

St. Sebastine's Higher Secondary School Management System

Project Report Submitted by

FEBIN BABU

Reg. No.:AJC21MCA-2057

In Partial fulfillment for the Award of the Degree Of

MASTER OF COMPUTER APPLICATIONS

(MCA TWO YEAR)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2022-2023

DEPARTMENT OF COMPUTER APPLICATIONS
AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, “St. Sebastine’s Higher Secondary School Management System” is the bona fide work of **FEBIN BABU (Regno: AJC21MCA-2057)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2022-23.

Ms. Meera Rose Mathew

Internal Guide

Ms. Meera Rose Mathew

Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report “**St. Sebastine’s Higher Secondary School Management System**” is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2022-2023.

Date:

FEBIN BABU

KANJIRAPPALLY

Reg: AJC21MCA-2057

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for his valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Meera Rose Mathew** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

FEBIN BABU

ABSTRACT

An online website for St. Sebastine's Advanced Secondary academy, which offers smooth working of the academy. It offers a straightforward stoner interface for streamlining pupil's information. It handles everything related to scholars. After log into the system the pupil can have numerous features, similar as apply leave, upload details, view affect etc. This point provides the complete information about this particular institution similar as faculty information, pupil progress, results etc. It provides complete details about the preceptors similar as name, place, education qualification etc. The preceptors have the power to upload the internal test results, attendance, and progress of the scholars. This website also provides an access to the website and can view the progress of the pupil and also have the installation to converse with the particular class schoolteacher. This design offers a dependable and effective system for managing pupil information and affiliated tasks, reducing homemade sweats and crimes. It'll give an easy- to- use interface for both scholars and faculty, easing smooth communication and timely updates. The addition of a Google restate point will feed to the requirements of non-native English speakers, making the website accessible to a wider followership. The website's design and functionalities will be developed keeping in mind the stoner's perspective and ease of use. In summary, this design aims to develop a comprehensive and stoner-friendly website for St. Sebastine's Advanced Secondary academy, feeding to the requirements of scholars, faculty, and parents likewise.

CONTENT

SL. NO	TOPIC	PAGE NO
1	INTRODUCTION	
1.1	PROJECT OVERVIEW	
1.2	PROJECT SPECIFICATION	
2	SYSTEM STUDY	
2.1	INTRODUCTION	
2.2	EXISTING SYSTEM	
2.3	DRAWBACKS OF EXISTING SYSTEM	
2.4	PROPOSED SYSTEM	
2.5	ADVANTAGES OF PROPOSED SYSTEM	
3	REQUIREMENT ANALYSIS	
3.1	FEASIBILITY STUDY	
3.1.1	ECONOMICAL FEASIBILITY	
3.1.2	TECHNICAL FEASIBILITY	
3.1.3	BEHAVIORAL FEASIBILITY	
3.1.4	FEASIBILITY STUDY QUESTIONNAIRE	
3.2	SYSTEM SPECIFICATION	
3.2.1	HARDWARE SPECIFICATION	
3.2.2	SOFTWARE SPECIFICATION	
3.3	SOFTWARE DESCRIPTION	
3.3.1	DJANGO	
3.3.2	MYSQL	
4	SYSTEM DESIGN	
4.1	INTRODUCTION	
4.2	UML DIAGRAM	
4.2.1	USE CASE DIAGRAM	
4.2.2	SEQUENCE DIAGRAM	
4.2.3	STATE CHART DIAGRAM	
4.2.4	ACTIVITY DIAGRAM	
4.2.5	CLASS DIAGRAM	
4.2.6	OBJECT DIAGRAM	
4.2.7	COMPONENT DIAGRAM	

4.2.8	DEPLOYMENT DIAGRAM	
4.2.9	COLLABORATION DIAGRAM	
4.3	USER INTERFACE DESIGN USING FIGMA	
4.4	DATA BASE DESIGN	
5	SYSTEM TESTING	
5.1	INTRODUCTION	
5.2	TEST PLAN	
5.2.1	UNIT TESTING	
5.2.2	INTEGRATION TESTING	
5.2.3	VALIDATION TESTING	
5.2.4	USER ACCEPTANCE TESTING	
5.2.5	AUTOMATION TESTING	
5.2.6	SELENIUM TESTING	
6	IMPLEMENTATION	
6.1	INTRODUCTION	
6.2	IMPLEMENTATION PROCEDURE	
6.2.1	USER TRAINING	
6.2.2	TRAINING ON APPLICATION SOFTWARE	
6.2.3	SYSTEM MAINTENANCE	
7	CONCLUSION & FUTURE SCOPE	
7.1	CONCLUSION	
7.2	FUTURE SCOPE	
8	BIBLIOGRAPHY	
9	APPENDIX	
9.1	SAMPLE CODE	
9.2	SCREEN SHOTS	

List of Abbreviation

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

A simple website for the higher secondary school which provides user interface for updating student's information. It handles everything related to students. The student can log into the system via the provided login credentials. After logging in the students have the access to upload their details after that they can view their profile. The students can view the assigned assignment and can download the question paper and after finishing the assignment they can upload the assignment through this website. The student can apply leave via this website. At last, the student can view the exam marks and can download the progress report. Although the student can rate themselves to know their taste in which career, they are good at.

This website also provides the access to teachers who have the power to upload marks, schedule the assignments, apply leave. The teacher can also predict the number of students who will pass or fail after the internal examinations. The parents can view their student's progress and can chat with the class teacher.

In admin side the admin is uploading the details of the students and teachers for log in. The admin adds the different courses and subjects and also assigning the different teachers to different subjects. The admin is the one who approves the leave applied by the teachers.

1.2 PROJECT SPECIFICATION

The project, simple website for a higher secondary school provides various functionalities such as registering, upload details etc. In the registration form, the student should enter the complete details such as name, address, email, phone, place, mark percentage, caste, course, house name etc. After registering the form will be verified by the admin of the institution and admin admits the students to the institution. After admitting, the student can access to the website which will provide the complete working of the school which includes the class timetable, general instructions, subjects etc. This site provides the complete information about this institution such as faculty information, student progress, results etc.

In this online school management system, there are mainly three users

- Admin
- Teacher
- Student

Admin

Admin has the whole power to manage the system. He has the power to add various courses, add teachers and students.

Teachers

Teachers has the power to add the progress of the students, attendance, upload marks send feedback.

Student

Students have the power to view their progress, details etc. Also, student can view important information.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is the process of acquiring and assaying data, diagnosing issues, and using the data to suggest system changes. The system druggies and system inventors must communicate considerably during this problem- working process. A critical stage of every system is the analysis or study phase. creation procedure. The system is strictly examined and assessed. The System judges act as doubters and claw deeply into the operation of the current frame. The input to the system is seen as a whole and includes linked. The process must be completely studied using a variety of ways, including questionnaires and interviews. These sources' data must be precisely examined to get at a decision.

2.2 EXISTING SYSTEM

Existing system is not an automated system. The online class links are shared on WhatsApp groups and each subject have WhatsApp groups. Notes are shared through this platform it is verydifficult to recover the early notes and difficult to obtain the progress of student in online mode classes. It is necessary to modify the existing system to include additional information and makethe system efficient, flexible, and secure. Using the new system students can easily access those study things. Also, the leave records are kept manual and may have the chance to lose the records.

2.2.1 NATURAL SYSTEM STUDIED

It's veritably hard to manage and record the diurnal working of the being system. The documents are stored in the form of records and papers etc. The possibility of circumstance of error is veritably high. It needed more space to store all the documents and it's veritably hard to pierce a specific data from the documents. The data isn't harmonious because if a data item needs to be changed also all the lines containing the data need to be changed or modified. It may produce a threat of outdated values of data. Poor data security is the most threatening problem in the train processing system, there's only veritably lower security in train processing system.

2.2.2 DESIGNED SYSTEM STUDIED

The system Online School Management system is a web-based application that helps the school, to automate the manual tasks and maintain the records of the school. Academic Enterprise Solutions, an academic web application which is generated for the smooth working of a particular school or educational institutions.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Difficult to access the course materials by Students
- Very time consuming to retrieve the student details.
- Difficult to manage the attendance.
- All data are kept in manually.
- The data is not consistent because if a data item needs to be changed then all the files containing the data need to be changed or modified.

2.4 PROPOSED SYSTEM

The system Online School Management system is a web-based application that helps the school, to automate the manual tasks and maintain the records of the school. It provides options to enter the details of student and teachers, option to apply leave and view the leave status. As compared to existing system studied this system provides more functionalities such as apply leave, send feedback through online via email, upload details view courses, download course materials and question paper.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Better management of student data
- Improves overall teacher productivity.
- Ensure data precision.
- Improved services.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Planning, organizing, and managing resources to ensure the achievement of project goals and objectives is the process of project management. A feasibility study is a preliminary examination of a prospective project or end to determine its merits and viability. A feasibility study aims to provide an objective assessment of the technical, economic, financial, legal, and environmental elements of a proposed project. The information can then be used by decision-makers to decide whether to proceed with the project or not. The findings of the feasibility study can also be used to develop a practical project plan and budget. It cannot be simple to determine whether a proposed project is worthwhile pursuing without a feasibility study. The document provides the feasibility of the project that is being designed and lists. Various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibility. The following are its features: -

3.1.1 Economic Feasibility

Proposed system is economically feasible as the only cost involved in having an average model mobile device and a stable internet connection. Economic feasibility analysis is the most used method for determining the efficiency of a new project. It is also known as cost analysis. It helps in identifying profit against investment expected from a project. Cost and time are the most essential factors involved in this field of study. The economic feasibility study for projects shows all the costs necessary for the project, which are the costs of establishing the project, as well as the size of the investment, estimating the size of the project's profits, and knowing the net profit during a specific period. and tools needed for the project. The proposed system is developed as part of project work, thus there is no manual cost spent for the proposed system. All the required resources were already available, which shows that the system is economically feasible for development. The project was developed at a low cost as it is completely developed using open-source software.

3.1.2 Technical Feasibility

The system needs to be assessed first from a technical standpoint. The outline design of the system requirement in terms of input, output, programs, and procedures must serve as the foundation for the assessment of this feasibility. After determining an outline investigation must continue to identify the necessary equipment kind. The technical feasibility study can be defined as the study related to all the technical aspects of the project.

3.1.3 Behavioral Feasibility

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible. Behavioral feasibility answers the following questions:

- Whether the proposed system will cause any harm to its users?
- Is this proposed system sufficient to support the users?

The project is behaviorally feasible because it satisfies all the objectives when developed and installed.

3.1.4 Feasibility Study Questionnaire

1. Name of the School? Is there any existing website for this School?

The name of the school is St. Sebastine's Higher Secondary School Nedumkandam, there is no existing online website for this school. All the works are done manually.

2. How is the admission process take place in this institution?

The admission process is taken place in offline mode. The students and parents are requested to come and fill the registration form and if the request is accepted a small amount of fees are requested to pay

3. How is the information shared with the students?

The information is shared via the what's app or via SMS.

4. Is there any facility to know about the students' progress?

No, the teachers are requested to prepare some documents to keep track the progress of the students.

5. How are the teachers appointed to this institution?

The teachers are requested to fill a registration form and submit their resume. After that the teacher wants to face an interview session.

6. How are the details of the students stored?

The details of the students are kept in file format.

7. The list of available courses?

- a. Bioscience (Physics, Chemistry, Biology, Mathematics, English)
- b. Home science (Physics, Chemistry, Biology, English)

-
8. List of second languages?
 - a. Malayalam
 - b. Hindi
 9. Number of seats available in each course?
 - a. Bio-science – 60 seats
 - b. Home Science – 60 seats
 - c. Commerce – 60 seats
 - d. Humanities – 60 seats
 10. Does the students and parents are able to Access the online website of the school?
Yes, the students, teachers and parents can access the online
 11. Is there any option to communicate with the teacher...?
Yes, this website provides a chat app feature to communicate with the class teacher
 12. Is there any medium to know that if student take leave?
Yes, this website provides a messaging system that if any student takes leave a mail will be generated to parent's email id.
 13. Which all measures could take to improve the performance of student?
Yes, this website provides a result analysis which predicts the final result percentage of the student in advance, which helps the teachers to schedule extra classes to improve the performance of the student

3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel core i3

RAM - 4 G B

Hard disk - 1 T B

3.2.2 Software Specification

Front End - HTML, CSS

Backend - Django, SQLite3

Client on PC - Windows 7 and above.

Technologies used - JS, HTML, Python, Django, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 DJANGO

Django is an open-source web framework written in the Python programming language. Named after the jazz guitarist Django Reinhardt, it is used by some of the largest websites in the world including Instagram, Mozilla, and NASA, but also lightweight enough to be a popular choice for

weekend side projects and startups. Its "batteries-included" approach means a powerful website can be generated quickly in the hands of a skilled developer Django adopts a "batteries-included" approach like Python and comes with several built-in features including an extensible authentication system, robust admin app, lightweight testing web server, and support for multiple databases including PostgreSQL, MySQL, MariaDB, Oracle, and SQLite. It is known for its leading security best practices and comes with comprehensive documentation, available either online or as a PDF/ePUB for offline consumption. As a mature project, Django rarely makes breaking changes and has a clear deprecation schedule for any updates. A major new version is released every nine months or so with monthly patch releases for security and bug fixes. There is also a vibrant ecosystem of third-party applications--visible on the Django Packages site--which provide additional functionality.

3.1.2 SQLITE3

SQLite is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. It is the most used database engine in the world. It is an in-process library, and its code is publicly available. It is free for use for any purpose, commercial or private. It is basically an embedded SQL database engine. Ordinary disk files can be easily read and write by SQLite because it does not have any separate server like SQL. The SQLite database file format is cross-platform so that anyone can easily copy a database between 32-bit and 64-bit systems. Due to all these features, it is a popular choice as an Application File Format.

Applications of SQLite

Due to its small code print and efficient usage of memory, it is the popular choice for the database engine in cellphones, PDAs, MP3 players, set-top boxes, and other electronic gadgets. It is used as an alternative for open to writing XML, JSON, CSV or some proprietary format into disk files used by the application. As it has no complication for configuration and easily stores file in an ordinary disk file, so it can be used as a database for small to medium sized websites. It is faster and accessible through a wide variety of third-party tools, so it has great application in different software platforms.

CHAPTER 4

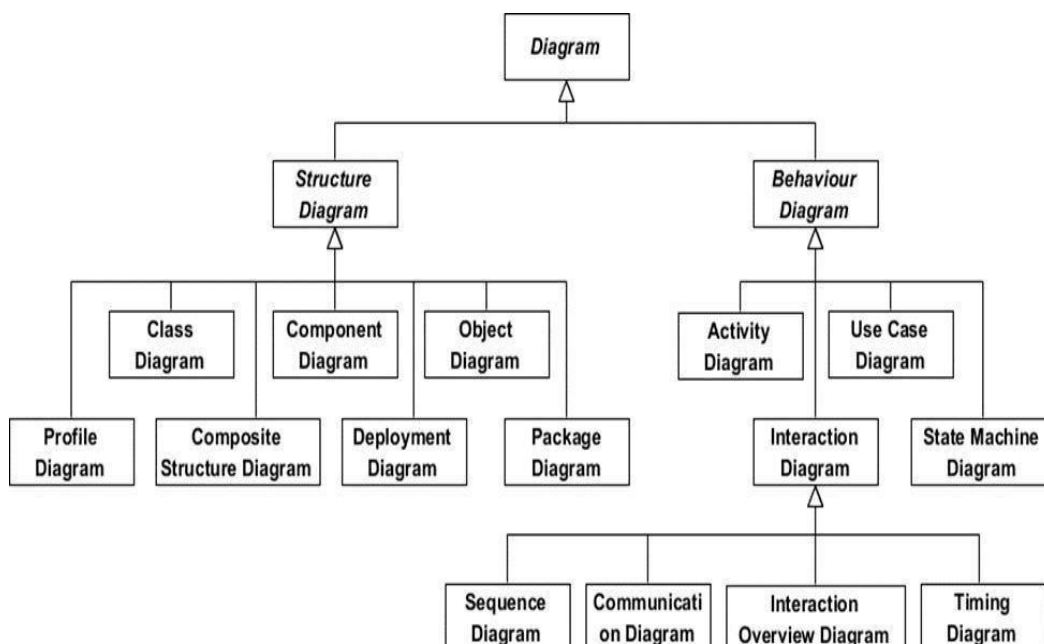
SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step in the development of any engineered system or product. Design is a creative process. A good design is the key to a system that works effectively. "Design" is the process of using many approaches and concepts to thoroughly outline a process or a system so that it can be physically implemented. It can be defined as the process of utilizing several approaches and concepts to precisely specify a component, a procedure, or a system to enable its physical realization. Regardless of the development paradigm employed, software design forms the technical core of the software engineering process. The system design creates the necessary architectural detail.

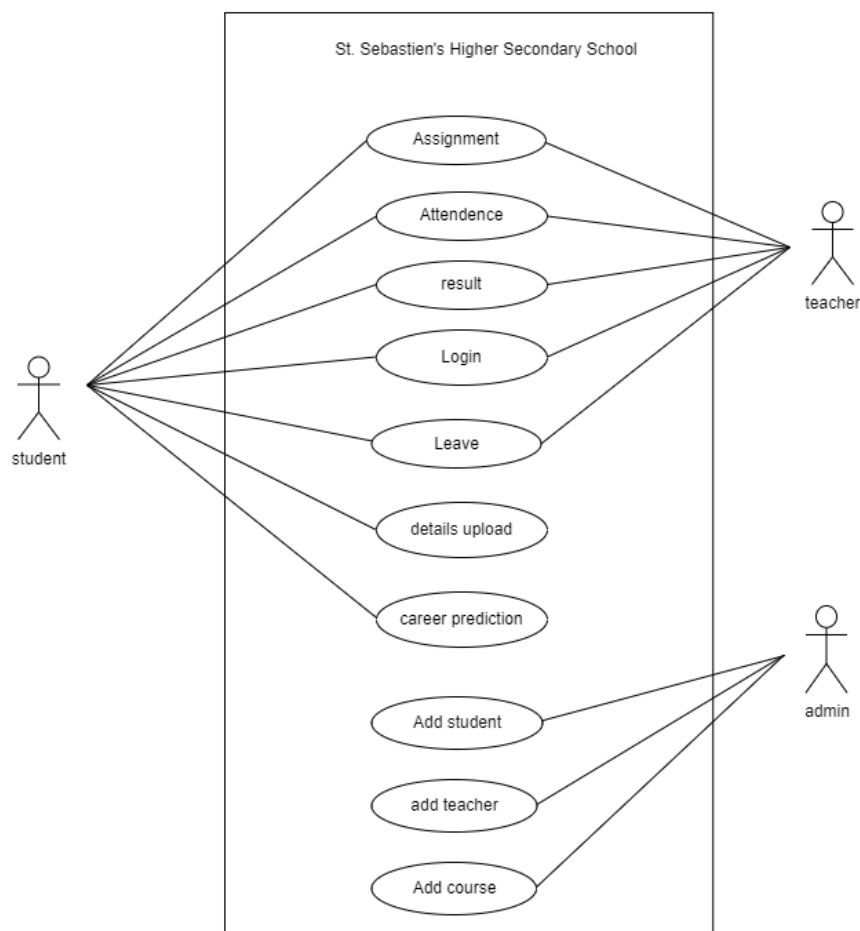
4.2 UML DIAGRAM

The components of the principles of object-oriented programming are represented by the language known as the Unified Modeling Language (UML), which is utilized in the industry of software engineering. It serves as the standard definition of the entire software architecture or structure. Complex algorithms are solved and interacted with in Object-Oriented Programming by treating them as objects or entities. Anything can be one of these things. It could either be a bank manager or the bank itself. The thing can be a machine, an animal, a vehicle, etc. The issue is how we connect with and control them, even though they are capable of and ought to execute duties. Interacting with other objects, sending data from one object to another, manipulating other objects, etc., are examples of tasks. There could be hundreds or even thousands of objects in a single piece of software.



4.2.1 USE CASE DIAGRAM

A use case diagram is a visual representation of the interactions between system components. An approach for identifying, outlining, and organizing system requirements is called a use case. The word "system" in this context refers to a project or business that is under development or operation, such as a mail-order goods sales and service web page. The Unified Modelling Language (UML) makes use of use case diagrams, a common notation for simulating systems and things in the actual world. Planning for overall requirements is one of the system objectives. Testing and debugging a software product, and verifying a hardware design. Performing a consumer service, developing, writing an online help guide, or focused task. Use cases in a product sales context, for instance, would include ordering of goods, catalogue revision, transaction processing, and client.



4.2.1 SEQUENCE DIAGRAM

A sequence diagram essentially shows how things interact with one another sequentially, or the order in which these interactions occur. A sequence diagram can also be referred to as event diagrams or event scenarios. Sequence maps define the actions that the system's components take and in what order. These schematics are Businesspeople and software developers frequently employ documentation and understanding specifications for both current and future systems.

Sequence Diagram Notations –

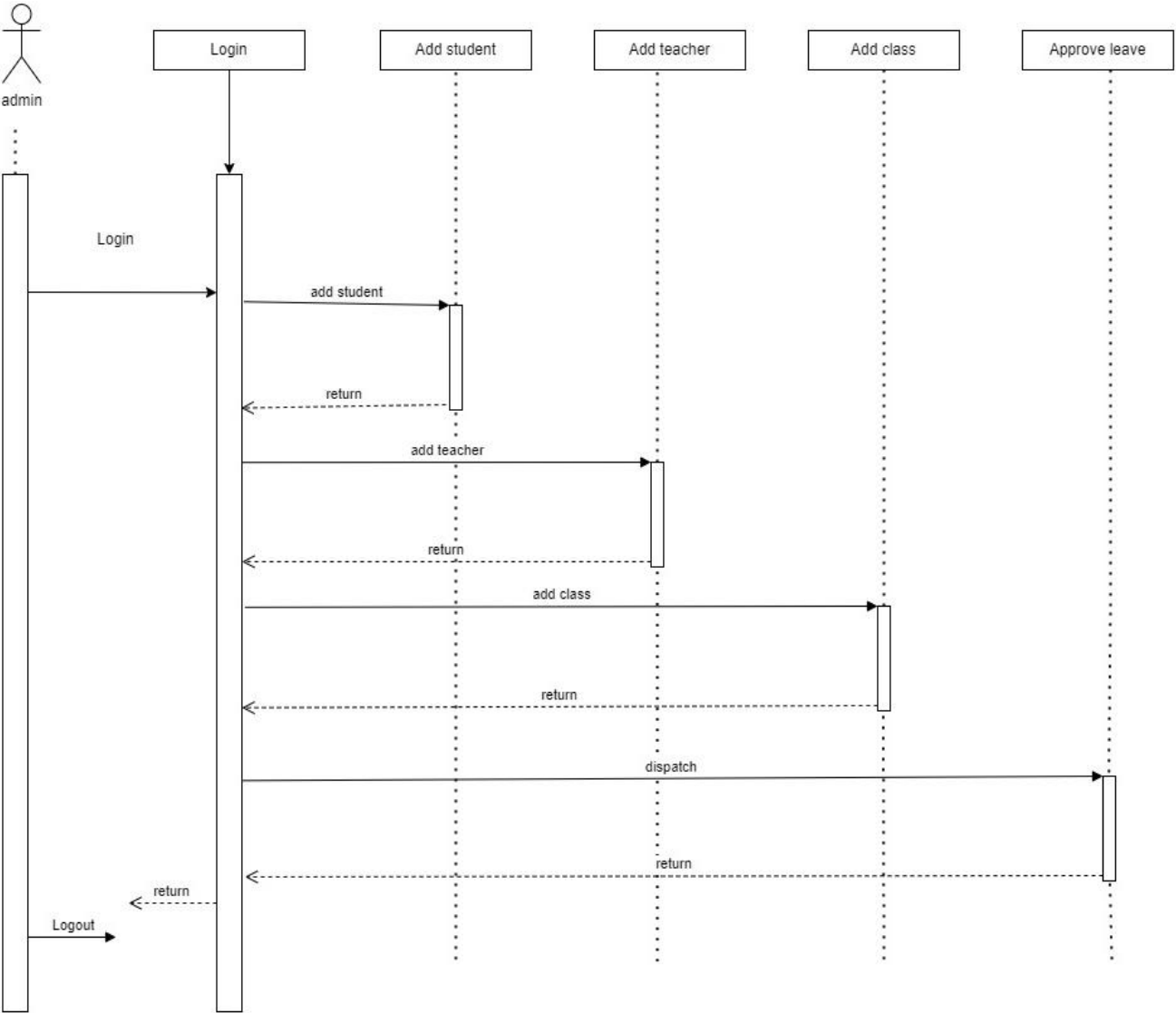
Actors – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

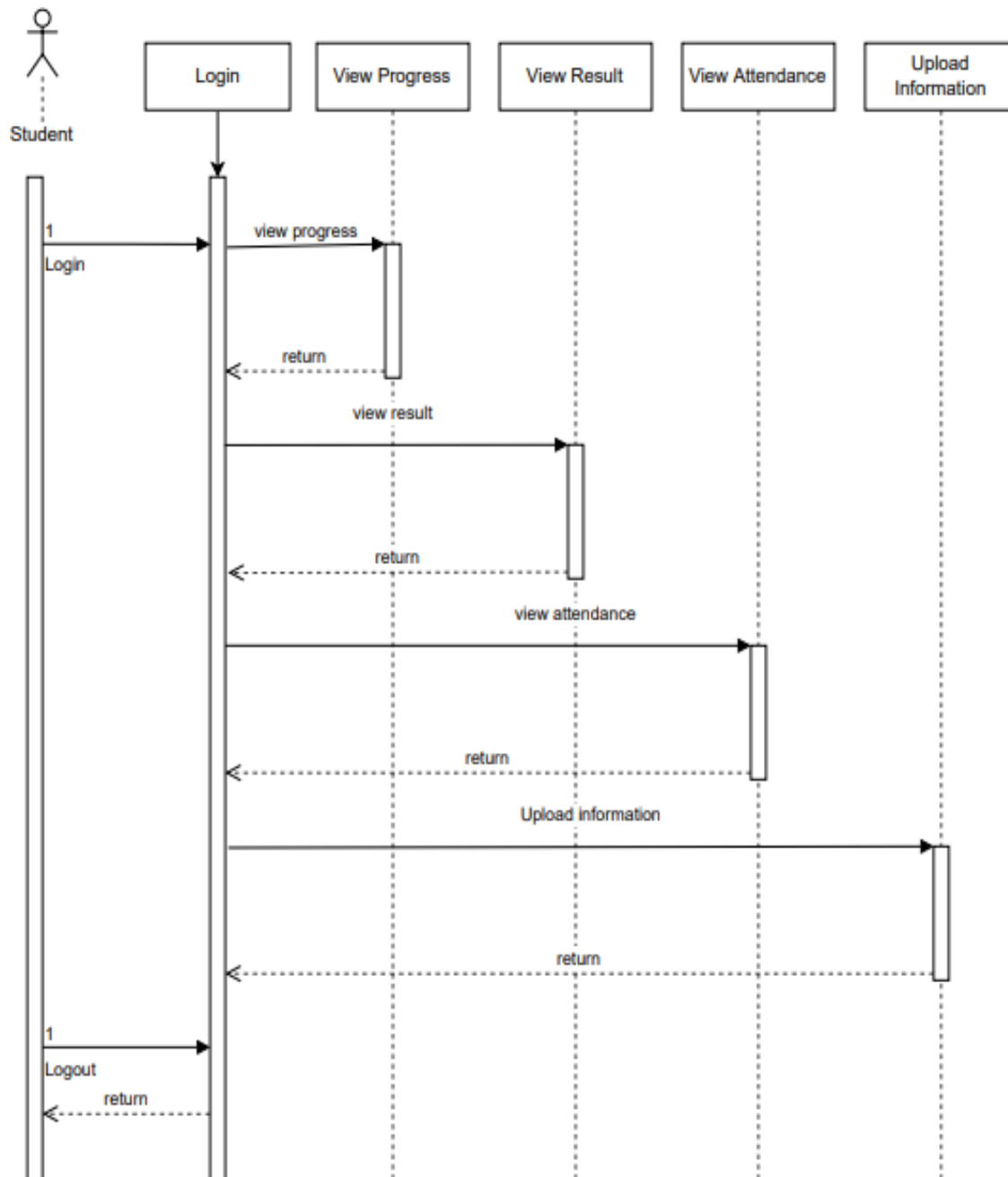
Lifelines – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram

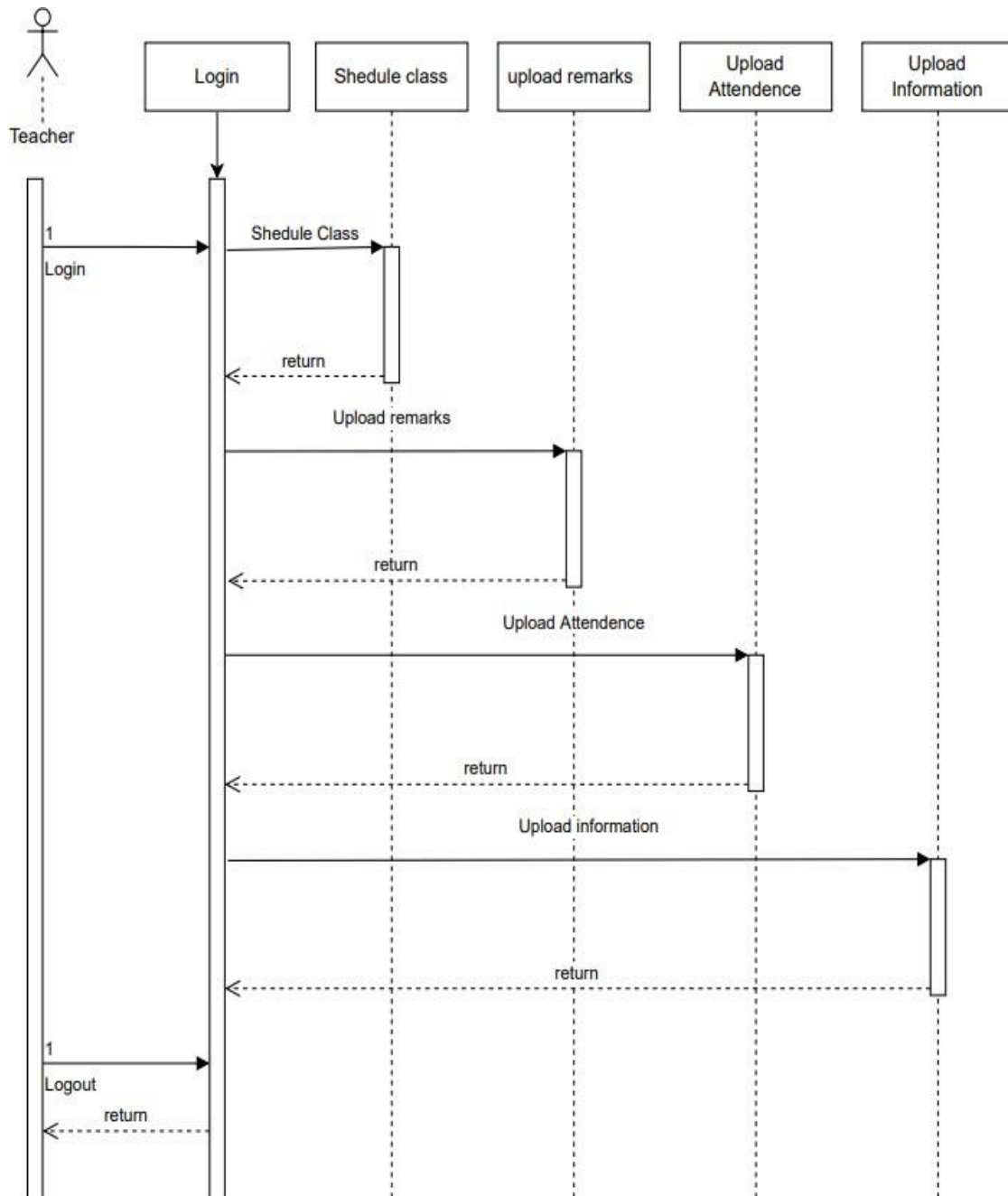
Messages – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Guards – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

ADMIN

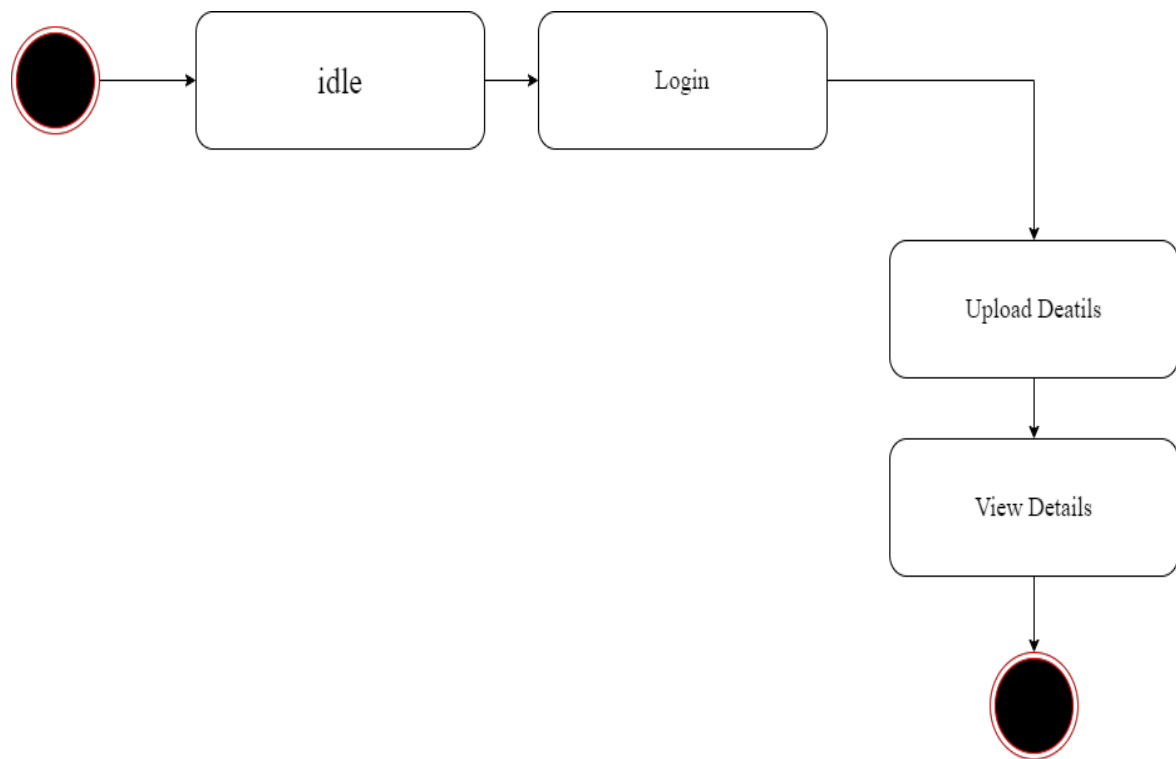


STUDENT

TEACHER

4.2.2 State Chart Diagram

The behavior of classes in reaction to outside stimuli is depicted in a state diagram. A state diagram specifically shows how one object behaves in response to a sequence of system events. It is also sometimes referred to as a state machine diagram. This UML diagram represents the dynamic flow of control for a specific object within a system as it changes states.

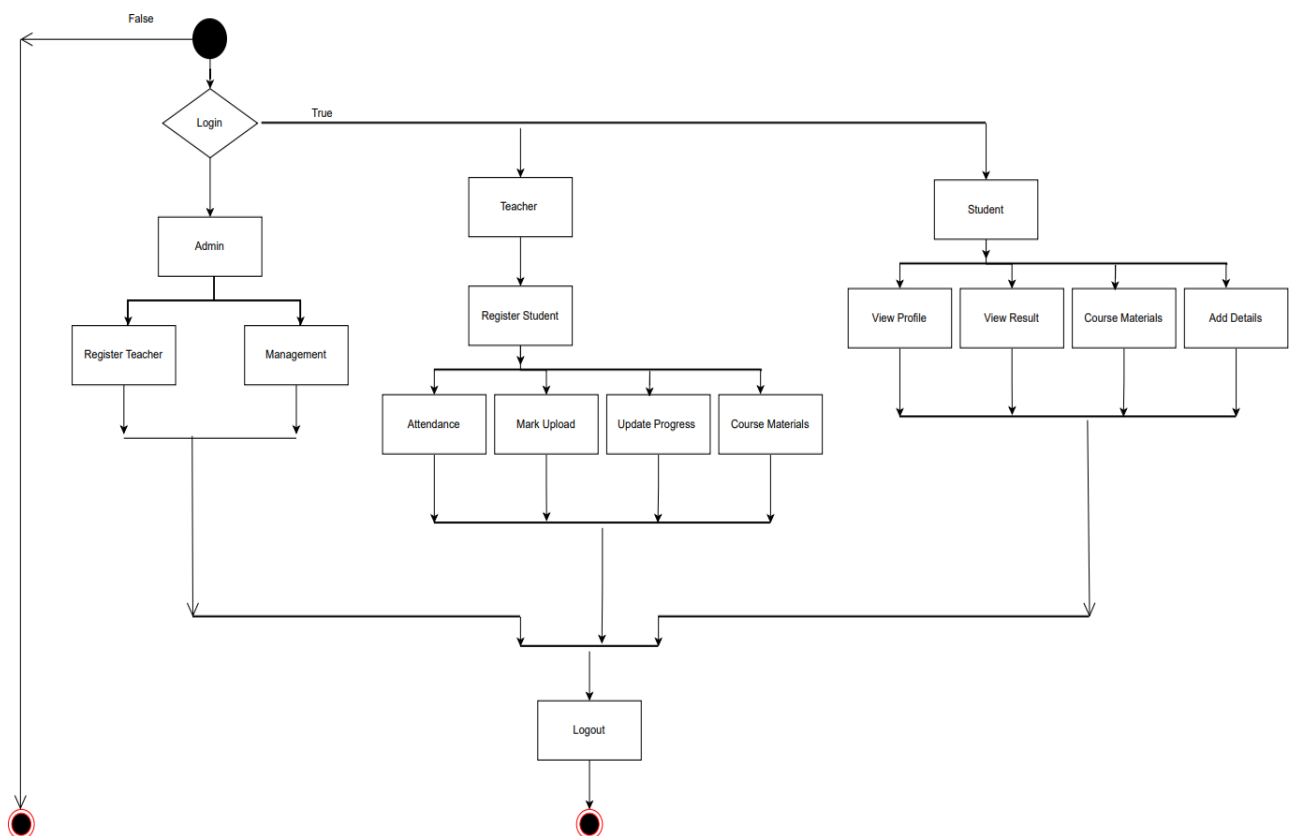


4.2.2 Activity Diagram

Activities, states, and transitions between activities and states are all included in activity diagrams.

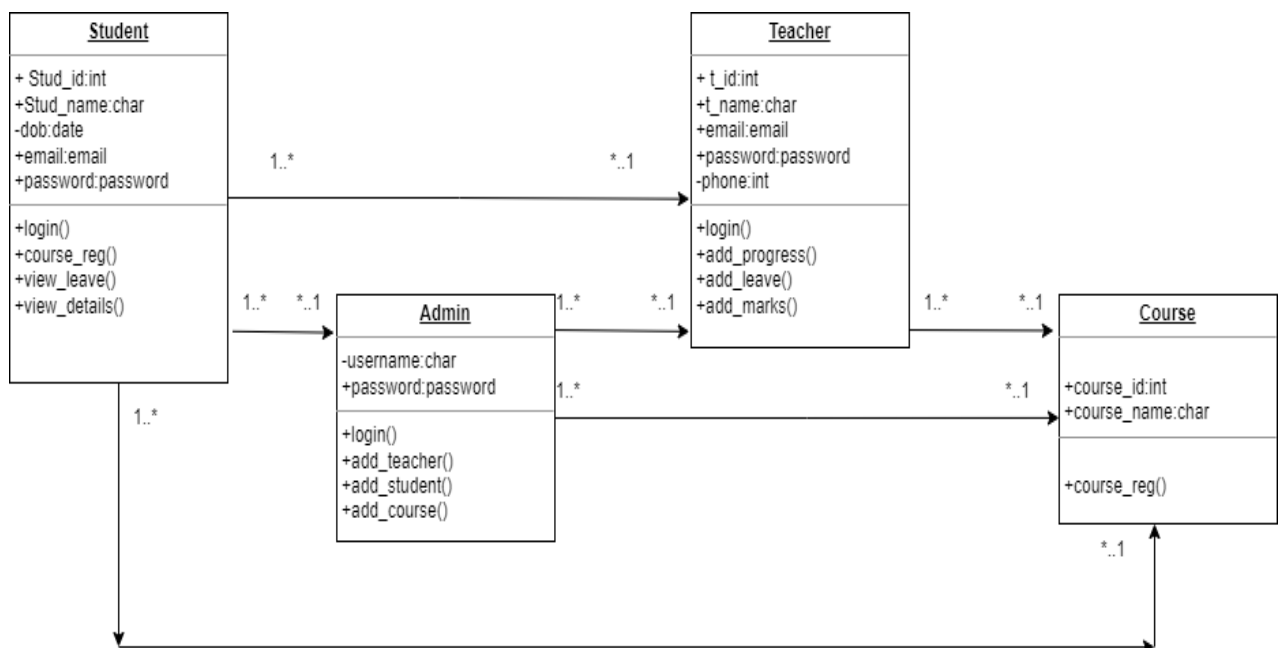
Activities Diagrams explain

- how a service is provided by coordinating activities
- the activities required to complete some operation.
- the connections between the events in a single use case.
- how a series of use cases work together to form a workflow for an organization



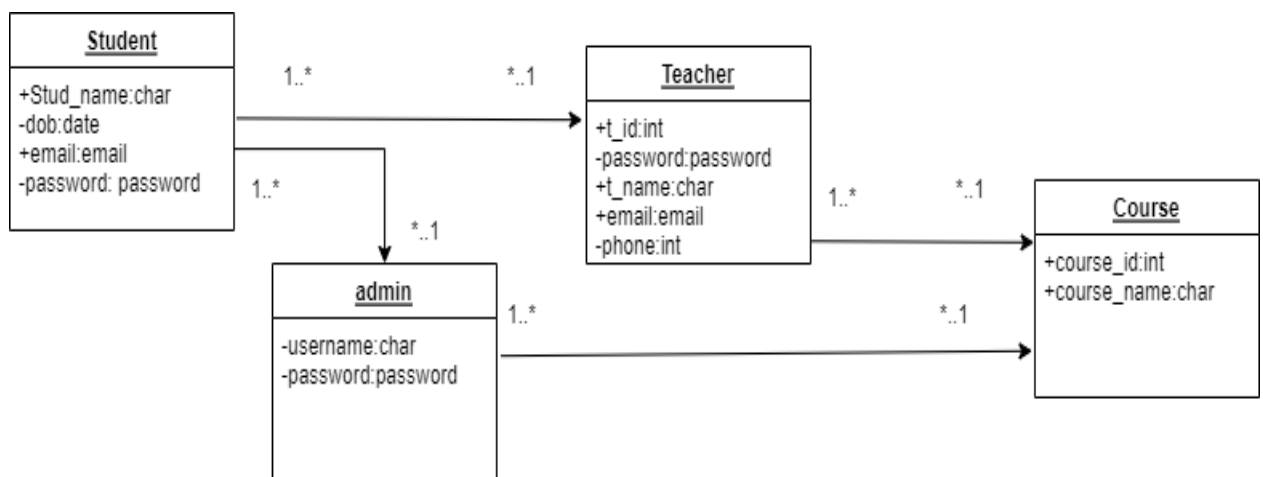
4.2.3 Class Diagram

Class diagrams is a Static diagram. It represents the application's static view. Class diagrams are used to create executable code for software applications as well as for visualizing, explaining, and documenting various elements of systems. The characteristics and functions of a class are described in a class diagram, along with the restrictions placed on the system. Because they are the only UML diagrams that can be directly transferred to object-oriented languages, class diagrams are frequently employed in the modelling of object-oriented systems. A collection of classes, interfaces, affiliations, collaborations, and constraints are displayed in a class diagram.



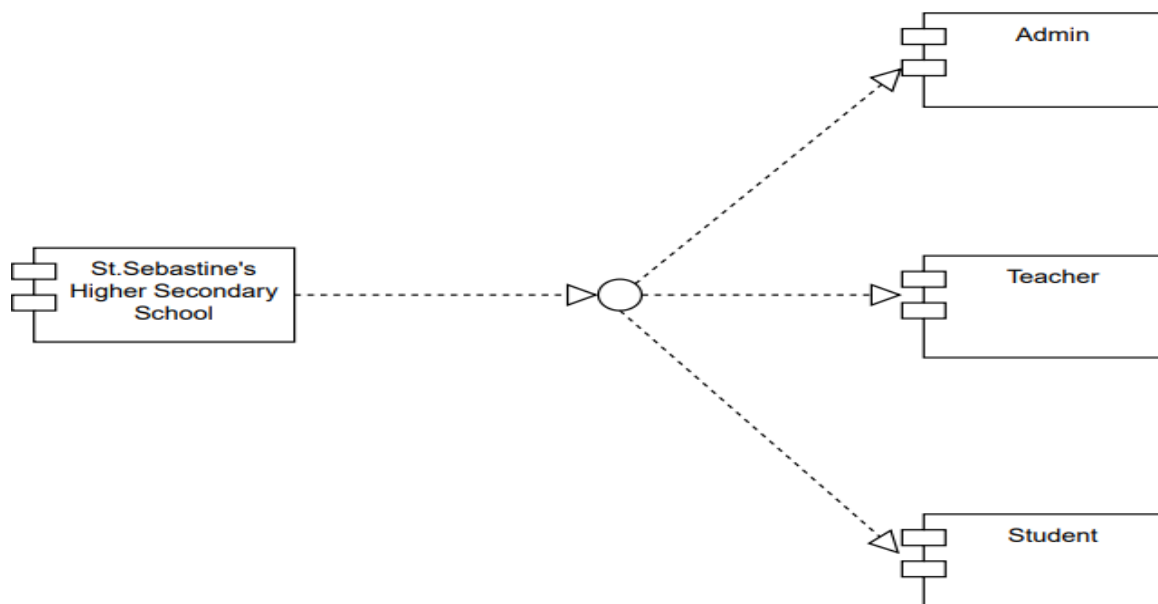
4.2.4 Object Diagram

Class diagrams are a requirement for object diagrams because they are the source of class diagrams. An object diagram illustrates a specific instance of a class diagram. The basic concepts of class diagrams and object diagrams are the same. Object diagrams are also used to describe a system's static view, which is a snapshot of the system taken at a particular point in time. You can see a group of things and their relationships by using object diagrams.



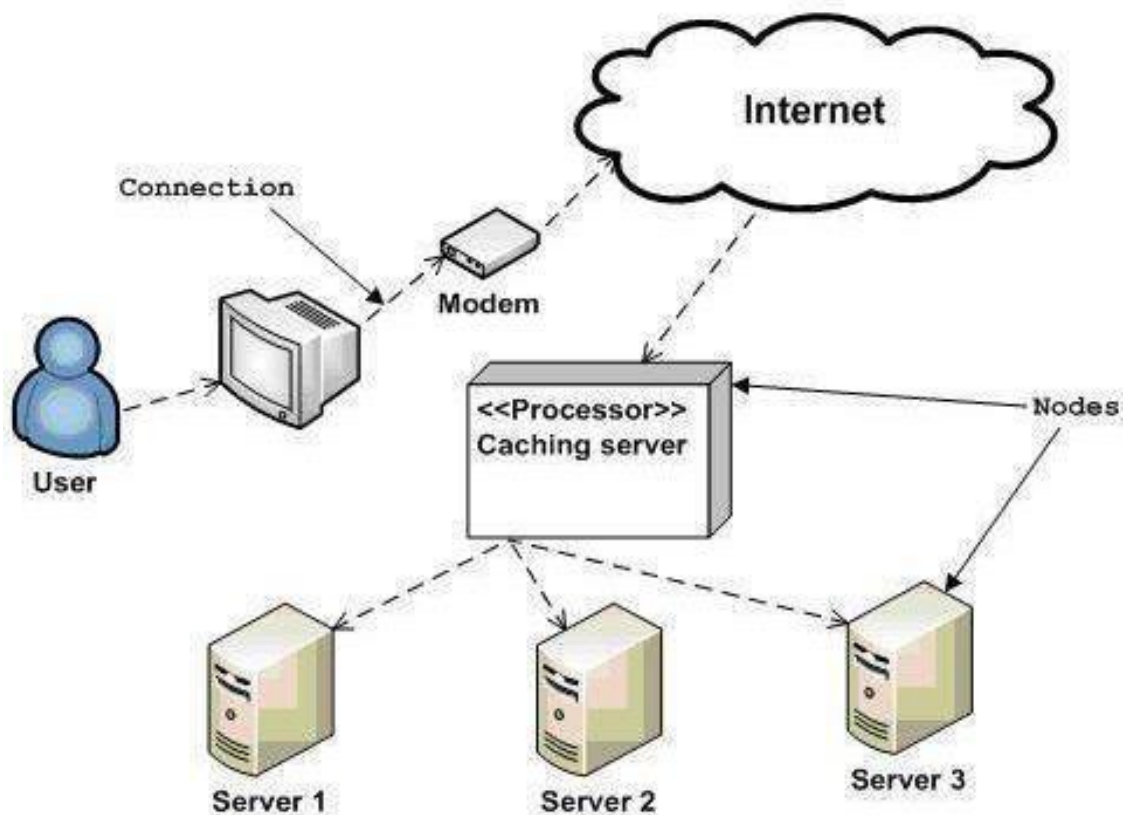
4.2.5 Component Diagram

Component diagrams come in a variety of behaviors and personalities. The physical parts of the system are represented using component diagrams. Executables, libraries, files, documents, and other items that are physically present in a node are just a few examples. Component diagrams are used to show how the components of a system are connected and arranged. These diagrams can also be used to construct systems that can be run.



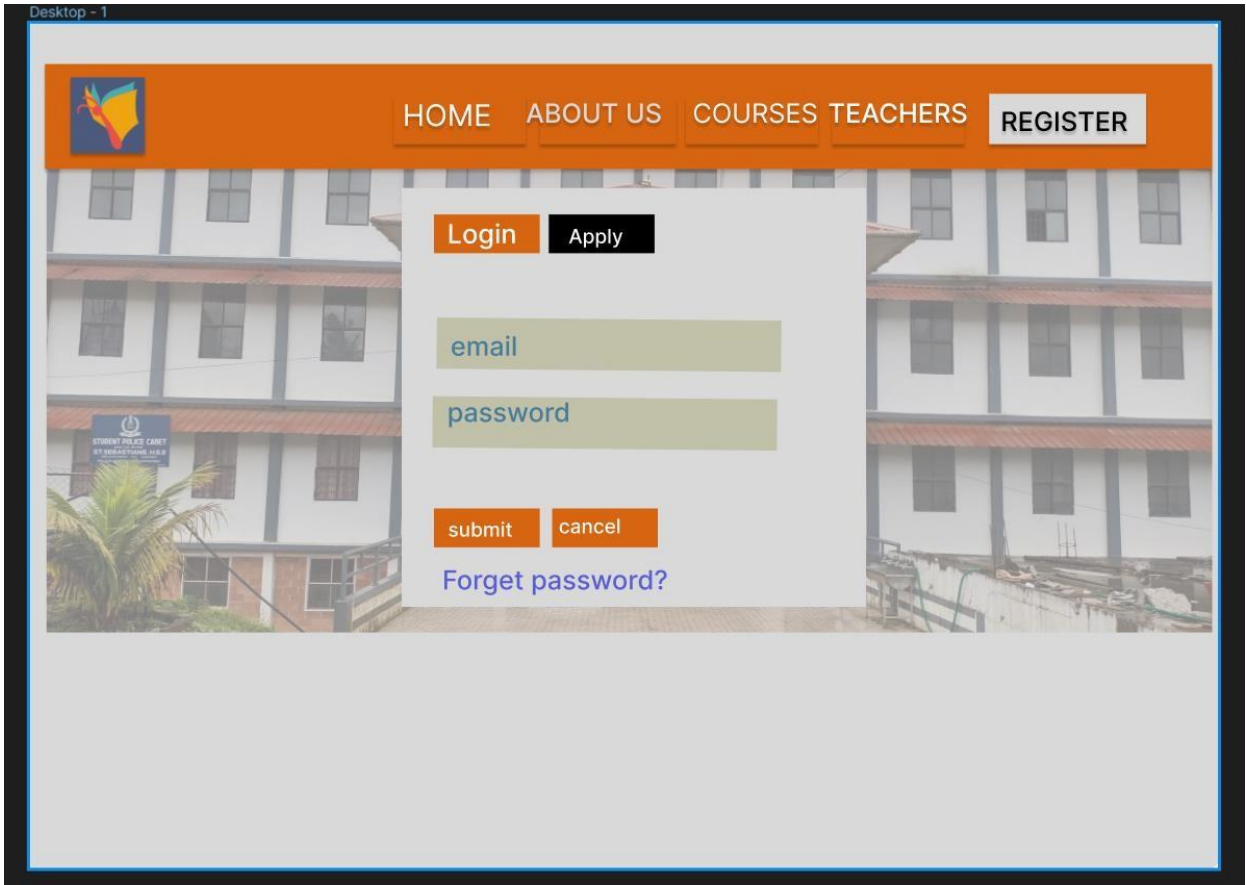
4.2.8 Deployment Diagram

Deployment diagrams show the topology of a system's physical components, where the software components are installed. Deployment diagrams are used to describe a system's static deployment view. The key elements of deployment diagrams are nodes and connections between them.



4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1. Form Name: Login



4.3 DATABASE DESIGN

A database is a structured system with the ability to store information and allow users to access that information quickly and effectively. Any database must be protected because its primary goal is its data. The database design process has two stages. User requirements are obtained in the first step, and a database is created to as clearly as possible meet these objectives. Information Level Design is the name of this stage, which is carried out independently of any specific DBMS. The second phase involves converting this information level design into a design for the DBMS that will be used to implement the system in issue.

4.4.1 Relational Database Management System (RDBMS)

In a relational model, the database is shown as a set of relations. each connection like a file or table of records with values. formal terminology for a relational model, A column header is known as an attribute, a row is known as a tuple, and the table is known as a relation. Each table in a relational database is made up of data that is stored in rows and columns. assigned an arbitrary name. In a story, each row represents a group of associated values.

Domains, Relations, and Attributes

A relation is a table. Tuples are the units of a table's rows. An ordered group of n elements is a tuple. Attributes are referred to as columns. Every table in the database has relationships already established between them. This guarantees the integrity of both referential and entity relationships. A group of atomic values make up a domain D. Specifying the data type from which the domain's data values are derived is a standard way to define a domain. To make it easier to understand the values of the domain, it is also helpful to give it a name. Each value in a relation is atomic and cannot be broken down.

4.4.2 Normalization

The simplest possible grouping of data is used to put them together so that future changes can be made with little influence on the data structures. Data normalization is a formal process. structures in ways that encourage integrity and remove duplication. Normalization is a method of dividing large datasets into smaller ones and removing superfluous fields. Into a smaller table. Additionally, it serves to prevent additions, deletions, and updates. Anomalies. Keys and relationships are two notions commonly used in data modelling. A table row is uniquely identified by its key. key uniquely identifies a row in a table. A primary key is an element, or set of components, in a table that serves as a means of distinguishing between records from the same table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalized means placing things in their natural form, as the

name suggests. By using normalization, the application developer aims to establish a coherent arrangement of the data into appropriate tables and columns, where names may be quickly related to the data by the user.

4.4.3 Sanitization

First Normal Form

According to the First Normal Form, any attribute's tuple's value must be a single value from its domain, which must only contain atomic values. the territory of that property. To put it another way, 1NF forbids "relations within relations. "Alternatively, "relations as attribute values within tuples." The sole attribute values that are allowed by 1NF are indivisible or single-atom values. The data must be entered into Initial as the first step. Standard Form. By putting the data in separate tables, you may donate this.in each table is of a similar type. A primary key or foreign key is assigned to each table as per the project's requirements. For each non-atomic relationship, we create new ones in this. attribute or nested relation. This eliminates repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only connection or nested attribute. This got rid of data groups that were repeated.

Example:

bch_id	bch_name	c_id	date	status
13	2022-STD5	32	2022-06-29	0
14	2022-STD6	33	2022-06-29	0
15	2022-STD7	34	2022-06-29	0
22	2022-STD1	59	2022-07-07	0
23	2022-STD2	60	2022-07-07	0
24	2022-STD3	61	2022-07-07	0
25	2022-STD4	78	2022-07-12	0

Second Normal Form

Accordance with Second Normal Form No non-key attribute should be functionally dependent on a portion of the primary key for relations when the main key has several attributes. This involves breaking down each partial key into its dependent characteristics and setting up a new relation for each one. Keep the original primary key and any properties that are entirely dependent on it in your database. This procedure aids in removing data that depends only on a small portion of the key. If and only if a relation satisfies all the

requirements for first normal form for the primary key and every non-primary key attribute of the connection is completely dependent on its primary key alone, then that relation is said to be in second normal form.

Table 1		Table 2	
STUD_NO	COURSE_NO	COURSE_NO	COURSE_FEE
1	C1	C1	1000
2	C2	C2	1500
1	C4	C3	1000
4	C3	C4	2000
4	C1	C5	2000
2	C5		

Third Normal Form

According to the Third Normal Form relation should not have a non-key attribute that is functionally determined by another non-key attribute or by a collection of non-key attributes. The primary key should not be transitively dependent, in other words. The non-key attributes that functionally determine other non-key attributes are decomposed in this way put up in relation. This action is made to remove anything that is not completely dependent on the Primary Key. Only when a relation is in second normal form and, more importantly, when its non-key characteristics do not depend on those of other non-key attributes, is it considered to be in third normal form.

EMP_ZIP	EMP_STATE	EMP_CITY
201010	UP	Noida
02228	US	Boston
60007	US	Chicago
06389	UK	Norwich
462007	MP	Bhopal

4.4.3 Sanitization

An automated procedure called "sanitization" is used to get a value ready for use in a SQL query. This process typically involves checking the value for characters that have a special significance for the target database. To prevent a SQL injection attack, you must sanitize (filter) the input string while processing a SQL query based on user input. For instance, the user and password input are a typical scenario. In that scenario, the server response would provide access to the 'target user' account without requiring a password check.

4.4.4 Indexing

By reducing the number of disk accesses needed when a query is completed, indexing helps a database perform better. It is a data structure method used to locate and access data in a database rapidly. Several database columns are used to generate indexes. The primary key or candidate key of the table is duplicated in the first column, which is the Search key. To make it easier to find the related data, these values are kept in sorted order. Recall that the information may or may not be kept in sorted order

4.5 TABLE DESIGN

1. Table name: **tbl_login**

Description: To store the username and password

Primary Key: email

SI.No	Field	Data_type	Key constraints	Description
1	fname	Varchar(30)	Not Null	First Name of the user
2	lname	Varchar(30)	Not Null	Last name of the user
3	email	email	Primary Key	Email id of user
4	Password	Password	Not Null	Password of the user
5	Department	Varchar(30)	Not Null	Department of the user
6	Year of join	Varchar(20)	Not Null	Year of Join
7	type	Varchar(30)	Not Null	Type of the user either student or Teacher

2. Table name: **teacher_detail**

Description: To store the teacher details

Foreign Key: email references from tbl_login

SI.No	Field	Data_type	Key constraints	Description
1	email	email	Foregin Key	Email id of teacher
2	fname	Varchar(30)	Not Null	First name of the user
3	lname	email	Primary Key	Last Name of the Teacher
4	hname	Password	Not Null	House Name of the Teacher
5	mobile	Varchar(30)	Not Null	Mobile number of the teacher
6	dob	Varchar(20)	Not Null	Date of birth
7	caste	Varchar(30)	Not Null	Caste of the teacher
8	place	Varchar(30)	Not Null	Place of the teacher

3. Table name: Student_detail

Description: To store the Student details

Foreign Key: email references from tbl_login

SI.No	Field	Data_type	Key constraints	Description
1	id	Int(5)	Primary Key	Id of student details
2	fname	Int(5)	Not Null	First Name of Student
3	lname	Varchar(10)	Not Null	Last Name of Student
4	hname	Varchar(15)	Not Null	House name of the student
5	father	Varchar(10)	Not Null	Name of the father
6	occupation	Varchar(15)	Not Null	Occupation of parent
7	Profile_pic	Varchar(10)	Not Null	Upload the photo
8	dob	Varchar(10)	Not Null	Date of birth
9	religion	Varchar(10)	Not Null	Religion of the student
10	pschool	Varchar(15)	Not Null	Previous school of the student
11	Mark_obtained	Varchar(15)	Not Null	Marks obtained in previous school
12	gender	Varchar(15)	Not Null	Gender of the student
13	gname	Varchar(15)	Not Null	Guardian's name
14	Gmobile	Varchar(15)	Not Null	Mobile number
15	gmail	email	Not Null	Email of the guardian

4. Table name: **Student_detail**

Description: To store the Student details

Sl.No	Field	Data_type	Key constraints	Description
1	Course_name	Int(5)	Primary Key	Id of the second language
2	C_image	Varchar(20)	Not Null	Name of the second language
3	Description	Varchar(20)	Not Null	Description about the Course

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software testing is the process of carefully controlling the execution of software to determine whether it behaves as intended. The words verification and validation are frequently used in conjunction with software testing. Validation is the process of examining or evaluating product, including software, to determine whether it complies with all relevant specifications. One type of verification, software testing, uses methods including reviews, analyses, inspections, and walkthroughs as well.

Checking that what has been specified matches what the user truly desired is the process of validation. The processes of static analysis and dynamic analysis are additional ones that are frequently related to software testing. Static analysis examines the software's source code, searching for issues and obtaining statistics without running the code. Dynamic analysis. Static analysis examines the software's source code, searching for issues and obtaining statistics without running the code. Dynamic analysis examines how software behaves while it is running to provide information including test coverage details, time profiles, and execution trails. A series of activities known as tests can be prepared in advance and carried out. Systematically, beginning with the module level, testing progresses toward integration. a system entirely based on computers. Testing is necessary and cannot be done without. There are numerous guidelines that can be used to ensure the accomplishment of the system testing objectives.

5.2 TEST PLAN

A test plan suggests several required steps that need be taken in order to complete various testing methodologies. The activity that is to be taken is outlined in the test plan. A computer program, its documentation, and associated data structures are all created by software developers. It is always the responsibility of the software developers to test each of the program's separate components to make sure it fulfils the purpose for which it was intended. In order to solve the inherent issues with allowing the builder to evaluate what they have developed, there is an independent test group (ITG). Testing's precise goals should be laid forth in quantifiable language. so that the cost to discover and remedy the problem, the mean time to failure. the cost to find and fix the defects, remaining defect density or frequency of occurrence and test workhours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

5.2.1 Unit Testing

Unit testing concentrates verification efforts on the software component or module, which is smallest unit of software design. The component level design description is used as a guide when testing crucial control paths to find faults inside the module's perimeter. The level of test complexity and the untested area determined for unit testing. Unit testing is white box focused, and numerous components may be tested simultaneously. To guarantee that data enters and exits the software unit under test properly, the modular interface is tested. To make sure that data temporarily stored retains its integrity during each step of an algorithm's execution, the local data structure is inspected. To confirm that each statement in a module has been executed at least once, boundary conditions are evaluated. Finally, each path for managing errors is examined. Before starting any other test, tests of data flow over a module interface are necessary. All other tests are irrelevant if data cannot enter and depart the system properly. An important duty during the unit test is the selective examination of execution pathways. Error circumstances must be foreseen in good design, and error handling paths must be put up to cleanly reroute or halt work when an error does arise. The final step of unit testing is boundary testing.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries. Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working and gives the expected result.

5.2.2 Integration Testing

Integration testing is a methodical approach for creating the program's structure while also carrying out tests to find interfacing issues. The goal is to construct a program structure that has been determined by design using unit tested components. The program is tested. Correction is challenging since the size of the overall program makes it challenging to isolate the causes. As soon as these mistakes are fixed, new ones arise, and the process repeats itself in an apparently

unending cycle. All the modules were integrated after unit testing was completed in the system to check for any interface inconsistencies. A distinctive program structure also developed when discrepancies in program structures were eliminated.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested with all code, class modules, forms, and modules. Popular names for this type of testing include system tests and black box testing. The functional requirements of the software are the main emphasis of the black box testing approach. That example, using Black Box testing, a software engineer can create sets of input conditions that will fully test every program requirement. The following sorts of problems are targeted by black box testing: erroneous or missing functions, interface errors, data structure or external data access errors, performance errors, initialization errors, and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

1. Input Screen Designs
2. Output Screen Designs

The above testing is done taking various kinds of test data. Preparation of test data plays a Vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future us.

5.2.5 Automation Testing

Software and other computer goods are tested automatically to make sure they abide by tight guidelines. In essence, it's a test to ensure that the hardware or software performs exactly as intended. It checks for errors, flaws, and any other problems that might occur throughout the creation of the product. Any time of day can be used to do automation testing. It looks at the software using scripted sequences. It then summarizes what was discovered, and this data can be compared to results from earlier test runs.

5.2.6 Selenium Testing

An open-source program called Selenium automates web browsers. It offers a single interface that enables you to create test scripts in several different programming languages, including Ruby, Java, NodeJS, PHP, Perl, Python, and C#. Web application testing for cross-browser compatibility is automated using the Selenium testing tool. No matter if a web application is responsive, progressive, or standard, it is employed to assure good quality. Selenium is a free software program.

Example:

Test Case 1

Code

```
package test1;
import org.openqa.selenium.By;
public class login {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\Sangeetha\\Downl...");
        WebDriver driver=new ChromeDriver();
        driver.get("http://localhost/Crime-Reporting-System_user/userlogin.php");
        driver.findElement(By.id("exampleInputEmail")).sendKeys("chithra@gmail.com");
        driver.findElement(By.id("exampleInputPassword1")).sendKeys("Chithra@12");
        driver.findElement(By.id("login")).click();
        String actualUrl="http://localhost/Crime-Reporting-System_user/userhome.php";
        String expectedUrl= driver.getCurrentUrl();
        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
            System.out.println("Test passed");
        } else {
            System.out.println("Test failed");
        }
    }
}
```

Eg.Screenshot

```

Starting ChromeDriver 103.0.5060.53 (a1711811edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/50600@#)
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver secure.
ChromeDriver was started successfully.
Jul 21, 2022 3:08:40 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 21, 2022 3:08:40 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed

```

Eg.Test Report

Test Case 1					
Project Name:					
Login Test Case					
Test Case ID: Test_1			Test Designed By:		
Test Priority(Low/Medium/High):			Test Designed Date:		
Module Name:			Test Executed By :		
Test Title :			Test Execution Date:		
Description:					
Pre-Condition :User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)
1					

2					
3					
4					
5					
6					
7					
Post-Condition:					

Test Case 2:

Code

Screenshot

Test report

Minimum 4 test cases (1 login 3 functionalities)

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. Gaining users' trust that the new system will function is perhaps the stage most important to the success of a new system. It will be precise and effective. It is mostly focused on user education and documentation. Normally, conversion happens about the same time the user is being or later, training. Simply said, implementation is the gathering of a new system design into operation, the procedure for transforming a fresh, updated system design into an functional one The user department now bears the most of the workload, faces the most disruption, and has the biggest influence on the current system. If the implementation is not well thought out or managed, confusion and mayhem may result. Implementation encompasses all of the steps used to switch from the old system to the new one. The new system could be entirely different, take the place of an existing manual or automated system, or it could be modified to work better. A reliable system that satisfies organizational needs must be implemented properly. System implementation refers to the process of using the built system. This comprises all the processes involved in switching from the old to the new system. Only after extensive testing and if it is determined that the system is operating in accordance with the standards can it be put into use. The system personnel assess the system's viability.

6.2 IMPLEMENTATION PROCEDURES

Software implementation refers to the complete installation of the package in its intended environment, as well as to the system's functionality and satisfaction of its intended applications. The software development project is frequently commissioned by someone who will not be using it. People first have their doubts about the software, but we must make sure that they do not become resistant. build up, as one must make sure that the active user must be aware of the benefits of using the new system. Their confidence in the software is built up. Proper guidance is imparted to the user so that he is comfortable in using the application. Before examining the system, the user must be aware that the server software needs to be running on the server to access the results. The actual process won't happen if the server object is not active and functioning on the server.

6.2.1 User Training

The purpose of user training is to get the user ready to test and modify the system. The people who will be involved must have faith in their ability to contribute to the goal and benefits anticipated from the computer-based system. Training is more necessary as systems get more complicated. The user learns how to enter data, handle error warnings, query the database, call up routines that will generate reports, and execute other important tasks through user training.

6.2.2 Training on the Application Software

The user will need to receive the essential basic training on computer awareness after which the new application software will need to be taught to them. This will explain the fundamental principles of how to use the new system, including how the screen work, what kind of help is displayed on them, what kinds of errors are made while entering data, how each entry is validated, and how to change the data that was entered. Then, while imparting the program's training on the application, it should cover the information required by the particular user or group to operate the system or a certain component of the system. It's possible that this training will vary depending on the user group and the level of hierarchy.

6.2.3 System Maintenance

The mystery of system development is maintenance. A software product works effectively during the maintenance phase of the software cycle. after a system has been implemented successfully, it should be appropriately maintained. System maintenance is a crucial phase in the software development life cycle. Maintenance is necessary for a system to be adaptable to changes in the system environment. Of all, maintaining software entails much more than merely "Finding Mistakes."

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

. St. Sebastian's HSS School Management System is simple website which provides many functionalities than the existing system. In this system the students want to login to the system to use the features provided in the system. There are three modules in this system admin, teacher, and student. Admin has the whole power to manage the system which includes approval of leave, adding courses etc. Teacher has the power to upload the documents, details, set examination etc. Student has the power to apply leave, upload the details, view examination, download the question paper, course materials etc. Admin can view all the details and can download the leave details as a csv file.

7.2 FUTURE SCOPE

.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Python: The Complete Reference Python – 20 March 2018 by Martin C. Brown
- Let Us Python by Kanetkar Yashavant
- Roger S Pressman, “*Software Engineering*”, 1994.
- PankajJalote, “*Software engineering: a precise approach*”, 2006.

WEBSITES:

- www.w3schools.com
- www.stackoverflow.com
- <https://www.djangoproject.com/>
- https://www.w3schools.com/Css/css_intro.asp
- https://stackoverflow.com/questions/730207/django-model-set-foreign-key-to-a-field-of-another-model#:~:text=You%20need%20to%20use%20%22to_field%22%20in%20%22models.ForeignKey%20%28%29%22,a%20different%20field%2C%20that%20field%20must%20have%20unique%3DTrue.

CHAPTER 9

APPENDIX

9.1 Sample Code

Main functionalities

9.1 Screen Shots

