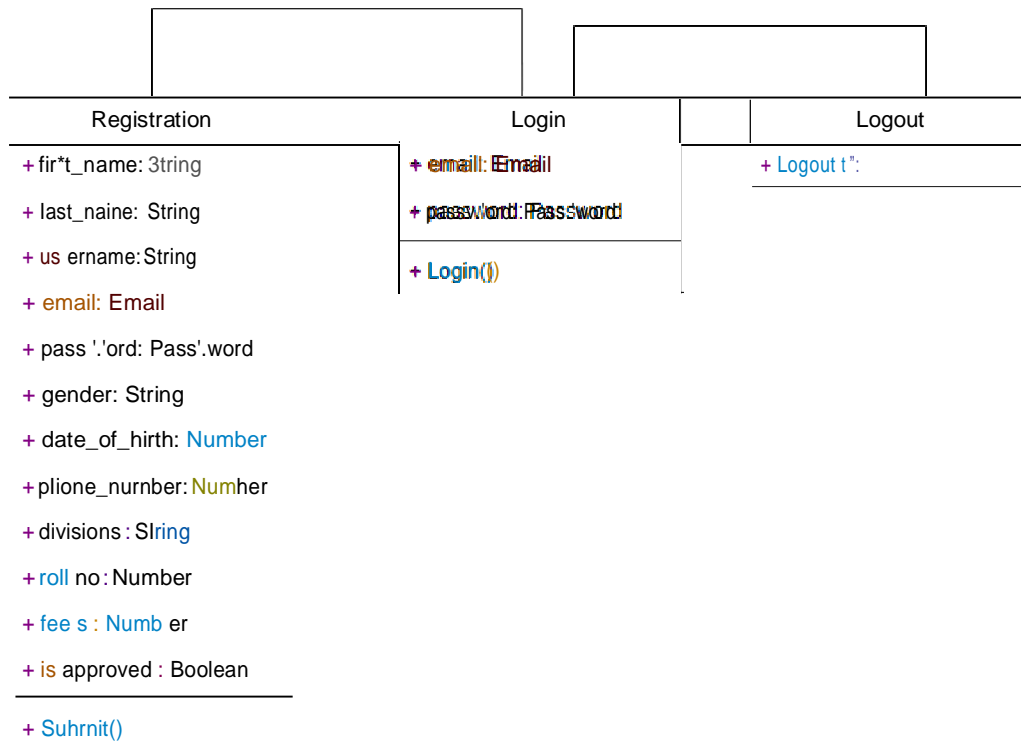
Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects .

A UML class diagram is made up of :

1. Set of classes
2. Set of relationships between classes

A class notation consists of three parts :

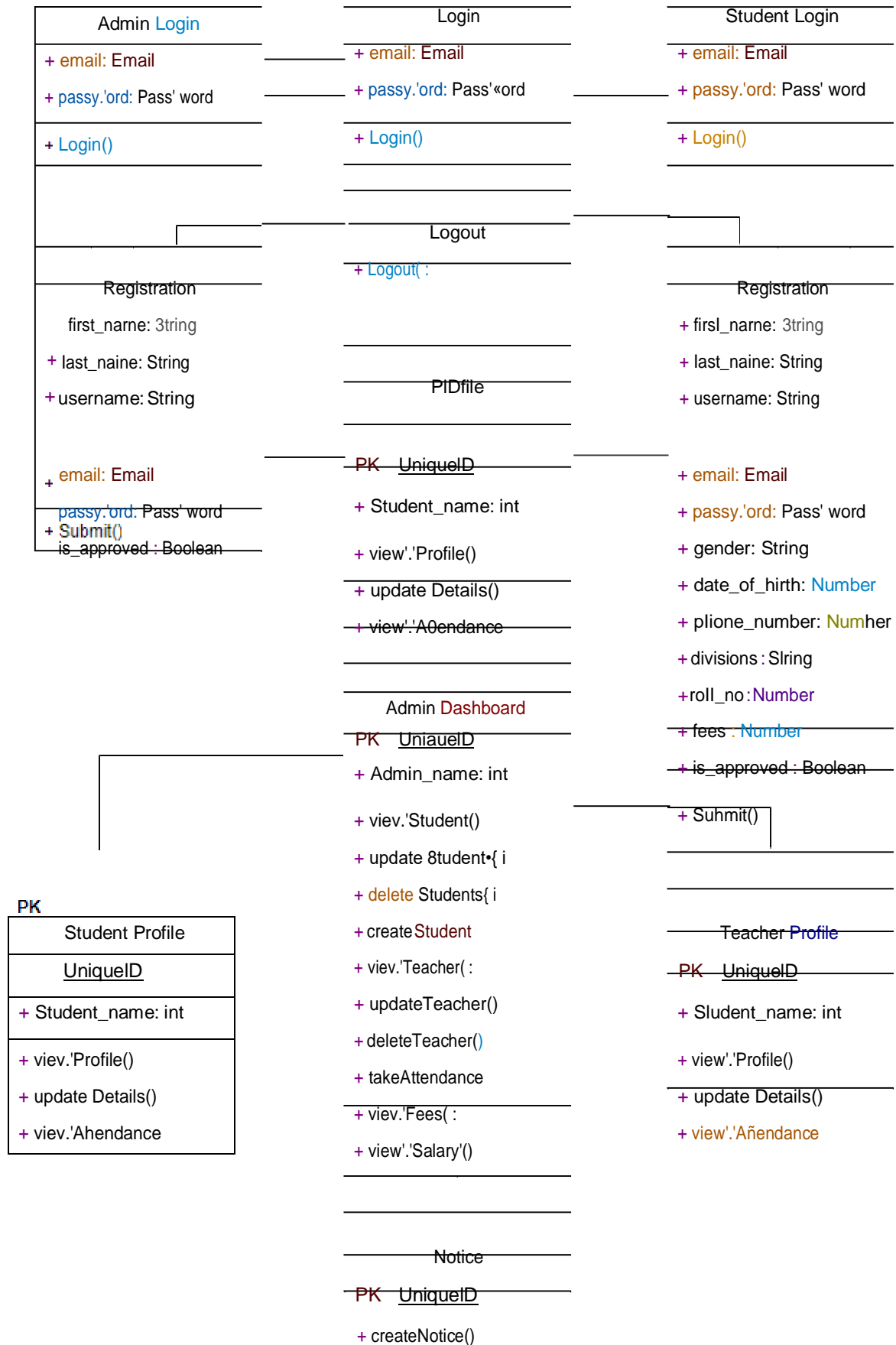
- 1. Class Name :** The name of the class appears in the first partition.
- 2. Class Attributes :** Attributes are shown in the second partition. The attribute type is shown after the colon. Attributes map onto member variables (data members) in code .
- 3. Class Operations (Methods) :** Operations are shown in the third partition. These are the services a class provides .The return type of a method is shown after the colon at the end of the method signature .The return type of method parameters are shown after the colon following the parameter name . .



## **Object Diagram**

Object is an instance of a particular moment in run time, including objects and data values. A static UML object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time, thus an object diagram encompasses objects and their relationships at a point in time. It may be considered a special case of a class diagram or a communication diagram.

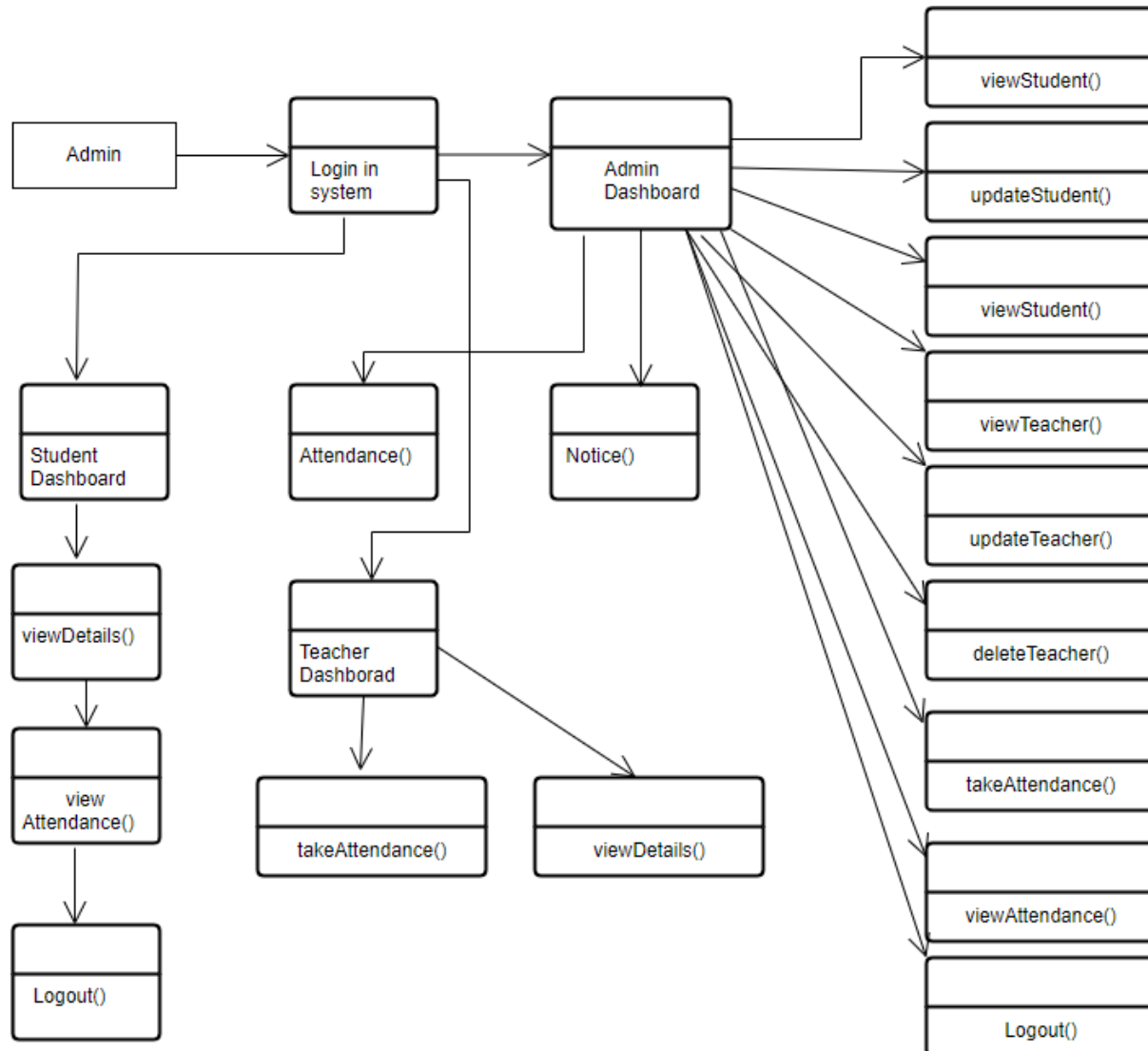
Object Diagram is used to verify the accuracy and completeness of the class diagram. An object diagram shows this relation between the instantiated classes and the defined class, and the relation between these objects in the system. They are useful to explain smaller portions of your system, when the system class diagram is complex.



## **Data Flow Diagram**

Data Flow Diagrams (DFD) are used to graphically represent the flow of Data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. A Data Flow Diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles, arrows, etc. to show data inputs, outputs, storage points and the routes between each destination .

# Data Flow Diagram





# Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function .

These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems .

## Sequence Diagram Notations

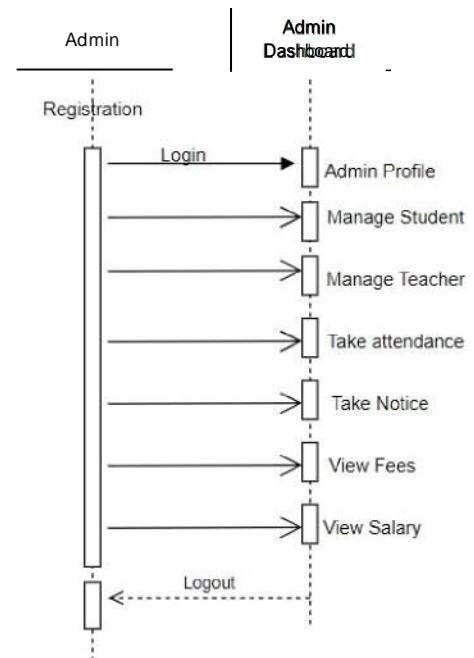
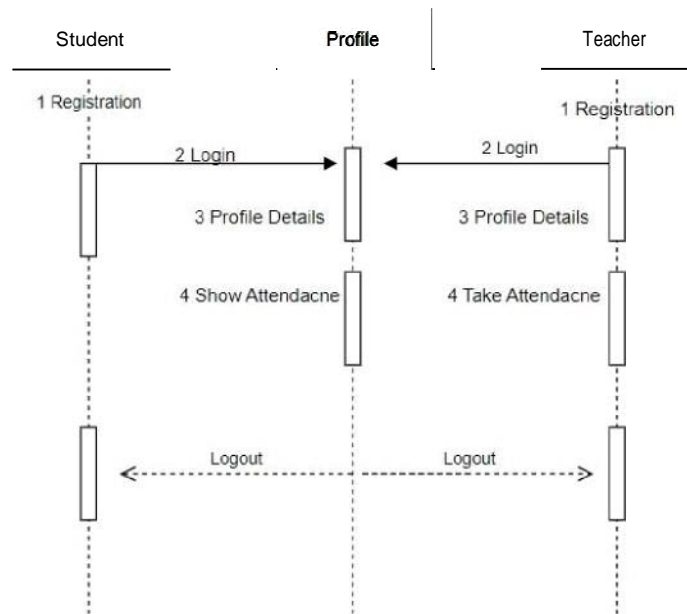
**Actors** - An actor in a UML diagram represents a type of role where it interacts with the system and its objects.

**Lifelines**- A lifeline is a named element which depicts an individual participant in a sequence diagram .

**Messages** - Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline.

## Use of Sequence Diagrams

1. Used to model and visualize the logic behind a sophisticated function, operation or procedure .
2. They are also used to show details of UML use case diagrams .
3. Used to understand the detailed functionality of current or future systems ..



# **Activity Diagram**

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

Activity is a particular operation of the system. They are used to construct the executable system by using forward and reverse engineering techniques .

## **The purpose of an activity diagram is :**

1. To Draw the activity flow of a system .
2. Describe the sequence from one activity to another .
3. Describe the parallel, branched and concurrent flow of the system .

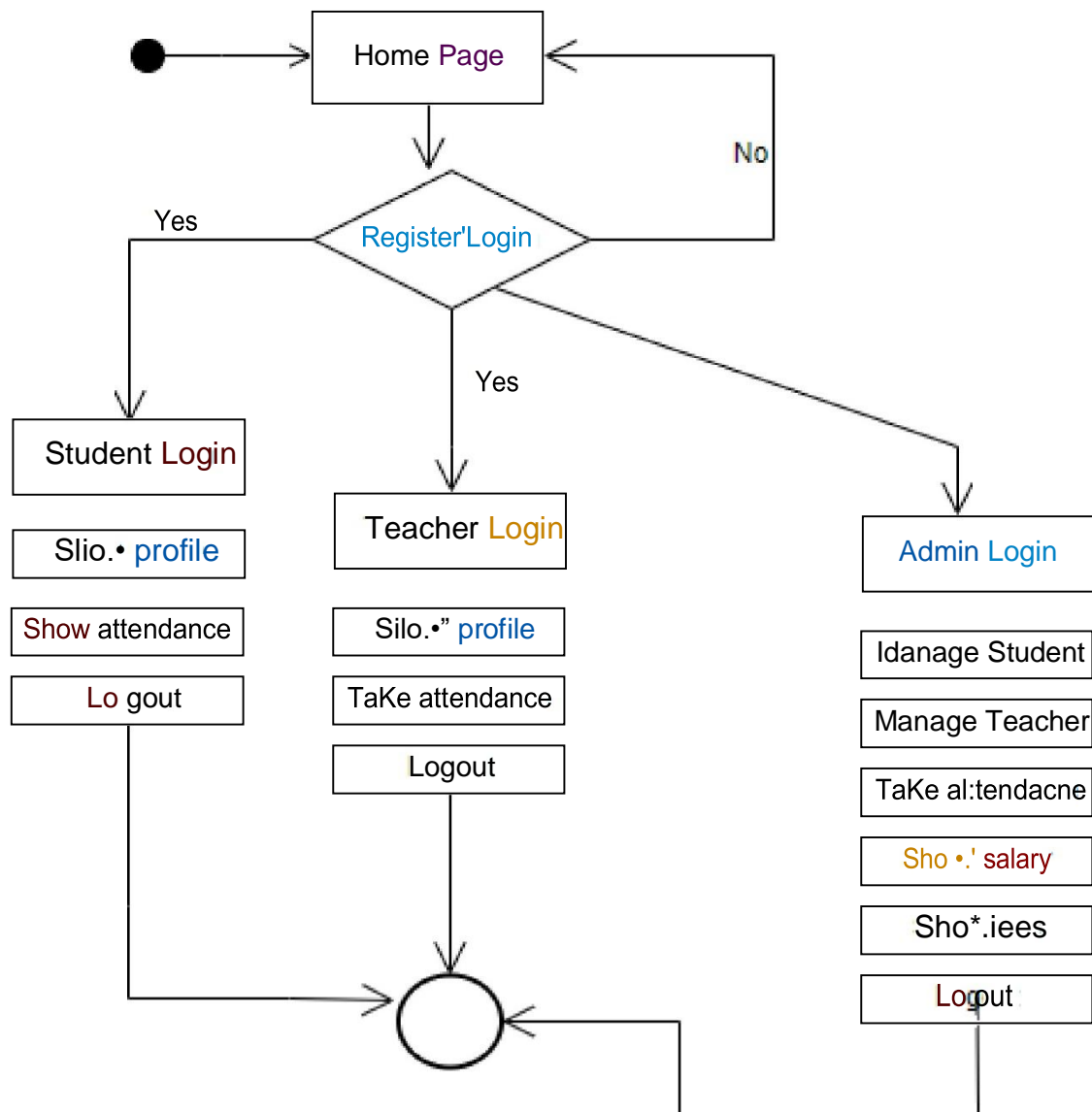
## **Activity Diagram consists of following elements**

4. Activities
5. Association
6. Conditions
7. Constraints

Activity diagram is suitable for modeling the activity flow of the system. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be databases, external queues, or any other system.

Activity diagram can be used for –

1. Modeling workflow by using activities .
2. Modeling business requirements .
3. High level understanding of the system's functionalities .
4. Investigating business requirements at a later stage .



# **Context Data Flow Diagram**

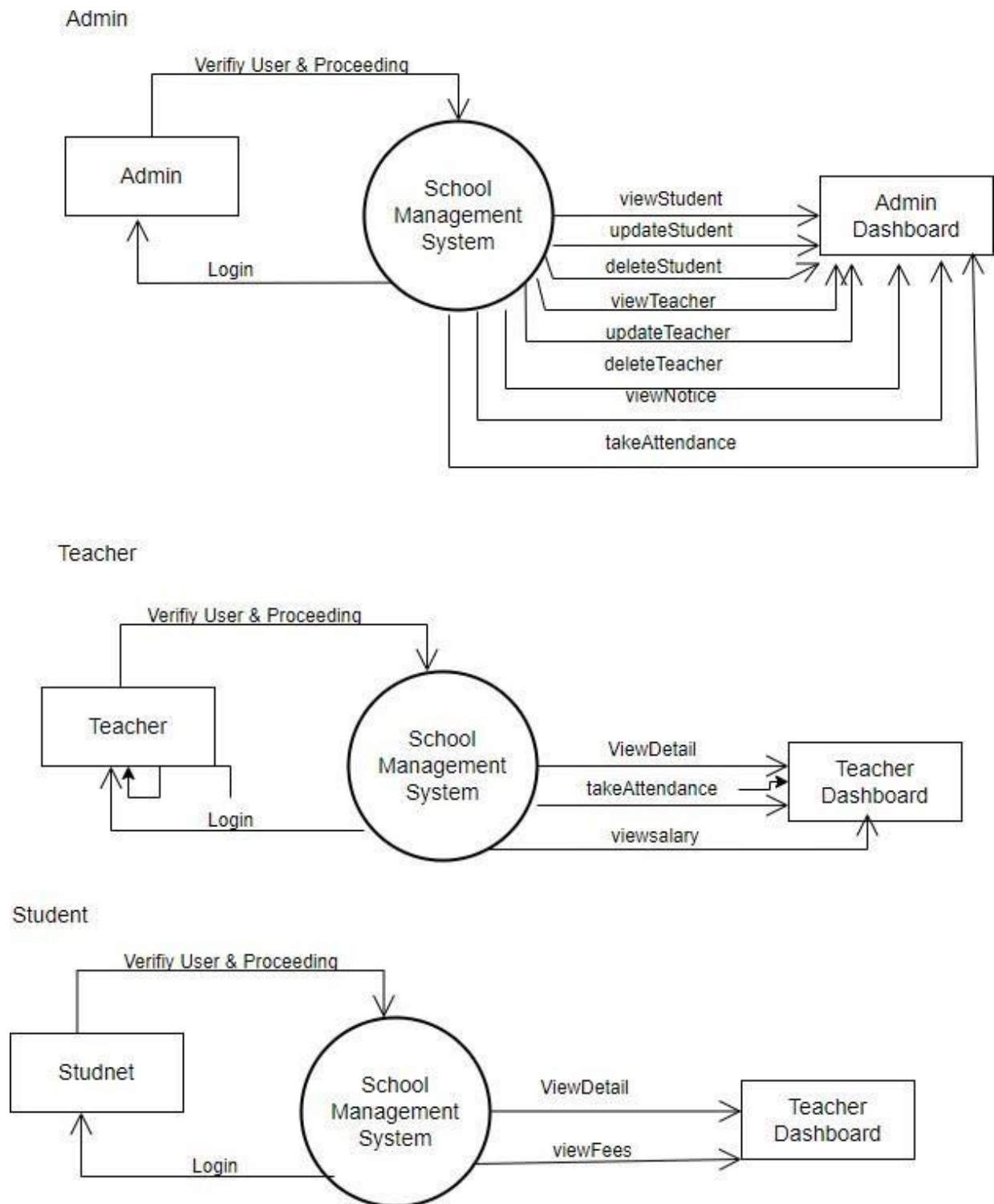
A context diagram, sometimes called a level 0 data-flow diagram, is drawn in order to define and clarify the boundaries of the software system. It identifies the flows of information between the system and external entities. The entire software system is shown as a single process

System context diagrams show a system, as a whole and its inputs and outputs from/to external factors. According to Kossiakoff and Sweet (2011)

System Context Diagrams represent all external entities that may interact with a system. Such a diagram pictures the system at the center, with no details of its interior structure, surrounded by all its interacting systems, environments and activities. The objective of the system context diagram is to focus attention on external factors and events that should be considered in developing a complete set of systems requirements and constraints.

System context diagrams are used early in a project to get agreement on the scope under investigation. Context diagrams are typically included in a requirements document. These diagrams must be read by all project stakeholders and thus should be written in plain language, so the stakeholders can understand items within the document .

# Context Data Flow Diagram



# **CHAPTER 10**

## **SQLite Data Structures**

### **ScreenShots**




## User :


### School\_user

#### Column

 id INTEGER

 password varchar(128)


 first\_name varchar(50)

 last\_name varchar(50)

 username varchar(50)

 email varchar(100)


 role smallint unsigned

 date\_joined datetime


 last\_login datetime


 created\_date datetime

 modified\_date datetime

 is\_admin bool

 is\_staff bool











 is\_active bool

 is\_superuser bool

## Student :

### School\_student

#### Column


-  id INTEGER
-  profile\_picture varchar(100)
-  gender varchar(10)
-  date\_of\_birth date
-  phone\_number INTEGER
-  divisions varchar(10)
-  roll\_no integer unsigned
-  fees integer unsigned
-  is\_approved bool
-  user\_id bigint

## Teacher :

### School\_teacher


#### Column

 id INTEGER

 profile\_picture varchar(100)


 gender varchar(10)

 date\_of\_birth date

 phone\_number INTEGER

 salary integer unsigned

 is\_approved bool

 user\_id bigint

## Attendance :

### School\_attendance

#### Column

 id INTEGER

 status varchar(10)

 student\_id bigint

 date datetime

## Notice :

### School\_notice

#### Column

 id INTEGER

 message varchar(100)

 date datetime

 post\_by\_id bigint

# **CHAPTER 11**

## **Project Screenshots**

# Home Page

School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#)



# Blog Section

## Features



Student



Teacher



Admin

## Blogs

The "School Management System" has been developed to override the problems prevailing in the practicing manual system.

This software is supported to eliminate and sometimes reduce the hardships faced by this existing system.

Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.



# Footer Section

## Highlights



Highly Reliable



Safe & Secure



Easy To Use



Cost Effective



Customer Support

## Contact

### School Management System

The "School Management System" has been developed to override the problems prevailing in the practicing manual system.

### MENU

Features  
Blogs  
Contact

### GET IN TOUCH

✉ schoolmanagement@gmail.com  
📍 Mumbai, Maharashtra  
☎ +91 1010101010

Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.



# Student Section

School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#)



## Student Panel

Welcome To School Management System

please select any flow to continue

[Sign up](#)

[Login](#)

Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.

# Student Registration

School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#)

## New Admission

First Name	Last Name
<input type="text"/>	<input type="text"/>
Username	Email Address
<input type="text"/>	<input type="text"/>
Enter Password	Confirm Password
<input type="password"/>	<input type="password"/>
Select Gender	Date Of Birth
<input type="text" value="Male"/>	<input type="text" value="18-06-2023"/>
Select Divisions	Roll No
<input type="text" value="One"/>	<input type="text"/>
Fees	Phone Number
<input type="text" value="1,000"/>	<input type="text"/>
Profile Picture	
<input type="button" value="Choose File"/> No file chosen	

*Already Have An Account ? [Login here](#)*

Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.

# Student Login

School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#)

## Student Login

---

Email Address

Password

Don't Have An Account ? [Singup here](#)

[Forgot Password ?](#)

Login

---

Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.

# Student Dashboard

School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#) [Logout](#)



Mohammad Zeeshan

[Dashboard](#)

[Attendance](#)

[Log Out](#)

Division

Ten

Roll No

1

Fees

10000

Contact Number

7400229131

Gender

Male

Date OF Birth

March 3, 2002

## Notice

Apr 16, 2023

By - Admin

This is admin notice

Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.

*Please enter date /or attendance*



Mohammad Zeeshon

[& Dashboard](#)

[@ Log Out](#)

Developed By [zeeshonn](#) | Copyright © 2023 All rights are reserved.

*Your Attendance of date **xl ••“l M, 2023***



Mohommod Zeeshon

[& Doshbeord](#)

[@ Log Out](#)

studentluome  
MohomIuodZeeehon

Present - Absent  
Absen

Developed By [zeeshonn](#) | Copyright © 2023 All rights are reserved.

# Teacher Section

School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#)



## *Teacher Panel*

Welcome To School Management System

please select any flow to continue

[Apply Now](#) [Login](#)

Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.

# Teacher Registration / Login

School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#)

## Apply Teacher job

First Name	Last Name
<input type="text"/>	<input type="text"/>
Email Address	Username
<input type="text"/>	<input type="text"/>
Enter Password	Confirm Password
<input type="password"/>	<input type="password"/>
Select Gender	Date Of Birth.
<input type="text" value="Male"/>	<input type="text" value="19-06-2023"/>
Salary	Phone Number
<input type="text"/>	<input type="text"/>
Profile Picture	
<input type="button" value="Choose File"/> No file chosen	

*Already Have An Account ? [Login here](#)*

Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.

School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#)

## Teacher Login

Email Address	Password
<input type="text"/>	<input type="password"/>

*Don't Have An Account ? [Singup here](#)*

[Forgot Password ?](#)

Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.

# Teacher Dashboard

School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#) [Logout](#)



Pratap Kapoor

 **Dashboard**

 Attendance

 Notice

 Log Out

Name

Pratap

Contact Number

9590569546

Join date

June 16, 2023

Salary

100000

Gender

Male

Date OF Birth

June 16, 2023

## Notice

April 16, 2023

By - Admin

This is admin notice



Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.



### Attendance of class *Ten*

Student Name	Present   Absent
vo honnrrnod zeeshon	<b>Present</b>
<div><div></div><div>Enter date in dd/mm/yyyy</div></div>	

#### Pratap Kapoor

@ Dashboard

Attendance

@ Notice

@ Log Out

*Please enter date for attendance of class **Ten***

Enter date (dd/mm/yyyy)

**Pratap Kapoor**

@ Dashboard

Attendance

@ Notice

@ Logout

Developed By [zeeshonn](#) | Copyright 2023 All rights are reserved.

***Attendance of class Ten and date April 15, 2023***

Student Name	Class	Present - Absent
Mohammad Zeeshan	Ten	Absent

Protop Kopaor

@ Dashboard

Attendance

@ Notice

@ Logout



Pratap Kapoor

 [Dashboard](#)

 [Attendance](#)

 [Notice](#)

 [Log Out](#)

### *Announce Something*

Write Your Message Here...

Submit

# Admin Section

School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#)



## Admin Panel

Welcome To School Management System

please select any flow to continue

[Sign up](#)

[Login](#)

Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.

# Admin Registration / Login

School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#)

## Admin Login

Email Address

Password

Don't Have An Account ? [Singup here](#)

Login

Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.

School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#)

## New Admin

First Name

Last Name

Email

Username

Enter Password

Confirm Password

Already Have An Account ? [Login here](#)


Sign Up

Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.

# Admin Dashboard







School Management System

[Home](#) [Features](#) [Blogs](#) [Contact](#) [Logout](#)




School Admin

[Dashboard](#)  
[Teacher](#)  
[Student](#)  
[Attendance](#)  
[Notice](#)  
[Salary](#)  
[Fees](#)  
[Log Out](#)

Total Teacher 1 	Total Student 1 	Teacher Salary 100000 ₹	Student Fees 10000 ₹
Pnding Teacher 0 	Pnding Student 0 	pending Salary None 	pending Fees None 

## Notice

April 16, 2023	By - Admin
This is admin notice 	

Developed By [zeeshann](#) | Copyright © 2023 All rights are reserved.



School Admin

 Dashboard

 **Teacher**

 Student

 Attendance

 Notice

 Salary

 Fees

 Log Out

All Teacher



Add Teacher



Approv Teacher



Teacher Salary







School Admin

 Dashboard

 Teacher

 **Student**

 Attendance

 Notice

 Salary

 Fees

 Log Out

All Student



Add Student



Approv Student



Student Fees





## School Admin

[@ Dashboard](#)[Teacher](#)[% Student](#)[Attendance](#)[@ Notice](#)[Salary](#)[Q Fees](#)[@ Log Out](#)*Student Attendance*

Class	Take Attendance	View Attendance
one	<a href="#">+</a>	<a href="#">👁</a>
Three	<a href="#">+</a>	<a href="#">👁</a>
Fe	<a href="#">+</a>	<a href="#">👁</a>
Seven	<a href="#">+</a>	<a href="#">👁</a>
Ten	<a href="#">+</a>	<a href="#">👁</a>



School Admin

@ Dashboard

Teacher

% Student

Attendance

@ Notice

 Salary

Fees

@ Log Out

### Attendance of class **Ten**

Student Name	Present/Absent
Mohammad Zeeshan	Present



**Please enter date for attendance Of class Ten**

Enter date mm/dd/yyyy

**Submit**

### School Admin

@ Dashboard

Teacher

% Student

Attendance

@ Notice

 **Salary**

Fees

@ Log Out



Attendance of class *Ten* and date *ARE!! 15, ZO23*

Student Nome	Closs	Present-Absent
Mohammad Zeesnan	Ten	Absent

#### School Admin

@ Dashboard

Teacher

% Student

Attendance

@ Notice

 Salary

Fees

@ Log Out

## Announce Something



Write Your Message Here .

School Admin

Submit

@ Dashboard

Teacher

% Student

Attendance

 Notice

 Salary

Q Fees

@ Log Out

## Teacher Salary



Nome  
Pratap Kapoor

**Salary**

100,0 00

### School Admin

@ Dashboard

Teacher

% Student

Attend once

@ Notice

 **Salary**

Fees

@ Log Out



### School Admin

@ Dashboard

Teacher

% Student

Attendance

@ Notice

Salary

Fees

@ Log Out

### Student Fees

Class

One

Two

Three

Five

seven

Egg hl

Nine

men

View Fees





# **Project Source Code**

## **School Management System**

# **CHAPTER 12**

## **Testing and Validations**

## **Why Is Testing Important?**

This is the most important part of the software life cycle. It provides better quality of software to end users; therefore, those end users won't come across software issues. Testing of any software is very important for validating functionality of the software. Testing will provide the following information: It finds issues during early phases, which can be fixed before finalization. It assures stability and reliability of software in different conditions. It helps to provide issue-free software for delivery. Any application must be tested with different methodologies. If the application is not tested properly, then some faulty application will be delivered to customers. Delivering such quality of application will reduce credibility, and the customers will be not delighted with application. Testing is usually conducted by development and quality assurance teams. This testing validates the functionality of the application.

# Testcase :

```
from django.test import TestCase
from .models import User, Notice, Attendance
# Create your tests here.

class UserTest(TestCase):
    def
create_user(self, first_name="demo", last_name="test", username="demo", email="demo_test@g
mail.com", role=1):
    return
User.objects.create(first_name=first_name, last_name=last_name, username=username, email=
email, role=role)

    def test_user(self):
        obj = self.create_user()
        self.assertTrue(isinstance(obj, User))
        self.assertEqual(obj.full_name(), obj.first_name+" "+obj.last_name)
        print("User created successfully : ", obj)

class NoticeTest(TestCase):
    def test_model_notice(self):
        message = Notice.objects.create(message="demo notice")
        self.assertEqual(str(message), "demo notice")

class AttendanceTest(TestCase):
    def test_model_attendance(self):
        status = Attendance.objects.create(status="present")
        self.assertEqual(str(status), "present")
```

# **CHAPTER 13**

## **System Maintenance**

System Maintenance stands for all the modifications and Updation done after the delivery of software product. The software requirements vary according to client needs. Hence the software must be customizable to be able to meet Customer Needs.

## **The Need for Modification in our System can be of the following form:**

**Market Conditions** - Policies, which changes over time, such as taxation and newly introduced constraints like, how to maintain bookkeeping, may trigger the need for modification.

**Client Requirements**- Policies, which changes over time, such as taxation and newly introduced constraints like, how to maintain bookkeeping, may trigger the need for modification.

**Host Modifications** - If any of the hardware and/or platform (such as operating system) of the target host changes, software changes are needed to keep adaptability.

**Organization Changes** - If there is any business level change at client end, such as reduction of organization strength, acquiring another company, organization venturing into new business, need to modify in the original software may arise.

**Corrective Maintenance** - Policies, which changes overtime, such as taxation and newly introduced constraints like, how to maintain bookkeeping, may trigger the need for modification.

**Adaptive Maintenance** - Policies, which changes over time, such as taxation and newly introduced constraints like, how to maintain bookkeeping, may trigger the need for modification.

**Preventive Maintenance** - It includes modifications and Updation to prevent future problems with the software. It aims to attend to problems, which are not significant at this moment but may cause serious issues in the future.

# **CHAPTER 14**

## **Future Enhancements**



## **Future Enhancements:**

Nothing is perfect in this world. So, we are also no exception. Although we have tried our best to present the information effectively, yet, there can be a further enhancement in the Application.

- We can give more advanced software for School Management System including more facilities.
- We will host the platform on online servers to make it accessible worldwide.
- In the future add an exam section and subject section and improve the UI section.
- In the future, we connect with the online system

# **CHAPTER 15**

## **Conclusion**

This Project “ **School Management System**” can play a critical role in improving academic success. By streamlining administrative tasks, improving communication, and providing access to valuable data, these systems can help create a more supportive and effective learning environment for students.



# **CHAPTER 16**

## **Bibliography**

**For Icons: [Bootstrap Icons](#)**

**For Coding Reference: [Python](#), [Django](#)**

**For UML Diagrams: [Diagram](#)**