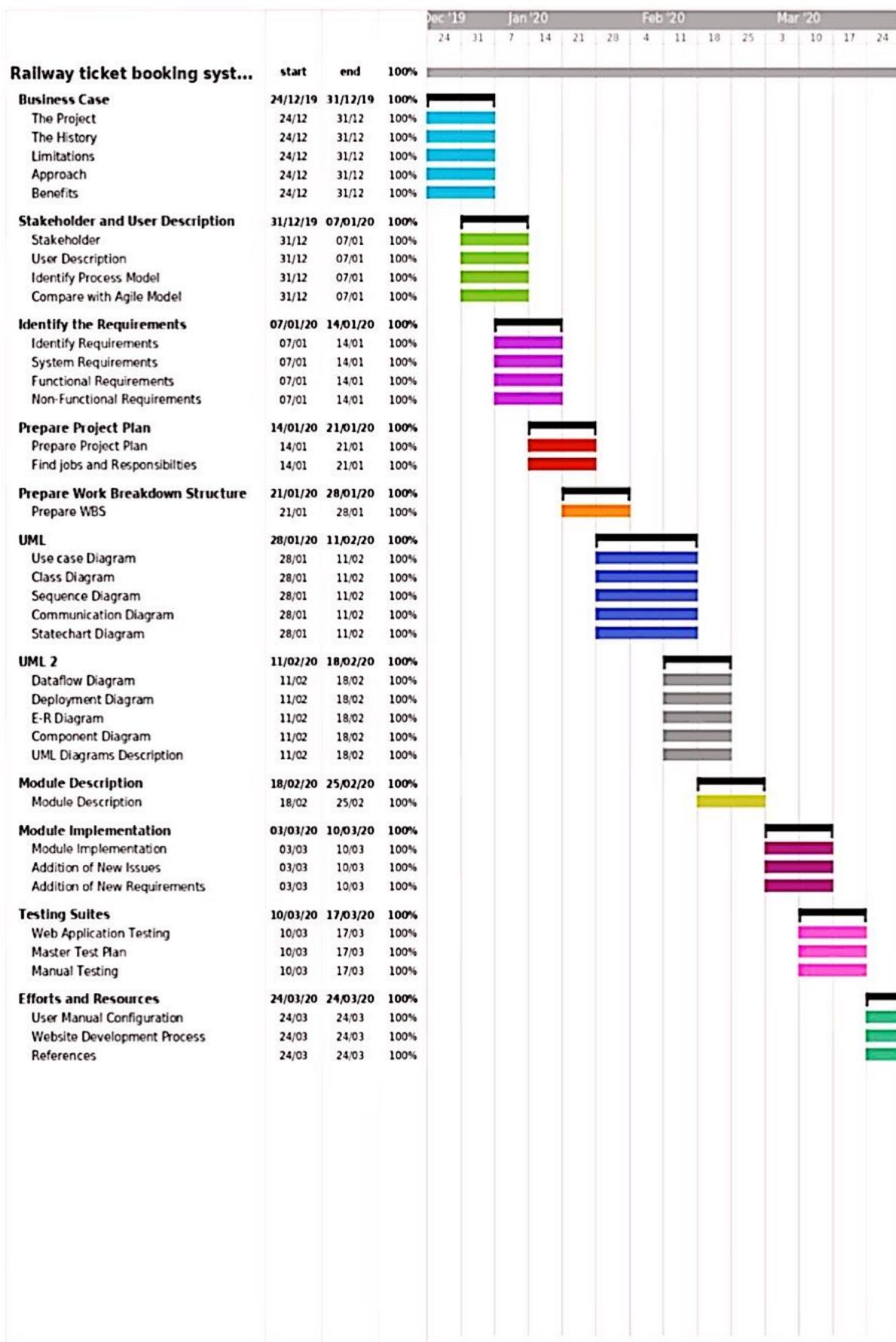


TABLE OF CONTENT

S.NO	Date	Content	Page No.	Sign.
1.		Business Case : 1.1.1- The Project 1.1.2- The History 1.1.3- Limitation 1.1.4- Approach 1.1.5- Benefits		
2.		Stakeholder and Process Model : 2.1.1- Stakeholder 2.1.2- User Description 2.1.3- Identify Process Model 2.1.4- Compare with Agile Model		
3.		Identify the requirements : 3.1.1- Server-End Requirement 3.1.2- System Requirements 3.1.3- Non-Functional Requirements 3.1.4- Functional Requirements		
4.		Project Plan : 4.1.1- Prepare Project Plan 4.1.2- Identify Jobs & Responsibilities		
5.		Project Effort Based on Resources : 5.1.1- Prepare Work Breakdown structure 5.1.2- Risk Management		
6.		Modelling UML Use case Diagram & Capturing USE case Scenarios : 6.1.1- Use Case Diagram 6.1.2- Identifying Actors & Use Cases 6.1.3- Use Case description		
7.		Modelling Class Diagram & functioning of classes : 7.1.1- Class Diagram 7.1.2- Class Diagram Description 7.1.3- Identifying Classes and relationship		

S.No	Date	Content	Page No.	Sign.
2 of 56		Modelling Structural UML Diagrams : 8.1.1- Sequence Diagram 8.1.2- Communication Diagram 8.1.3- State chart diagram		
9.		Modelling Dataflow Diagram : 9.1.1- Dataflow Diagram 9.1.2- Deployment diagram Description		
10.		Modelling Deployment Diagram : 10.1.1- Deployment Diagram 10.1.2- Nodes in Diagram 10.1.3- Diagram Description		
11.		Modelling Entity—Relationship Diagram : 11.1.1- Entity-Relationship Diagram 11.1.2- Diagram Description		
12.		Modelling Component Diagram : 12.1.1- Component Diagram 12.1.2- Component Diagram Description		
13.		Module Description : 13.1.1- Defining And description of modules		
14.		Implementation : 14.1.1- Module Implementation 14.1.2- Addition Of new Issues 14.1.3- Addition of New Requirements		
15.		Designing Test Suites : 15.1.1- Web Application Testing 15.1.2- Master Test Plan Diagram 15.1.3- Master Test Plan Table		
17.		Testing : 17.1.1- Manual Testing & sets of Results		
18.		User Manual & configuration : 18.1.1- User Manual And Configurations Doc.		
19.		Efforts And Resources : 19.1.1- Website Development Process 19.1.2- References		



1.1 PROBLEM STATEMENTS:

One page business case template:-

ONE PAGE BUSINESS CASE TEMPLATE



DATE 9th January 2020
SUBMITTED BY AMANPREET SINGH SETIA
TITLE / ROLE TEAM LEADER

THE PROJECT
In bullet points, describe the problem this project aims to solve or the opportunity it aims to develop.

- CLOUD-BASED ATTENDANCE SYSTEM.
- THIS PROJECT AIMS TO RECORD THE ATTENDANCE, IN-TIME AND OUT-TIME OF AN EMPLOYEE AND TO STORE IN AZURE CLOUD.
- THIS INFORMATION CAN BE ACCESSED ANYTIME BY THE ADMIN.
- TOTAL WORKING HOURS OF EMPLOYEE CAN BE CALCULATED BY IN-TIME AND OUT-TIME.
- THE ADMIN OF THE SYSTEM CAN ADD NEW EMPLOYEE BY FILING A REGISTRATION FORM.

THE HISTORY
In bullet points, describe the current situation.

- IF WE TALK ABOUT CURRENT SCENARIO OF OUR EDUCATION SYSTEM THEN WE FOUND THAT WE HAVE A LOT OF TECHNOLOGIES TO USE BUT STILL WE ARE FOLLOWING A TRADITIONAL SYSTEM IN WHICH LECTURERS TAKE ATTENDANCE MANUALLY HOWEVER BY USING TECHNOLOGIES SUCH AS RFID AND IOT WE CAN BUILD A MUCH BETTER SYSTEM IN WHICH THE ATTENDANCE CAN BE ENTERED AUTOMATICALLY WHICH WILL REDUCE BURDEN ON LECTURERS.

LIMITATIONS
List what could prevent the success of the project, such as the need for expensive equipment, bad weather, lack of specialist training, etc.

- AN ACTIVE INTERNET CONNECTION IS REQUIRED OR ELSE THE ATTENDANCE WON'T BE RECORDED.
- IT'S A REQUIREMENT THAT THE COMPUTER HAS TO MEET THE FOLLOWING HARDWARE REQUIREMENTS ARE LISTED BELOW.
 - PROCESSOR: CORE-i3
 - HARD DISK: 160 GB
 - MEMORY: 2GB
 - VIRTUAL RAM: 4GB OR ABOVE
 - SQL SERVER 2008
 - VISUAL STUDIO 2010

THE PROBLEM STATEMENT:

Introduction:- In this century, where people are always in a rush, find it very difficult to find time for doing essential jobs like taking attendance .But it's not their fault as they have a lot of other work to do. That is why we have leave such kind of things to professionals. Here we are able to take attendance which is cloud based and teachers can take attendance on their smartphones!

APPROACH

list what is needed to complete the project.

- THIS PROJECT IS COMPLETELY BASED ON CLOUD COMPUTING.
- WE WILL USE A COMPUTER TO RECORD THE ATTENDANCE OF AN EMPLOYEE.
- ATTENDANCE WILL BE STORED IN AZURE CLOUD SERVER.
- IT IS COMPLETELY INTERNET CONNECTION BASED.

BENEFITS

In bullet points, list the benefits that this project will bring to the organization.

- THE DATABASE IS SECURED AS IT IS STORED IN AZURECLOUD SERVER.
- EMPLOYEE WORKING HOURS CAN BE CALCULATED IN A SYSTEMATIC MANNER.
- SINCE EMPLOYEE WORKING HOURS ARE CALCULATED SALARY AND OVERTIME CAN BE COMPUTED VERY EARLY.

DISCLAIMER

Any articles, terminals, or information you may find on the website are for reference only. While we strive to keep the information up to date and correct, we make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability, or availability with respect to the website or the information, articles, templates, or related graphics contained on the website. Any reliance you place on such information is therefore strictly at your own risk.

Proposal :- The above problem can be overcome by us because we have a brand new solution for this. Here instead of having a "smartphonic attendance", we will take biological attendance through a simple fingerprint scanner.

2.1 STAKEHOLDER AND USER DESCRIPTION

2.1.1 STAKEHOLDERS:

Project Name: CLOUD BASED ATTENDANCE SYSTEM				
Prepared By: RA1811033010084 RA1811033010066 RA1811033010087 RA1811033010090				
Date: 09/01/2020				
Project Stakeholder Name	Specific Information needs	Project Interests	Impact on Project	Role
Developers	Conference / once in a week	Bug fixes and development	Positive	Decision maker
Students/ Employees	Daily	End users	Positive	Recipient
Admin	Weekly once	Verification of attendance	Positive	Consultant
College	As per requirement	Sponsorship	Positive	Collaborator

2.1.2 USER STORY

USER STORY TEMPLATE

PIO	USER STORY	SUCCESS CRITERIA	OWNER	POINT AND STATUS
1	As a director of SRMIST I want software application on cloud based attendance system so that attendance can be updated .	It can be updated regularly and easily.	SRMIST	 devices with enabled finger print sensors are required.

2.1.3

2.1.4 COMPARISION BETWEEN WATERFALL AND AGILE MODEL

WHY AGILE MODEL IS BETTER THAN WATERFALL MODEL??????

- The Agile Model is based on iterative development and hence it divides the entire project into smaller parts which reduces the risk factor which is not the case in waterfall model.
- The Waterfall model cannot accept the changes in requirements but in agile model it's easy to change the system requirements.
- In agile model, the entire project is divided into smaller parts which helps to minimize the project risk and to reduce the overall project delivery time requirements.
- In waterfall model since risk factor is high, it is not suitable for complex projects.
- In waterfall model the testing is done in later stage it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare, which is not the case in agile model.
- In waterfall model, it follows a sequential approach whereas in agile model it explains the process in order of incremental approach.
- In agile it performs the testing concurrently with software development whereas in waterfall model the testing comes after the build phase only.
- In agile model the distance between the customer and developer is in short whereas in waterfall model it is long.
- In agile there can be done any change in the project but in waterfall model there is no changes throughout the project work.

3.1 IDENTIFYING THE REQUIREMENTS FROM THE PROJECT STATEMENT

3.1.1 REQUIREMENTS:

Requirements are defined during the early stages of the system development as a specification of what should be implemented. A collection of requirements is a requirements document. They may be user level facility description, detailed specification of system behavior, general system property, a specific constraint on the system or information on how to carry on computation. The three types of requirements are explained below.

3.1.2 SYSTEM REQUIREMENTS:

- Web ordering system
- Menu Management System
- Order Retrieval System
- Location tracking system(GPS)
- Rideshare Requesting system
- Client feedback system

3.1.3 FUNCTIONAL REQUIREMENTS:

- Create an account
- Managing the account
- Login to the system
- Navigate through Restaurant Menu
- Select the item from Menu
- Customize the options for the selected Menu
- Add options to their current order
- Review their current order
- Remove all the items from the current order
- Provide delivery and payment details
- Place an order
- Receive confirmation in the form of order number

3.1.3 NON FUNCTIONAL REQUIREMENTS:

- Portability
- Readability
- Availability
- Maintainability
- Security
- User Friendly
- Performance
- Efficiency
- Safety
- Privacy

4.1>Cost Analysis

Estimating Lines of Code needed for the complete development of the product.

Around 1370 Lines of code will be needed for the complete development of the product

I

a. Using COCOMO model for performing cost estimation for the product.

A top-down model can depend on many different factors, instead of depending only on one variable, giving rise to multivariable mode. One approach for building multi-variable models is to start with an initial estimate determined by using the static single-variable model equations, which depend on size, and then adjusting the estimates based on other variables. This approach implies that size is the primary factor for cost other factors have a lesser effect. Here we will discuss one such model called the CONSTRUCTIVE COST MODEL. This model also estimates of total effort in terms of person-months. basic steps in model are:

Obtain an initial estimate of the development effort from the estimate of thousands of delivered lines of source code (KLOC).

Billing System uses around 1365 lines of code. The lines of code are divided into various modules according to their functionality

- Header files - 10 LOC
- Class used in project - 50 LOC
- Global declarations of stream object - 15 LOC
- Function to write in file- 15 LOC
- Function to read all record - 30 LOC

4.1>Cost Analysis

Estimating Lines of Code needed for the complete development of the product.

Around 1370 Lines of code will be needed for the complete development of the product

a. Using COCOMO model for performing cost estimation for the product.

A top-down model can depend on many different factors, instead of depending only on one variable, giving rise to multivariable mode. One approach for building multi-variable models is to start with an initial estimate determined by using the static single-variable model.

This approach implies that size is the primary factor for cost other factors have a lesser effect. Here we will discuss one such model called the CONSTRUCTIVE COST MODEL. This model also estimates of total effort in terms of person-months. Basic steps in model are:

Obtain an initial estimate of the development effort from the estimate of thousands of delivered lines of source code (KLOC).

Billing System uses around 1365 lines of code. The lines of code are divided into various modules according to their functionality

- Header files - 10 LOC
- Class used in project - 50 LOC
- Global declarations of stream object - 15 LOC
- Function to write in file - 15 LOC
- Function to read all record - 30 LOC
- Function to read specific file - 30 LOC
- Function to modify record - 35 LOC
- Function to delete record - 35 LOC
- Function to display all products - 35 LOC
- Phase order & generating bill - 50 LOC
- Voice phase - 500 LOC
- Phase stock exchange - 300 LOC
- Phase history - 200 LOC
- Introduction function - 30 LOC
- Administrator MENU - 40 LOC
- Main function of program - 40 LOC

The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single-variable model, using KLOC as the measure of size. To determine the initial effort in person-months the equation used is of the type $E_i = a \cdot b^{(KLOC)}$

a and b depend on the project type. In COCOMO, projects are categorized into three types - organic, semidetached, and embedded. These categories roughly characterize the complexity of the project with organic projects being those are relatively straight forward and developed by small teams and embedded are those that are ambitious and novel with stringent constraints from the environment and high requirements for such aspects as interfacing and reliability. The constants a and b for different systems is:

The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single-variable model, using KLOC as the measure of size. To determine the initial effort E_i in person-months the equation used is of the type $E_i = a \cdot (KLOC)^b$. The value of the constants a and b depend on the project type. In COCOMO, projects are categorized into three types—organic, semidetached, and embedded. These categories roughly characterize the complexity of the project with organic projects being those are relatively straightforward and developed by small teams and embedded are those that are ambitious and novel with stringent constraints from the environment and high requirements for such aspects as interfacing and reliability. The constants a and b for different systems is:

	a	b	c	d
• Organic	3.2	1.05	2.5	0.38
• Semidetached	3.0	1.12	2.5	0.35
• Embedded	2.8	1.20	2.5	0.32

We are considering our project as an Organic Model. Hence, $a = 3.2$ and $b = 1.05$.

$$\text{Initial Effort} = a \cdot (KLOC)^b, \text{development time} = c \cdot (\text{effort})^d$$
$$= 3.2 \cdot (1.3)^{1.05}$$
$$= 2.5 \cdot (4.7)^{0.38}$$
$$= 4.7 \text{ PM}$$
$$= 4.5 \text{ months}$$

The value of the constants for a cost model depends on the process and has to be determined from past data. COCOMO has instead provided "global constant values". These values should be considered as values to start with until data for some projects is regression analysis. In COCOMO, effort for a phase is a defined percentage of the overall effort. The percentages for an organic software project are given in Table. Using this table, the estimate of the effort required for each phase can be determined from the total effort estimate.

TOTAL EFFORT ESTIMATE = $1.115 \cdot 4.7 = 5.24 \text{ PM}$

(ASSUMING average estimate 1.115)

DEVELOPMENT TIME = $2.5 \cdot (5.24)^{0.38} = 4.6 \text{ months}$

I

JOB ROLES AND RESPONSIBILITIES:

1. PROJECT SPONSOR:

Project sponsor is the one who provides the financial support for the whole project development and execution. They also monitor the process and clarify scope questions. They also provide expert judgement and dictate milestones, key events, or the project end date.

2. SUBJECT MATTER EXPERTS (SME):

SME is a person who has a special skill or knowledge on particular job or topic. They have a deep understanding of particular process, function or machine and hence they help to solve the technical challenges faced by the project.

3. PRODUCT OWNER:

3. PRODUCT OWNER:

A Product Owner in the Scrum Framework is the single person who is responsible for the success of a Product and for maximizing the value of that Product. In the Scrum Framework, a few of the Product Owners' responsibilities are described, such as Product Backlog management, maximizing value and stakeholder management.

4. PROJECT MANAGER (PM):

A project manager is a professional in the field of project management. Project managers have the responsibility of the planning, procurement and execution of a project, in any undertaking that has a defined scope, defined start and a defined finish; regardless of industry.

5. TECHNICAL LEAD:

This person is responsible for overall planning, execution and success of overall complex software solutions to meet customers needs. They have to implement best practice and coding standards to the project. They have to manage the technical scope of the project during and after the delivery.

6. SOFTWARE DEVELOPERS:

Individual who builds and creates an application is called software developers. They write, debug and then execute the source code of the software application. They are also called as programmers since they are the ones that create the programs or the source code.

7. SOFTWARE TESTERS:

Their main role is to check if the actual result match the expected result and to ensure the software system is defect free. Software testing helps to identify the errors or the missing requirements due to which it becomes a vital role. It can be either manually or by using automated tools.

8. USER ACCEPTANCE TESTERS:

This testing is performed by end user or client to accept the software system before moving the software application to the moving environment. This is usually the last process involved in the project management.

9. WEB DEVELOPER:

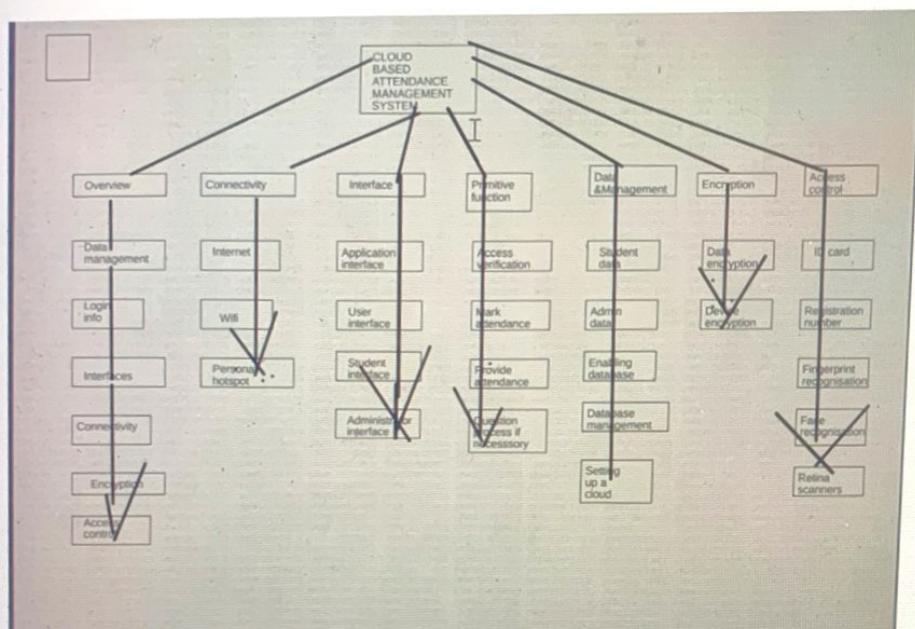
A web developer is a programmer who specializes in, or is specifically engaged in, the development of World Wide Web applications using a client-server model. The applications typically use HTML, CSS and JavaScript in the client, PHP, ASP.NET (C#) or Java in the server, and http for communications between client and server. A web content management system is often used to develop and maintain web applications.

5.1 PROJECT EFFORT BASED ON RESOURCE

5.1 PROJECT EFFORT BASED ON RESOURCE

5.1.1 WORK BREAKDOWN STRUCTURE:

The work breakdown structure of the project is given below:



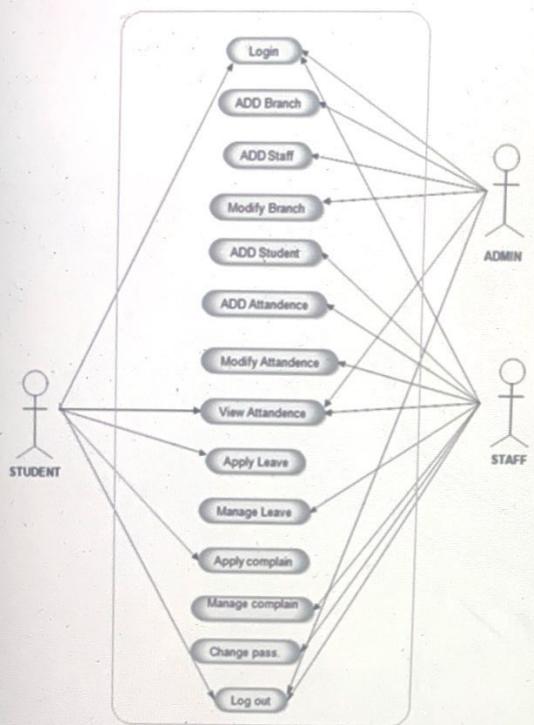
5.1.2 RISK ANALYSIS:

1. Data issues: sometimes there may be high risk of storing data or data exposed to some people
2. High proxy: network traffic must be monitored consistently.

6.1 MODELING USECASE DIAGRAM AND SCENARIOS

6.1.1 Use Case Diagram

Use case Diagram



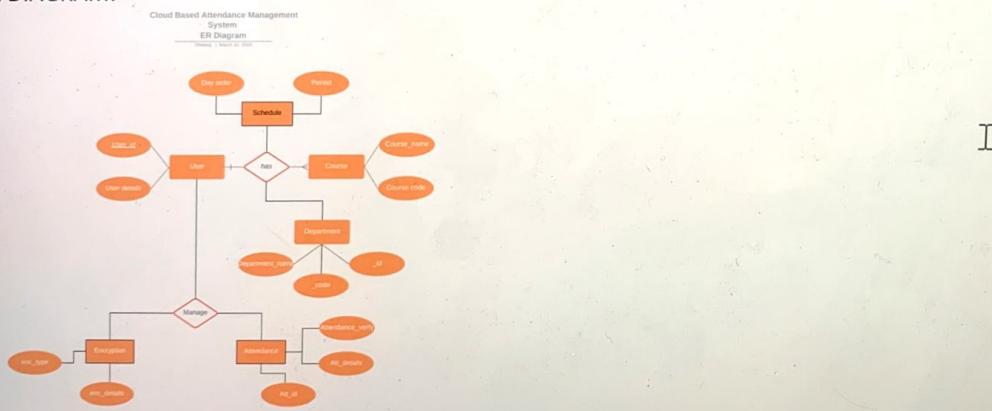
6.1.1 USE CASE DIAGRAM:

7.1ER MODELING FROM THE PROBLEM STATEMENT

7.1.1ER DIAGRAM DESCRIPTION:

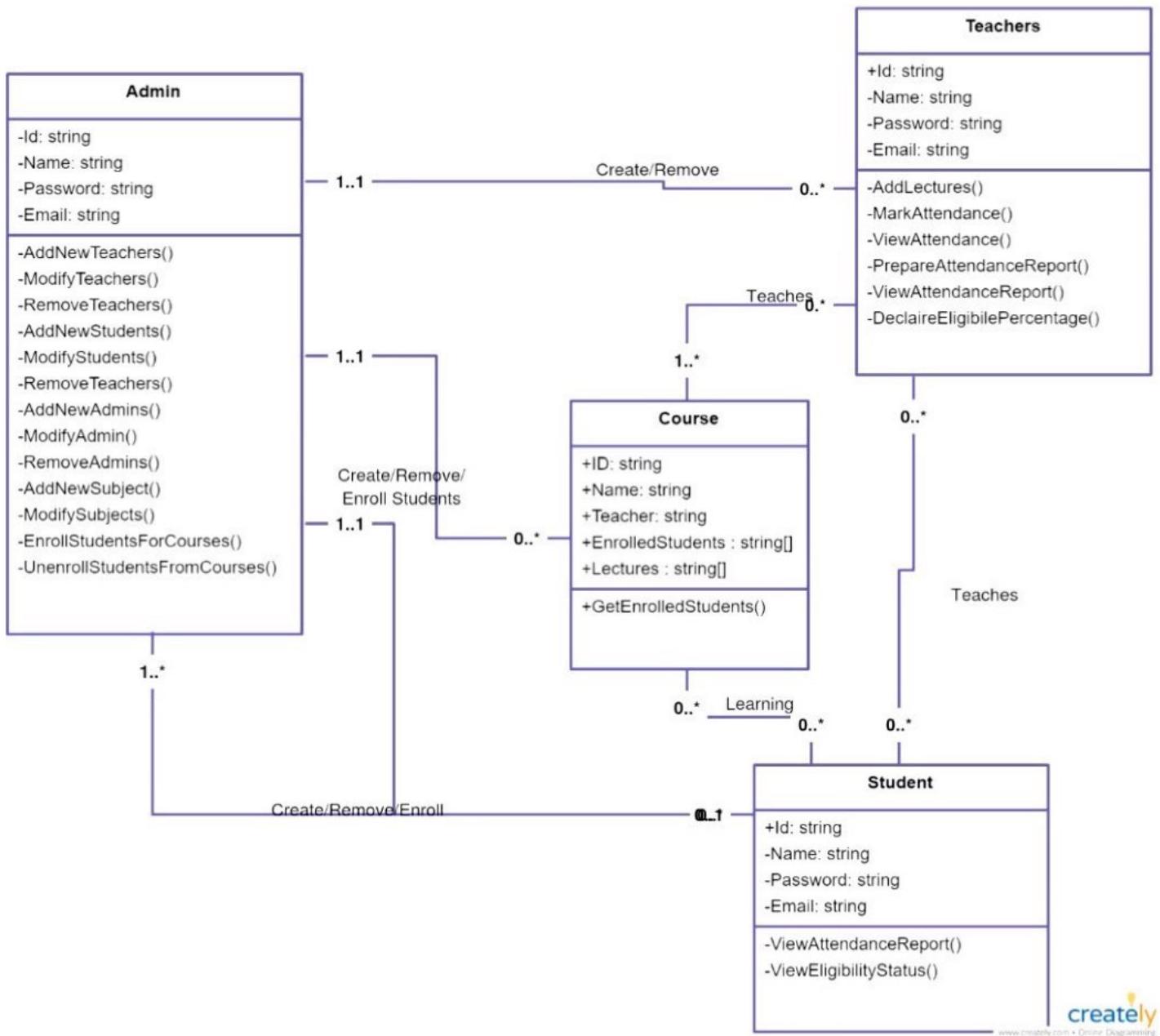
- An entity relationship diagram is a data modelling technique that graphically an information system's entities and relationships between those entities.
- The details of the product are stored in the product tables. Each entity has a primary key and unique keys.
- The entity set presented here are customer,order,products,employee and vehicle.
- All the entities like product, customer, confirmorder, product quality are normalized and reduce duplicacy of records.
- There is one to one and one to many relationships are available between customer,product quality and product.

7.ER DIAGRAM:



10.1 Domain Classes from the given problem:

10.1.1 Domain Class Diagram:



10.1.2 Domain Class Description:

A **domain model** is a *visual* representation of conceptual classes or real - situation objects in a domain [M095, Fowler96]. Domain models have also been called **conceptual models** (the term used in the first edition of this book), **domain object models**, and **analysis object models**.

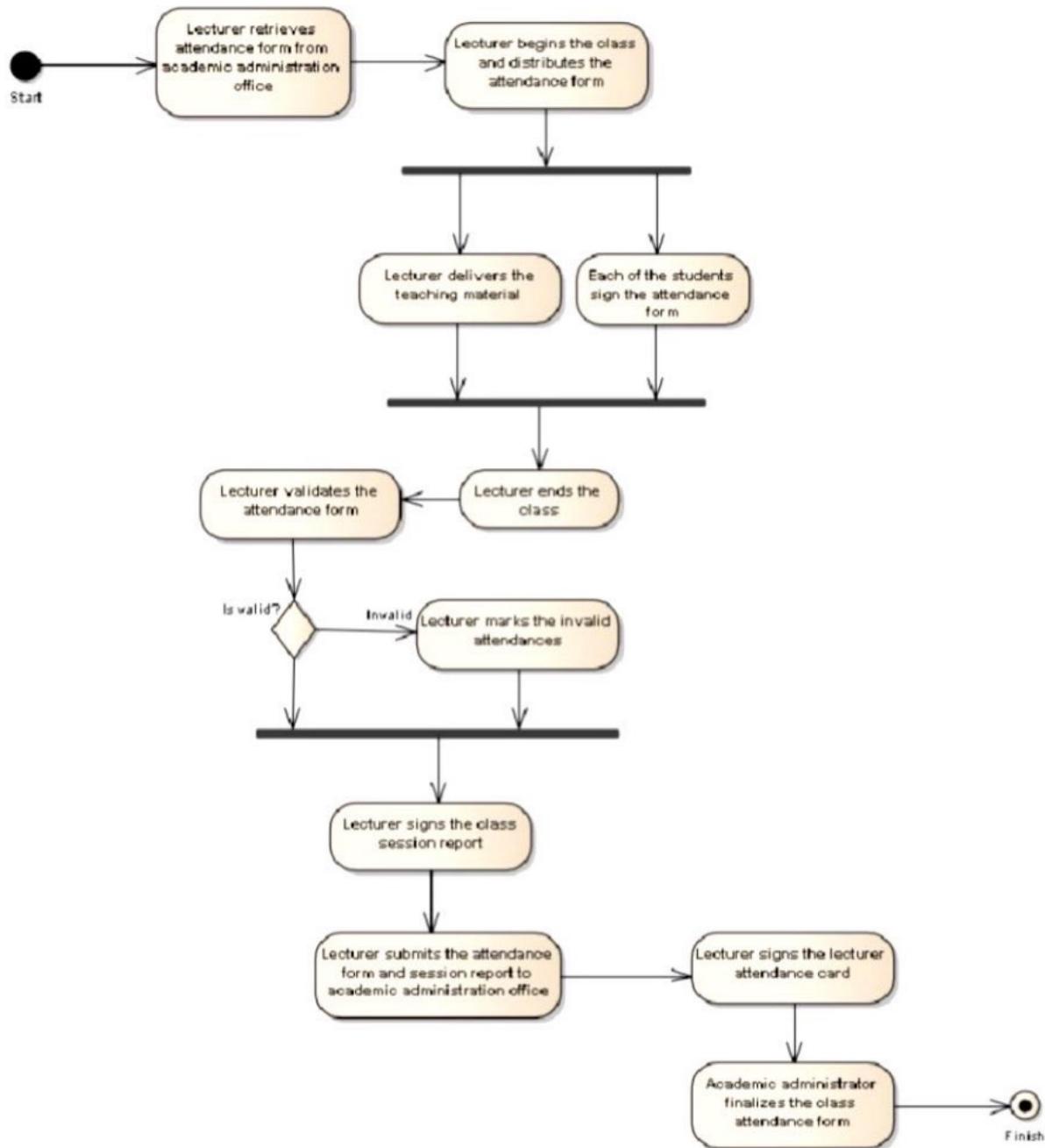
The UP defines the Domain Model as one of the artifacts that may be created in the Business Modeling discipline. More precisely, the UP Domain Model is a specialization of the UP **Business Object Model** (BOM) "focusing on explaining 'things' and products important to a business domain" [RUP]. That is, a Domain Model focuses on one domain, such as POS related things. The more broad BOM, not covered in this introductory text and not something I encourage creating (because it can lead to too much up - front modeling), is an expanded, often very large and difficult to create, multi - domain model that covers the *entire* business and all its sub - domains.

Applying UML notation, a domain model is illustrated with a set of class diagrams in which no operations (method signatures) are defined. It provides a *conceptual perspective*. It may show:

- domain objects or conceptual classes
- associations between conceptual classes
- attributes of conceptual classes

11.1 State Chart UML Modeling:

11.1.1 State Chart Diagram:



11.1.2 State Chart description:

Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination.

Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using Statechart diagrams –

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

11.1.3 Symbols used in a State Chart and Activity diagram:

Initial State

The initial state represents the source of all objects:



It is not a normal state, because objects in this state do not yet exist.

State

The state of an object is always determined by its attributes and associations. States in statechart diagrams represent a *set* of those value combinations, in which an object *behaves the same* in response to events:



Therefore, not every modification of an attribute leads to a new state.

Transition

A transition represents the change from one state to another:



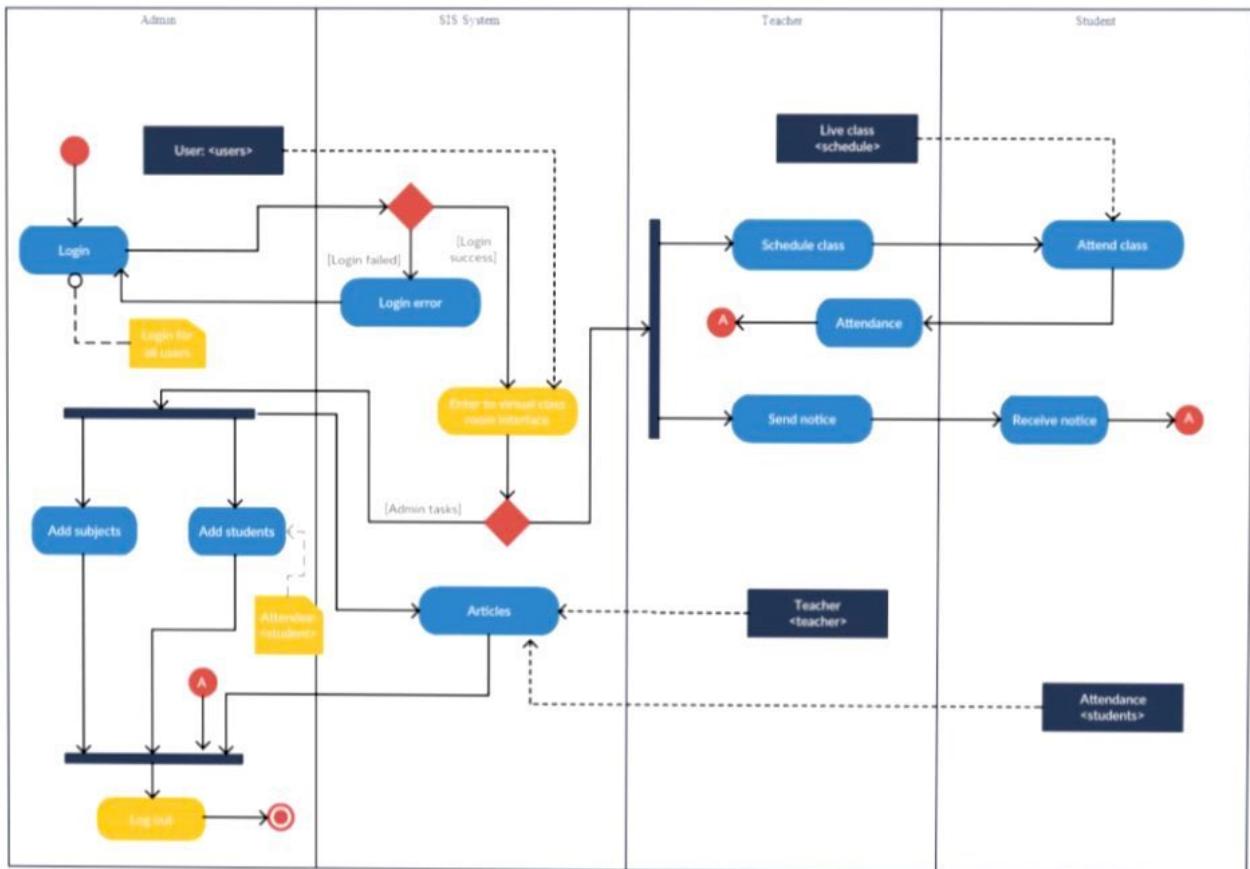
Final State

The final state represents the end of an object's existence:



A final state is not a real state, because objects in this state do not exist anymore.

11.2.1 Activity UML Modeling diagram:



11.2.2 Activity Diagram description:

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

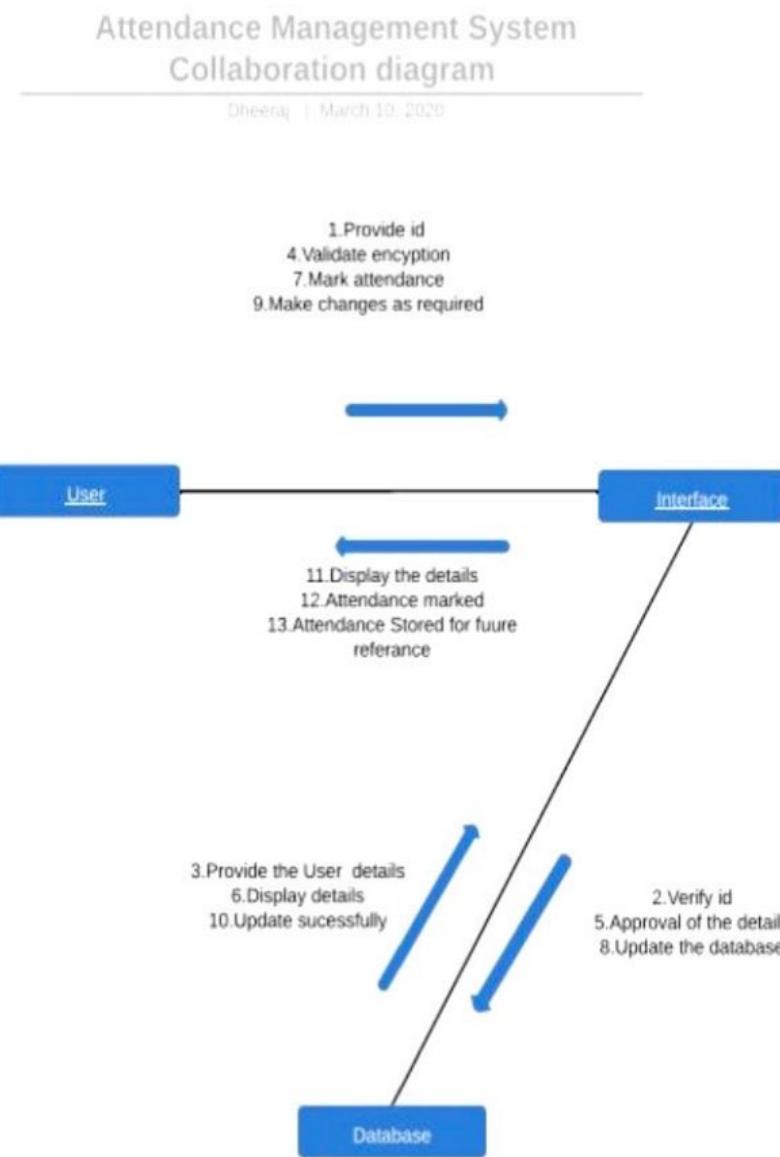
It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as –

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

12.1 Collaboration UML Diagram:

12.1.1 Diagram:



12.1.2 Collaboration Diagram Description:

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the

Unified Modeling Language ([UML](#)). These diagrams can be used to portray the dynamic behavior of a particular [use case](#) and define the role of each object.

Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams.

12.1.3 Notations of a collaboration diagram

A collaboration diagram resembles a [flowchart](#) that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in [real time](#). The four major components of a collaboration diagram are:

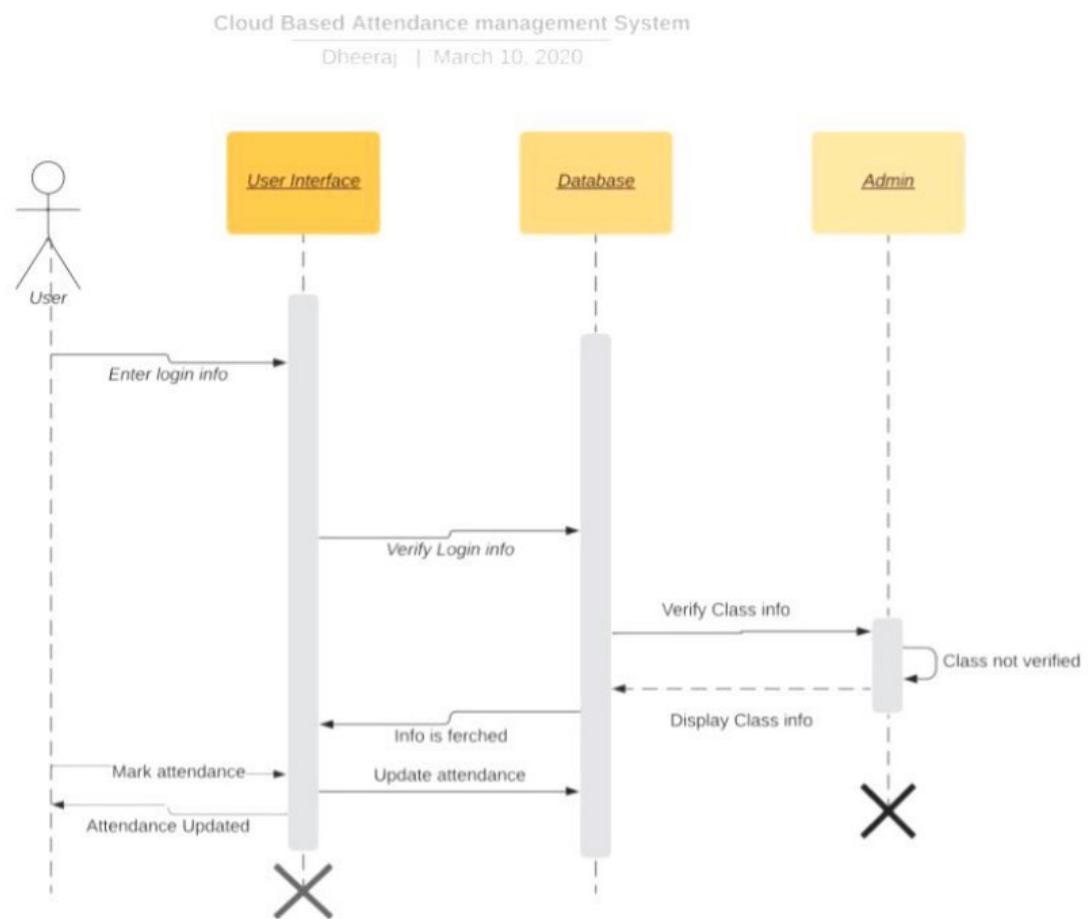
1. Objects- Objects are shown as rectangles with naming labels inside. The naming label follows the convention of object name: class name. If an object has a property or state that specifically influences the collaboration, this should also be noted.
2. Actors- Actors are instances that invoke the interaction in the diagram. Each actor has a name and a role, with one actor initiating the entire use case.
3. Links- Links connect objects with actors and are depicted using a solid line between two elements. Each link is an instance where messages can be sent.

8 of 18

- messages- Messages between objects are shown as a labeled arrow placed near a link. These messages are communications between objects that convey information about the activity and can include the sequence number.

The most important objects are placed in the center of the diagram, with all other participating objects branching off. After all objects are placed, links and messages should be added in between.

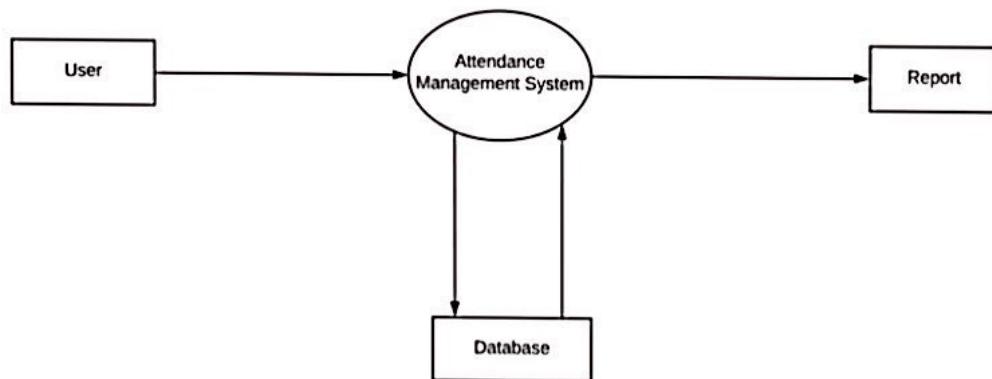
13.1 UML Sequence Diagram:



14.1 Data Flow Modeling:

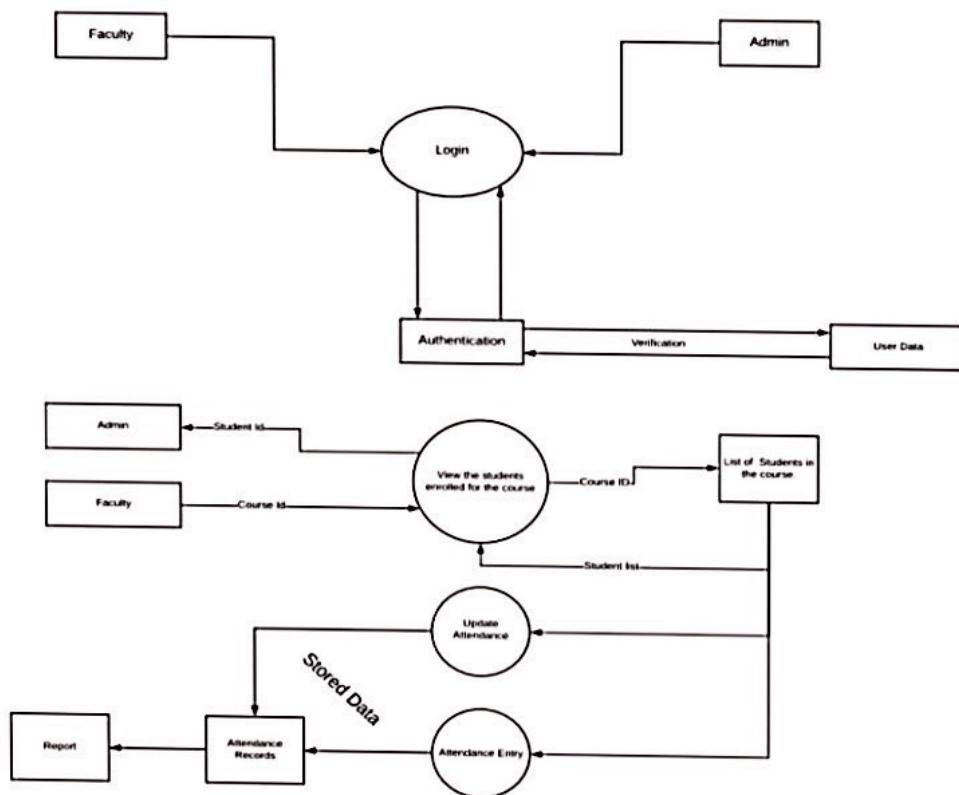
14.1.1 Data Flow Level 0:

DataFlow Diagram
DFD Level : 0

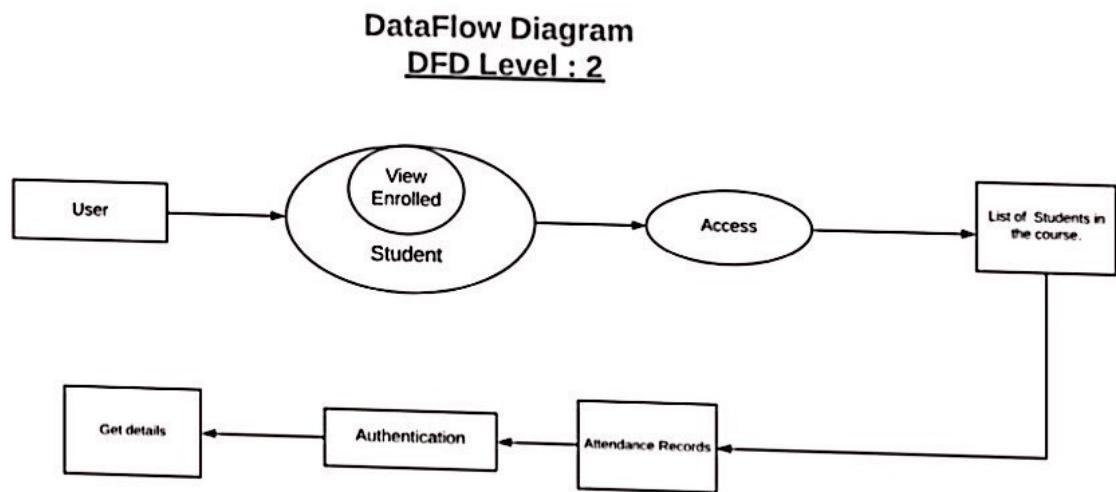


14.1.2 Data Flow Level 1:

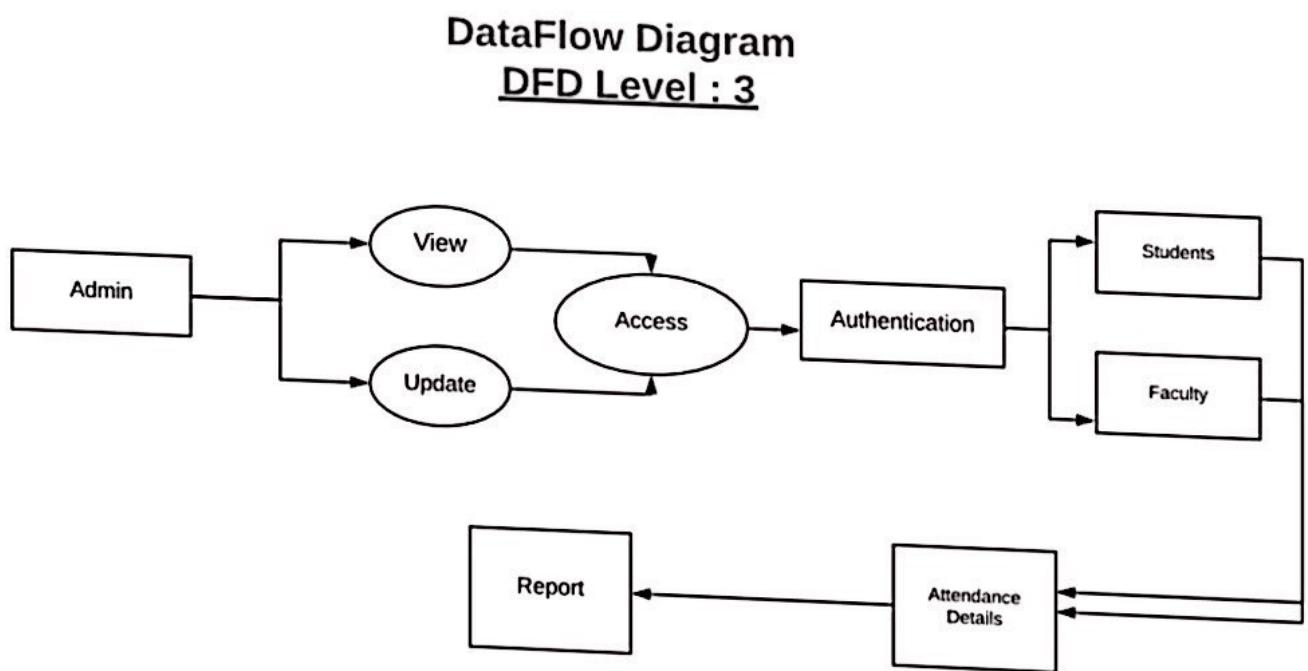
DataFlow Diagram
DFD Level : 1



14.1.3 Data Flow Level 2:



14.1.4 Data Flow Level 3:



15.1 Implementation:

15.1.1 Login page and its HTML code with CSS and JS:



15.1.2 MySQL work bench Student Database view:

Showing how students table data looks like)

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree view shows the 'student' schema selected. Under 'Tables', there are several tables: student, address, attendance, courses, marks, subjects, and student. The 'student' table is currently selected and displayed in the main results grid. The results grid shows the following data:

#	ID	Name	Classmate	Phone	Email	g_name	g_phone
1	1234567890	Al Roker	b_tech 4th year	9876543211	abc@123.com	Latha Devi	9898989898
2	1234567891	deeksha	b_tech 4th year	9876543211	abc@123.com	Father	9898989898
3	1234567892	anand	b_tech 4th year	9876543211	abc@123.com	Mother	9898989898
4	1234567893	anusha	b_tech 3rd year	9876543211	abc@123.com	Sister	9898989898
5	1234567894	anusha	b_tech 3rd year	9876543211	abc@123.com	Brother	9898989898

Below the results grid, the SQL editor shows the query used to retrieve the data: `SELECT * FROM student LIMIT 5, 1000;`

16.1 Software Testing:

Software testing can be stated as the process of verifying and validating that a software or application is bug free, meets the technical requirements as guided by it's design and development and meets the user requirements effectively and efficiently with handling all the exceptional and boundary cases.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy and usability. It mainly aims at measuring specification, functionality and performance of a software program or application.

Software testing can be divided into two steps:

- 1. Verification:** it refers to the set of tasks that ensure that software correctly implements a specific function.

2. Validation: it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Verification: "Are we building the product right?"

Validation: "Are we building the right product?"

What are different types of software testing?

Software Testing can be broadly classified into two types:

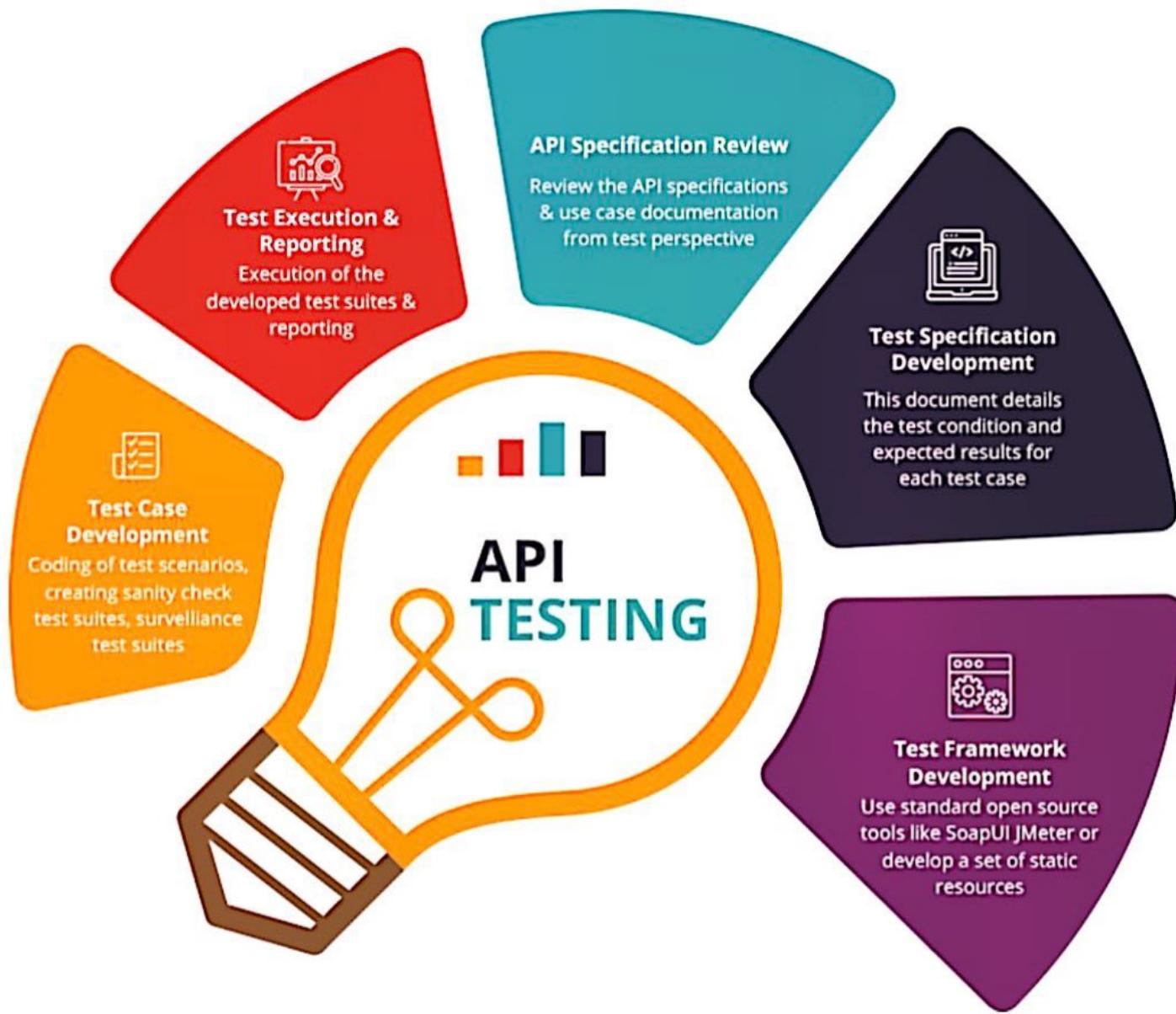
1. Manual Testing: Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

2. Automation Testing: Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.

Apart from regression testing, automation testing is also used to test the application from load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money in comparison to manual testing.

16.1.1 Master Testing Plan:



16.1.2 Manual Testing Table:

S . N o	Test Case Id	Test Case Name	Test Case Description	P	Expected Result	Actual Result	Status
1 .	Login Admin	Validate Login	To Verify the Login of Admin	2	Admin Logged in Successfully	Login Successful	Pass
2 .	Login Staff	Validate Login	To Verify the Login of Staff	2	Admin Logged in Successfully	Login Successful	Pass
3 .	Updating Information	Validation of information to be updated	To Update existing information	1	Info is Updated	Update Successful	Pass
4 .	Encryption	Validate Verification	Verification by a specific encryption method	3	Access granted only if verified.	Access granted for verified personnel	Pass
5 .	Password Retrieval	Retrival of forgotten Password	An E-mail is sent to retrieve forgotten password	4	Forgotten password is retrieved	Password Retrieved.	Pass
6 .	Active Server	Verify Its availability	Word Secure and StableServer	0	Stable connection	Connection Stable	Pass

Sn o	Test case id	Test case name	Test case desc	Step	Expected result	Actual Result	Test case status pass/fail
1	Create student details	Validate allocation form	To allocate separate roll no for the students	Nothing entered and click submit button	An error message student name not equal to null must be displayed	Inserted successful	Pass
2	Create staff details	Validate allocation form	To allocate separate subject user name password for the staffs	Nothing entered and click submit button	An error message staff details password,username not equal to null must be displayed	Inserted successful	Pass
3	Create time table	Validate allocate period form	To verify that data stored on database	Nothing entered and click submit button	An error message not click not allocation subject table not equal to null must be displayed	Inserted successful	Pass
4	View	Check details of all data	To verify that data stored on database	generated	An error message return null will be displayed	An error message return null will be displayed	fail

17.1 Conclusion and What it could Progress into:

Attendance management is significant to all organizations such as educational institutions. It can manage and control the success of any organization by keeping track of people within the organization such as students to maximize their performance. The proposed system offers the process of monitoring attend students, it aims to help the teacher in the classroom or laboratories to manage and record students' presence electronically and directly without the need to list on paper so it will save time and effort. The system can analyze the data and display statistics about the student's absences, printing reports about absence percentages and students' warnings for the specified period. The developed system easy to use and friendly that has an attractive and simple GUI design so that insertions, deletions, and changes of data can do easily without interacting with the tables, so it was designed with the MVC architecture and implemented with the powerful Laravel Framework. In order to increase use of the application and make it easy to use and attractive, JavaScript, jQuery and AJAX have been used. MySQL use for the application Database which stores the data for a long period. The application's test case revealed that the system is working exciting and is ready to use to manage students' attendance for any department of the University, College or Institute. Since our system is modular and can extend effortlessly, the future work ambitions are to make the system take attendance by other methods such as face recognition and using Biometrics (fingerprint) techniques, NFC mobile devices, or RFID Systems. Furthermore, we would like to make the system to manage and record the attendance for the staff of the university.

18.1 References:

- <http://google.com>
- <http://w3school.com>
- <https://www.geeksforgeeks.org>
- <http://www.tutorialspoint.com>

BY

RA1811033010066

RA1811033010087

RA1811033010084