**Logo

Description automatically generated**

**Attendance Management System**

A

PROJECT REPORT

SUBMITTED TO THE

VIT-AP UNIVERSITY FOR FULFILLMENT DEGREE

OF

**BTECH**

IN

COMPUTER SCIENCE & ENGINEERING

**GUIDED BY**

**Dr. Anil Vitthalrao Turukmane**

**(Professor, SCOPE)**

**SUBMITTED BY**

**Rampam Greeshma Geethika(20BCD7094)**

**SCHOOL OF COMPUTER SCIENCE & ENGINEERING, VIT-AP UNIVERSITY**

**DECEMBER 2022**

**INDEX:**

|  |  |
| --- | --- |
| **Contents** | **Page No.** |
| Abstract | 1 |
| Introduction | 2-3 |
| Project Overview | 3-4 |
| UML Diagrams | 5-13 |
| Test Cases | 11-14 |

**UML Diagrams contents:**

|  |  |
| --- | --- |
| **UML Diagram** | **Page No.** |
| Data Flow Diagram | 5-6 |
| Use Case Diagram | 7 |
| Sequence Diagram | 8 |
| Class Diagram | 9 |
| Component Diagram | 10 |
| Deployment Diagram | 11 |

**Software Requirement System (SRS)**

* **ABSTRACT:**

The attendance Management system deals with the maintenance of students’ and staff’s attendance details. It generates attendance of students on basis of presence in class. It is maintained daily of their attendance. The staff will be provided with a separate username & password to make the student’s status.

The staff handling the subjects responsible to make attendance for all students. The attendance will be calculated only if the student is present during that period. The student’s attendance report based on weekly and consolidated will be generated. The staff’s attendance report based weekly and consolidated will be generated.

* **INTRODUCTION:**
* **OBJECTIVE:**

“Attendance Management System” is a software developed for maintaining the attendance of the student on the daily basis in the college. Here the staff, who are handling the subject, will be responsible to mark the attendance of students. Each staff will be given a separate username and password based on the subject they handle. An accurate report based on student attendance is generated here. This system will also help in evaluating the attendance eligibility criteria of a student. A report of the student’s attendance on a weekly and monthly basis is generated.

* **SYSTEM ANALYSIS:**

Analysis can be defined as breaking up any whole to find out its nature, function, etc. It defines design as making preliminary sketches of; to sketch a pattern or outline for a plan. To plan and carry out, especially by the artistic arrangement or in a skillful wall. System analysis and design can be characterized as a set of techniques and

processes, a community of interests, a culture, and an intellectual orientation.

The various tasks in the system analysis include the following.

* Understanding application.
* Planning.
* Scheduling.
* Developing candidate solutions.
* Performing trade studies.
* Performing cost-benefit analysis.
* Recommending alternative solutions.
* Selling of the system.
* Supervising, installing, and maintaining the system.

* **EXISTING SYSTEM:**

The Existing system is a manual entry for the students. Here the attendance will be carried out in the handwritten registers. It will be a tedious job to maintain the record for the user. The human effort is more here. The retrieval of the information is not as easy

as the records are maintained in the handwritten registers.

* **PROPOSED SYSTEM:**

To overcome the drawbacks of the existing system, the proposed system has evolved. This project aims to reduce the paperwork and save time to generate accurate results from the student’s attendance. The system provides the best user interface.

Efficient reports can be generated by using this proposed system.

* Advantages of the Proposed System
* It is trouble-free to use.
* It is a relatively fast approach to enter the attendance
* Is highly reliable, approximate result from user
* Best user Interface
* Efficient reports
* **FEASIBILITY STUDY:**

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which possibly give an indication of how the new system should look like. This is where creativity and imagination are used.

The solution should provide enough information to make reasonable estimates about project costs and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need

significantly change the original goal.

The feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable.

* **PROBLEM DEFINITION:**

This system developed will reduce manual work and avoid redundant data. By maintaining attendance manually, then efficient reports cannot be generated. The system can generate an efficient weekly, a consolidated report based on attendance. As the attendance is maintained in registers it has been a tough task for admin and staff to maintain for a long time. Instead, the software can keep long and retrieve the information when needed.

* **PROJECT OVERVIEW:**

The attendance Management System has two main modules for proper

functioning Admin module is sights for creating any new entry of faculty and student details. The user has the right of making daily attendance and generating reports. Attendance reports are taken by giving details of students details, dates, and classes.

* **MODULE DESCRIPTION:**

The system should be designed in such a way that only authorized people should be allowed to access some particular modules. The records should be modified by only administrators and no one else. The user should always be in control of the application and not the versa. The user interface should be consistent so that the user can handle the application with ease and speed. The application should be visual, contained is actually clear.

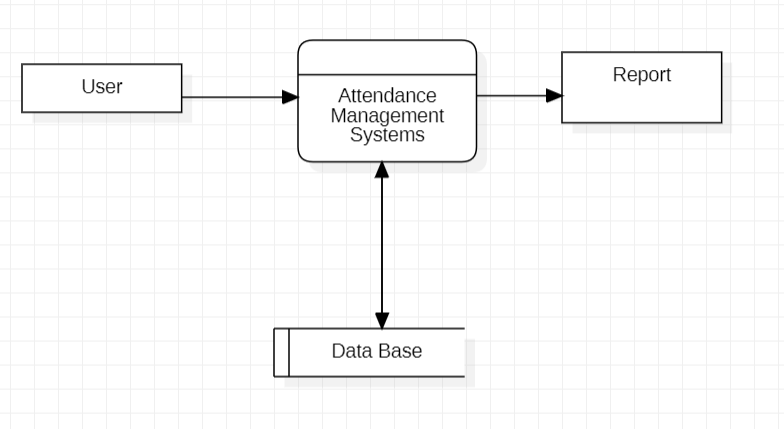
* **ADMINISTRATOR MODULE:**
* ***Student Details:***
* This module deals with the allocation of roll no and personal details for a new batch. It will generate personal details of students and academic details of the students with the photos.
* ***Staff Details:***
* It helps to allow the subject and the subject code to the particular staff.
* It provides the facility to have a username and password for the staff.
* ***Timetable details:***
* It will retrieve the subject information from the subject database and assign time table to the staff.
* It will help the admin, and staff to make the entry of attendance based of the subject and period allotted to the respective staff.
* ***Attendance details:***
* It will be made to the attendance database of all students. Entered attendance to store in the database subject, period wise into the particular date.
* It will help to get a report weekly and consolidate attendance.
* ***Report details:***

The report can be taken daily, weekly and consolidated:

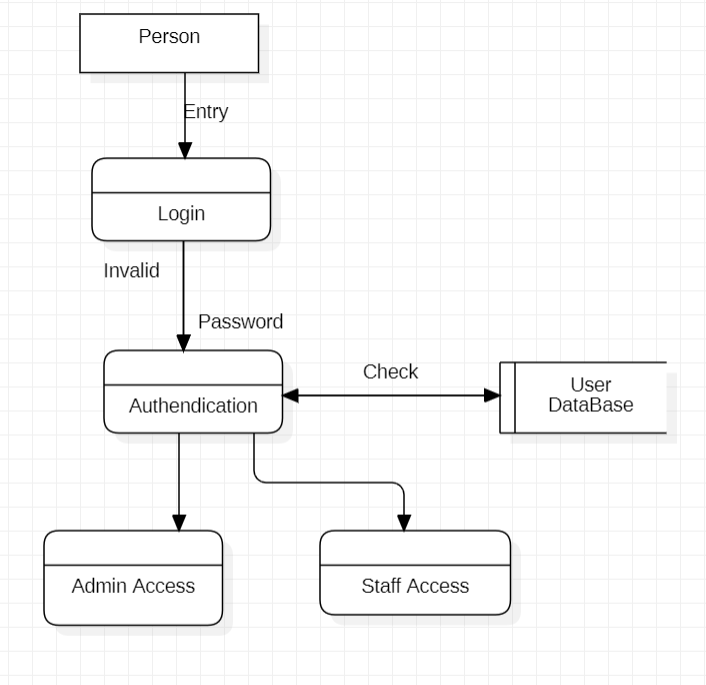
* weekly report get all hour details of attendance starting date to ending date and display the status
* Consolidate report to get all student attendance details starting date to ending date status to help with the student’s eligibility criteria to attend the examination.
* **STAFFS MODULE:**
* ***Attendance details:***
* It assists the staff to mark attendance to the students for their subject. This will authenticate the staff before making the entry.
* ***Report details:***
* weekly report gets particular hour details of attendance from starting date to ending date and displays the status.
* consolidate report to get all student attendance details from starting date to ending date status to help with the student’s eligibility criteria to attend the examination
* **DATA FLOW DIAGRAMS(DFD):**

A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method.

***DFD level 0:***

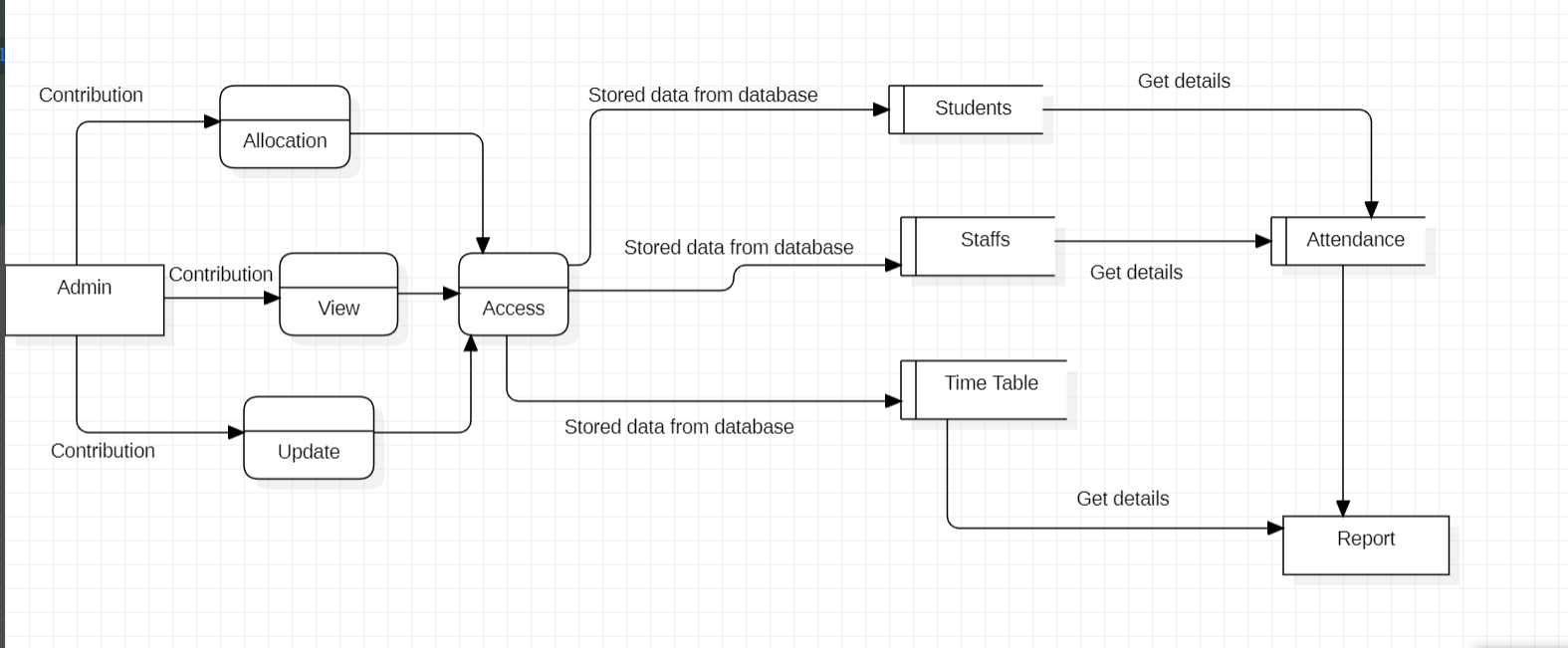
****

***DFD level 1:***

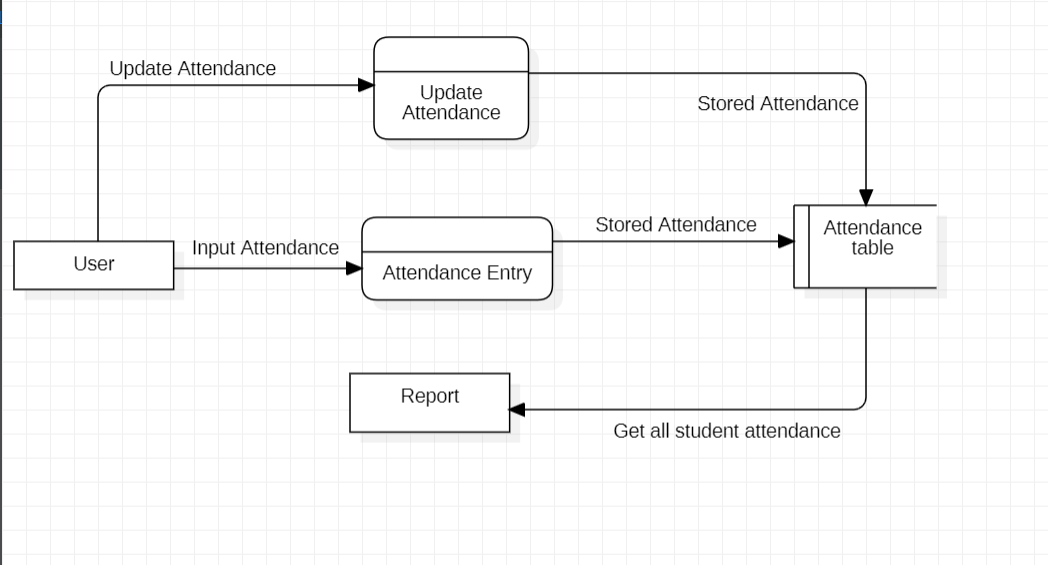
****

***DFD level 2:***

***Admin:***

****

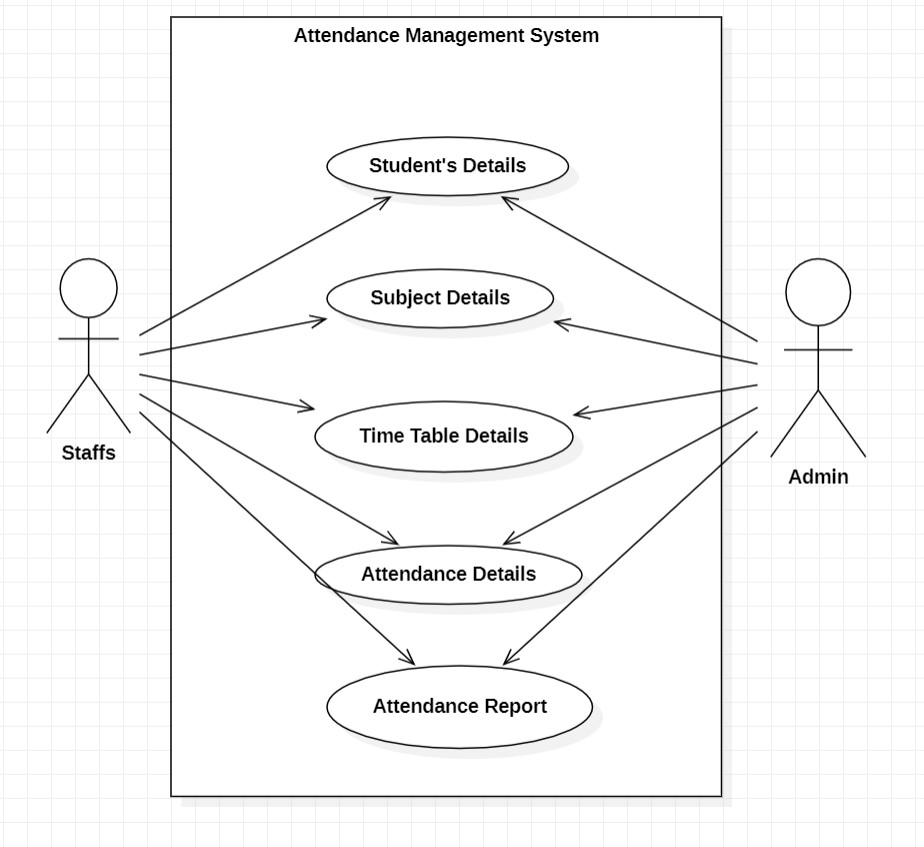
***staffs:***

****

* ***USE CASE DIAGRAM:***

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system.

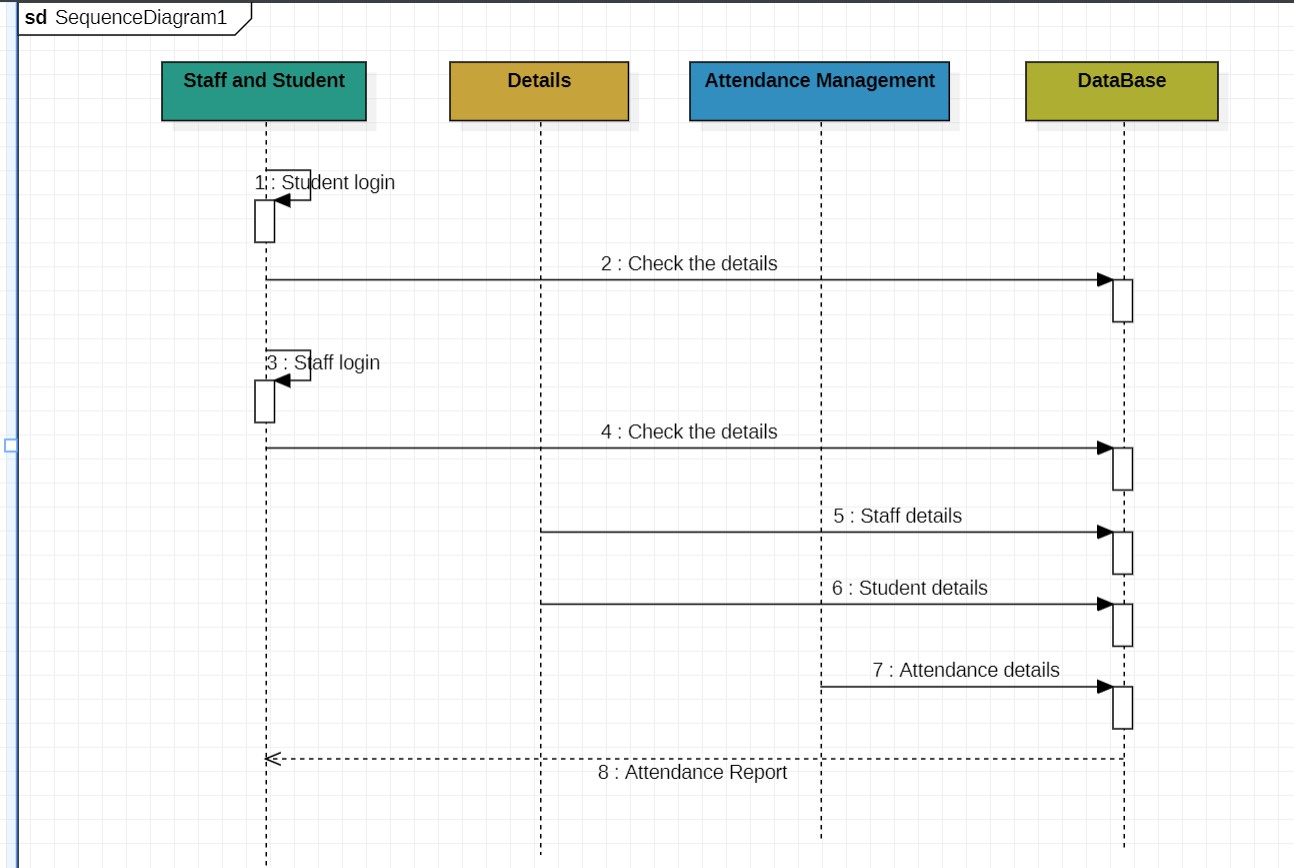
A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.



* **UML SEQUENCE DIAGRAM:**

A sequence diagram is an interaction diagram. From the name, the diagram deals with some sequences, which are the sequence of messages flowing from one object to another.

Interaction among the components of a system is very important from an implementation and execution perspective. The sequence diagram is used to visualize the sequence of calls in a system to perform specific functionality.

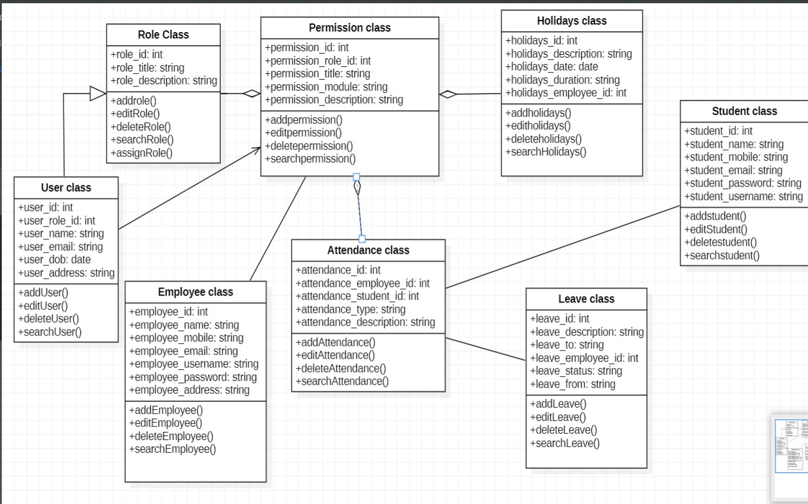


* **CLASS DIAGRAM:**

Class diagrams are the most common diagrams used in UML. A class diagram consists of classes, interfaces, associations, and collaboration. Class diagrams basically represent the object-oriented view of a system, which is static in nature.

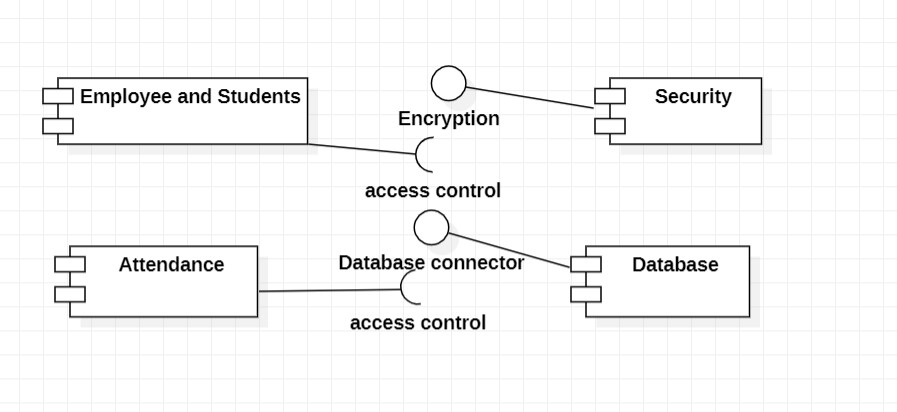
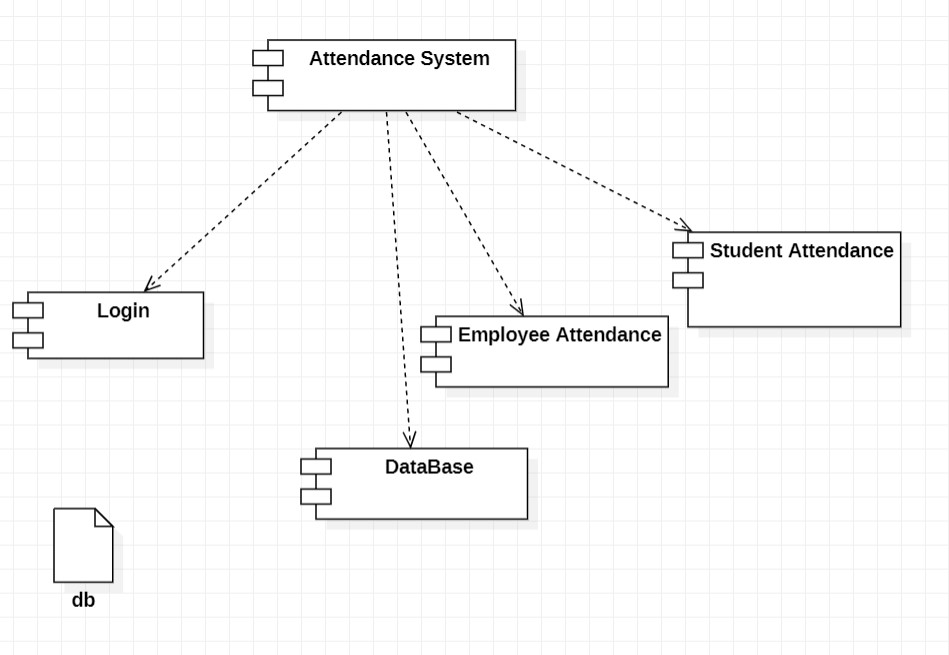
Active class is used in a class diagram to represent the concurrency of the system.

The class diagram represents the object orientation of a system. Hence, it is generally used for development purposes. This is the most widely used diagram at the time of system construction.



* **COMPONENT DIAGRAM:**

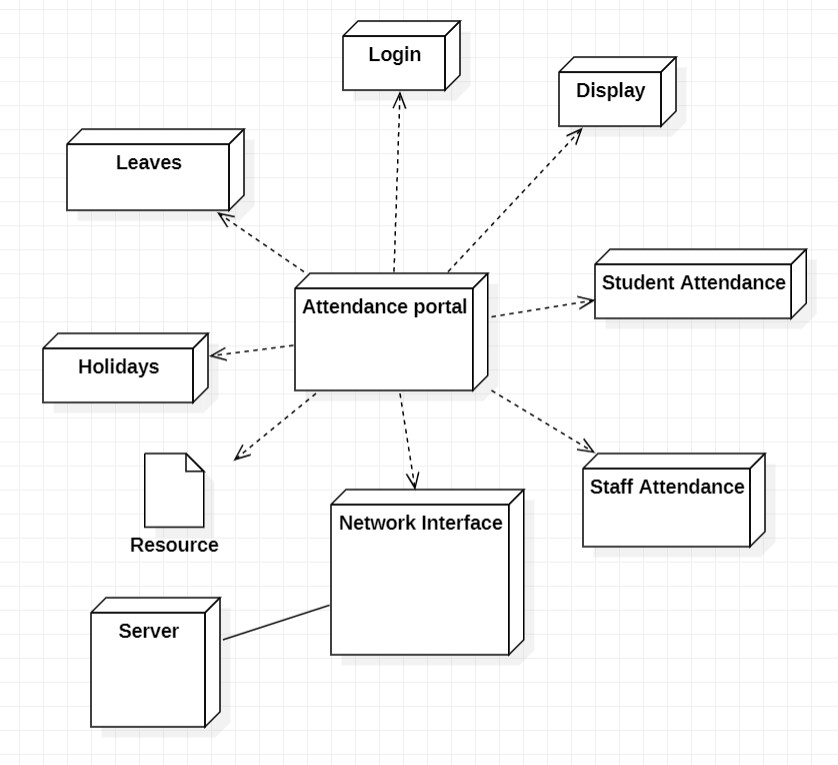
The purpose of a component diagram is to show the relationship between different components in a system. For the purpose of UML 2.0, the term "component" refers to a module of classes that represent independent systems or subsystems with the ability to interface with the rest of the system.

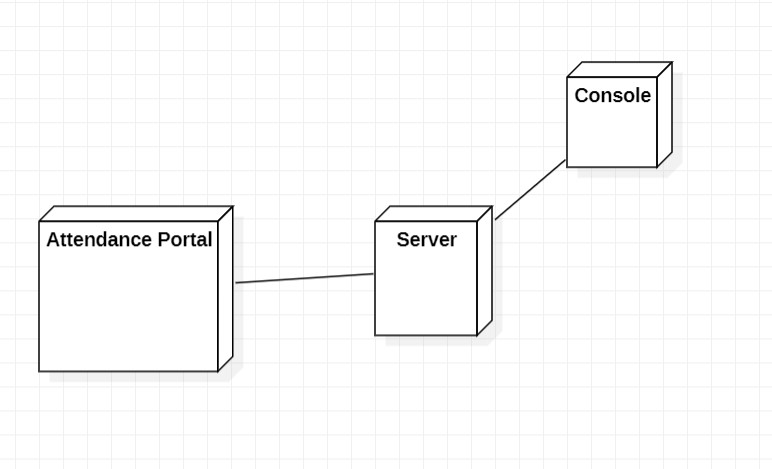


* **DEPLOYMENT DIAGRAM:**

Deployment diagrams are a set of nodes and their relationships. These nodes are physical entities where the components are deployed.

Deployment diagrams are used for visualizing the deployment view of a system. This is generally used by the deployment team.





* **TEST CASES:**

***Introduction***

Once source code has been generated, software must be tested to uncover (and correct) as many errors as possible before delivery to the customer. Our goal is to design a series of test cases that have a high likelihood of finding errors. To uncover the errors software techniques are used. These techniques provide systematic guidance for designing a test that

1. Exercises the internal logic of software components and
2. Exercises the input and output domains of the program to uncover errors. In program function, behavior, and performance.

***Steps:*** Software is tested from two different perspectives:

1. Internal program logic is exercised using ―White box test case design Techniques.
2. Software requirements are exercised using ―block box test case design techniques.

In both cases, the intent is to find the maximum number of errors with the Minimum amount of effort and time.

***Testing Methodologies:***

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and a set of milestones for the manager. Because the test strategy steps occur when deadline pressure begins to rise, progress must be measurable and problems must surface as early as possible. The following testing techniques are well known and the same strategy is adopted during this project testing.

* **Unit testing:**

Unit testing focuses verification effort on the smallest unit of software design- the software component or module. The unit test is white-box oriented. Unit testing is implemented in every module of the student attendance management System. by giving correct manual input to the system, the data are stored in the database and retrieved. If you want the required module to access input or get the output from the End user. any error will accrue the time will provide a handler to show what type of error will accrue.

* **System testing:**

System testing is a series of different tests whose primary purpose is to fully exercise the computer-based system. Below we have described the two types of testing which have been taken for this project. it is to check all modules worked on an input basis. if you want to change any values or inputs will change all information. so specified input is a must.

* **Performance Testing:**

Performance testing is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted.

This project reduces attendance tables and codes. it will generate reports fast.no have extra time or waiting for results. Entered correct data will show results in a few milliseconds. just used only the low memory of our system. Automatically do not get access to another software. Get user permission and access to other applications.

***Test cases***

A test case is an object for execution for other modules in the architecture and does not represent any interaction by itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases: -manual and automated. A manual test case is executed manually while an automated test case is executed using automation.

In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same.

Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if functional test cases are not yet written, it is ok for testers to check for error conditions when performing normal functional test cases. It should be clear which test data if any is expected to trigger errors.

**TEST CASE:**

* **Test cases for a Login form:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test\_ID** | **Test Condition** | **Expected Result** | **Remark** |
| T001 | Valid User ID and password are entered | The main page is displayed | Pass |
| T002 | User ID or password is invalid | Invalid message box is displayed | Pass |
| T003 | If the user ID and password are not entered | Enter log-in details, a message box is displayed | pass |

* **Report form:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Condition** | **Expected Result** | **Remark** |
| T004 | Attendance report | Excel file of the attendance | pass |
| T005 | Leave form | Valid Form | pass |
| T006 | Leave form | Invalid Form | fail |