PAPER NAME

03_LightedToLighten.docx

WORD COUNT	CHARACTER COUNT
12523 Words	75770 Characters
PAGE COUNT	FILE SIZE
87 Pages	4.1MB
SUBMISSION DATE	REPORT DATE
Apr 4, 2024 9:17 AM GMT+5:30	Apr 4, 2024 9:18 AM GMT+5:30

● 40% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 26% Internet database
- Crossref database
- 39% Submitted Works database
- 8% Publications database
- Crossref Posted Content database

1.1 PROJECT OVERVIEW

A Sunday School Management System's main goal is to centralize and simplify a parish's administrative processes while facilitating communication, boosting engagement with students and teachers, and enhancing the organization and accessibility of resources or information. The system can increase productivity, improve communication and engagement, and provide better tracking and reporting of activities by digitizing and automating many administrative tasks. Additionally, it enables users to utilize resources and services remotely, which can be helpful in circumstances like the COVID-19 pandemic.

1.2 PROJECT SPECIFICATION

Users:

1. Priest:
    - Profile management
    - View Sunday school activities or events.
    - Advanced Search functionality.

2. Teachers:

- Profile management
- Add, edit, and view student details of their class
- Record attendance.
- Upload educational resources.
- Mark updating
- Schedule online classes
- Counseling sessions for students
- Set online exam

3. Students:

- Profile management
- View attendance status
- Leave application
- Download educational resources
- View internal exam results and diocesan exam results.
- Certificate request
- Talent search registration and fee payment
- Attend Online Classes

4. Headmaster
    - Teachers and student management
    - Teacher’s allotment
    - Circulars and Announcement
    - Progress Report generation
    - Schedule exams
    - Student grade promotion
    - Certificate generation
    - Organizing activities

5. Parents:
    - Profile management
    - Attendance monitor
    - Student academic performance evaluation
    - Download mark sheet
    - Access to learning resources

CHAPTER 2

SYSTEM STUDY

2.1INTRODUCTION

A system study is a key step in the system development process when the current system's operations and features are extensively analyzed to acquire comprehensive insights. This phase mostly entails knowing the system's present structure, the subtleties of its components, and how they interrelate. By going deep into the present processes, inefficiencies, bottlenecks, and possible areas of improvement may be found. The knowledge acquired from a system study creates the basis upon which new system requirements are developed, ensuring that the succeeding design and implementation stages are anchored in a full understanding of both user demands and current obstacles. In essence, a system study sets the foundation for informed decision-making and targeted changes throughout the system development lifecycle.

2.2EXISTING SYSTEM

2.2.1 NATURAL SYSTEM STUDIED

According to this system, Sunday school is held in a place where both the students and the teachers can physically attend.

Face-to-face interactions foster a strong sense of community and belonging through activities like religious instruction and current affairs discussions.

System studied:

St.George Marthoma Sunday School, Vazhoor

2.2.2 DESIGNED SYSTEM STUDIED

This system provides a general overview of the entire educational institution while concentrating primarily on the congregation.

2.3DRAWBACKS OF EXISTING SYSTEM

- Only those who can physically visit the location have access to its educational services and activities.
- Communicating events, news, or schedule changes can be challenging and may not reach all members of the community.
- Traditional management techniques can frequently be less effective, requiring more time, resources, and labor-intensive work.
- Information about the students, teachers, and administration is not provided on

samajam.marthoma.in.

2.4PROPOSED SYSTEM

- This system, which focuses on a hybrid approach that combines traditional and digital methods, can help users enjoy the benefits of the above-mentioned systems.
- It focuses on managing users and digitizing religious events for people who cannot attend services in person.

2.5ADVANTAGES OF PROPOSED SYSTEM

- The proposed system typically provides a central repository for all data, making it easy to access and update information, and can improve communication and coordination among teachers, students, and parents.
- The system will also provide a suite of tools for administrators to manage users, schedule events, track donations, and maintain the website.
- A well-designed, user-friendly Sunday school website can play a crucial role in ensuring the continuity of religious activities and community support in times of uncertainty.

22

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study is an analysis and evaluation of a proposed project to determine if it is technically, financially, and operationally feasible. It's often conducted before a project is initiated to assess the potential success and risks of the project. The proposed system may be practical if it satisfies user needs, integrates smoothly with current systems, and complies with legal requirements. Clergy, teachers, students, parents, and the headmaster are just a few of the people who use this system, and it meets their needs. It is user-friendly and has the necessary features, such as resource libraries, communication tools, and event scheduling. The proposed system facilitates Sunday school operations such as resource allocation, member management, and priest tasks. Additionally, it ought to increase the community's involvement and engagement. The new system can only be integrated with communication tasks since Sunday school currently does not use any software for administrative or other purposes.

3.1.1 Economical Feasibility

A system is economically feasible if it delivers financial benefits that outweigh the costs of development, implementation, and maintenance. For a Sunday school Management System, here are several considerations for economic feasibility:

According to this, the system's total cost of ownership—which includes expenses for development, application, instruction, maintenance, and upgrades—will be calculated. Ensure these costs are within the system's budget.

The proposed system helps in the automation of administrative tasks and can cut down on the time and resources needed for management tasks, resulting in a reduction in labor costs. The proposed system provides a secure online platform for donations that might encourage more frequent and larger donations, potentially increasing its revenue. The system can provide better tracking and reporting of income and expenses, enabling more informed and efficient financial decision-making. It might have a one-time fee for maintenance and hosting.

What are the costs of conducting a full system investigation?

➢ the proposed system is developed as part of project work; there is no manual cost to spend on the proposed system.

The cost of the hardware and software

➢ All the resources are already available

3.1.2 Technical Feasibility

It examines whether the system can be created using the available technology and expertise. This depends on the accessibility of reliable hardware and software, developer talent, and crucial infrastructure.

The current technology infrastructure of Sunday school can support the proposed system's requirements. Also, the users have reliable internet connectivity, necessary hardware, and software compatibility.

The suggested system has the same functionalities as the natural system and has a simple user interface. The system can securely handle sensitive data, which calls for technical solutions for data encryption, secure user authentication, etc.

Is the project feasible within the limits of current technology?

➢ Yes

Technical issues raised during the investigation are:

➢ Nothing

Can technology be easily applied to current problems?

➢ Yes

Does technology have the capacity to handle the solution?

➢ Yes

3.1.3 Behavioral Feasibility

Behavioural feasibility assesses how well the proposed system fits within the existing business environment and workflows, how it will meet user requirements, and how users will respond to it.

This may involve conducting surveys or interviews to understand their needs, concerns, and willingness to learn and use the new system.

All users will receive sufficient training to ensure they are aware of how to use the system. This could entail producing user guides and holding training sessions. The system should ideally streamline administrative tasks, simplify communication, and enhance activity management without causing significant disruption. The proposed system will improve current workflows.

Is there sufficient support for the users?

➢ Yes.

Will the proposed system cause harm?

➢ No

3.1.4 Feasibility Study Questionnaire

1. Who is the main head of Sunday school?

   The headmaster is the main head and performs most of the duties personally.

2. What are your current management processes?

   Teacher and student management, resource management

3. How do you currently handle Sunday school registration and updates?

   Students are mostly registered and updated using the traditional approach.

4. What systems do you use for financial management and donation tracking?

   Combination of manual bookkeeping, cash handling, and personal checks

5. Do you have any system to track attendance at the organization?

   No, traditional methods are used to monitor attendance.

6. Do you have any software or system in place for registration?

   No, primarily the person in charge is informed of reservations.

7. How do you handle your prayer request management?

   Daily and weekly management of prayer requests

8. How do you manage and plan the annual budget and financial reports?

   The parish treasurer maintains financial records in a conventional manner.

9. What systems do you have in place for the collection and safe storage of sensitive data?

   Data is currently gathered and kept in a register.

10. What mechanisms do you have for users to give feedback or voice concerns about Sunday school operations?

    Regular parish meetings and a tangible suggestion box are required. Parishioners might also send a letter or message to the parish council.

3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	- i5-10210U CPU, 32 or 64
RAM	- 8.00 GB
Hard disk	-

3.2.2 Software Specification

Front End	-	Html, CSS, JS, Bootstrap
Back End	-	Python, Django
Database	-	SQLite
Client PC	-	Windows 7 and above.

3.3 SOFTWARE DESCRIPTION

3.3.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.

Python is used for server-side web development, software development, mathematics, and system scripting, and is popular for Rapid Application Development and as a scripting or glue language to tie existing components because of its high-level, built-in data structures, dynamic typing, and dynamic binding. Program maintenance costs are reduced with Python due to the easily learned syntax and emphasis on readability. Additionally, Python's support of modules and packages facilitates modular programs and the reuse of code. Python is an open-source community language, so numerous independent programmers are continually building libraries and functionality for it.

3.3.2Django

Django is a high-level Python web framework that enables the rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

Django helps you write software that is:

Complete

Django adheres to the "Batteries included" approach, offering developers nearly all the necessary tools and features they might require right from the start.

Versatile

Django is a versatile web framework that has been utilized to create a wide range of websites such as content management systems, wikis, social networks, and news sites. It is compatible with any client-side framework and can present content in various formats including HTML, RSS feeds, JSON, and XML.

Secure

Django is a framework used for web development that is built to help developers avoid making common security mistakes. It is designed to automatically take necessary security measures to ensure the website is protected from vulnerabilities such as SQL injection, cross-site scripting, cross-site request forgery, and clickjacking.

Scalable

Django employs an architecture that follows a component-based approach with a "shared-nothing" philosophy. Its ability to separate various components facilitates scalability by enabling the addition of hardware at different levels, such as caching servers, database servers, or application servers, as traffic increases.

Maintainable

Django system utilizes certain plan standards and designs that advance the advancement of code that can be effectively kept up and reused. The system moreover takes after the Do not Rehash Yourself (DRY) guideline to dodge duplication of code.

Portable

Django is written in Python, which runs on many platforms. Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

3.3.3 SQLITE

SQLite is a lightweight and self-contained relational database management system. It's noted for its simplicity, speed, and ease of integration, making it a popular option for embedded systems and applications. SQLite runs without a separate server process and offers direct access to the database using a simple and efficient query language. It's appropriate for small to medium-sized applications.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Systems plan is the method of characterizing components of a framework like modules, design, components, and their interfacing and information for a system based on the desired prerequisites. It is the method of characterizing, creating, and planning frameworks that fulfill the particular needs and prerequisites of a trade or organization. A systemic approach is required for a coherent and well-running framework. A Bottom-Up or Top-Down approach is required to take into consideration all related variables of the framework. An architect employs the modeling dialects to specify the data and information in a structure or framework that's characterized by a reliable set of rules and definitions. The plans can be characterized in graphical or printed modeling dialects.

4.2 UML DIAGRAM

The Unified Modeling Language (UML) is a powerful tool for describing software structure and behavior through intuitive graphical notation. This versatile modeling language is widely employed for the design, analysis, and implementation of software systems. By providing a straightforward and standardized way to visualize a software system's architectural properties, UML has established itself as the go-to standard for building Object-Oriented Software. Whether you're a business user, developer, or anyone in need of data modeling, UML blueprints can be an asset. UML is not a development methodology or programming language.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

Use-case diagrams in UML are used to illustrate the behavior of a system and capture its requirements. They provide a high-level overview of the system's functionality and scope, as well as the interactions between the system and its players. The use cases and actors in a use-case diagram explain what the system does and how the players use it, but not how the system runs internally. Use-case diagrams help to clarify the context and needs of a full system or its main components. A complex system can be represented with a single use-case diagram, or multiple diagrams can be generated to describe different parts of the system. Typically, use-case diagrams are created in the early stages of a project and referred to throughout the development process. The following topics describe the elements of a use-case diagram:

Use cases

A use case describes a function that a system performs to achieve the user’s goal. A use case must yield an observable result that is of value to the user of the system.

Actors

An actor represents a role of a user that interacts with the system that you are modeling. The user can be a human user, an organization, a machine, or another external system.

Subsystems

In UML models, subsystems are a type of stereotyped component that represent independent, behavioral units in a system. Subsystems are used in class, component, and use-case diagrams to represent large-scale components in the system that you are modeling.

Relationships in use-case diagrams

In UML, a relationship is a connection between model elements. A UML relationship is a type of model element that adds semantics to a model by defining the structure and behavior between the model elements.

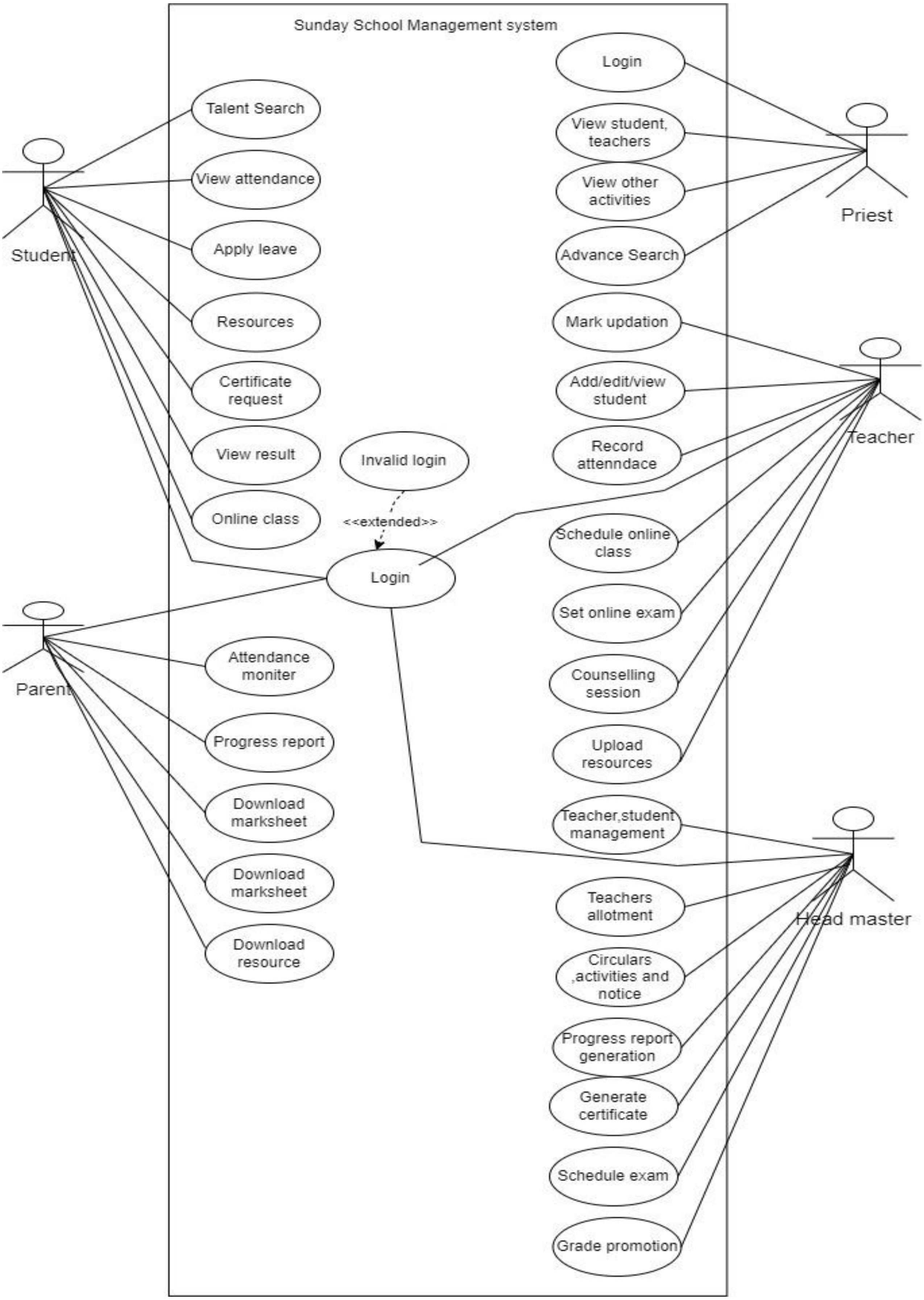


Fig 1. Use case diagram for Sunday School Management System

4.2.1 SEQUENCE DIAGRAM

Sequence diagrams are diagrams that represent the interactions between items in sequential order. Often, developers assume that sequence diagrams are created only for them, much like class diagrams. However, one of the key applications of sequence diagrams is in the transition from requirements stated as use cases to the next more formal level of refinement. Use cases are generally developed into one or more sequence diagrams.

The primary purpose of a sequence diagram is to define event sequences that result in a desired outcome.

Purpose of Sequence Diagram:

To model the interaction between active objects in a system, it is important to model the interaction between object instances within a cooperation that implements a use case. Additionally, it is necessary to model the interaction between elements inside a partnership that achieves an operation. You can either model generic interactions, which show all potential pathways through the interaction, or instances of an interaction, which only show one path through the interaction.

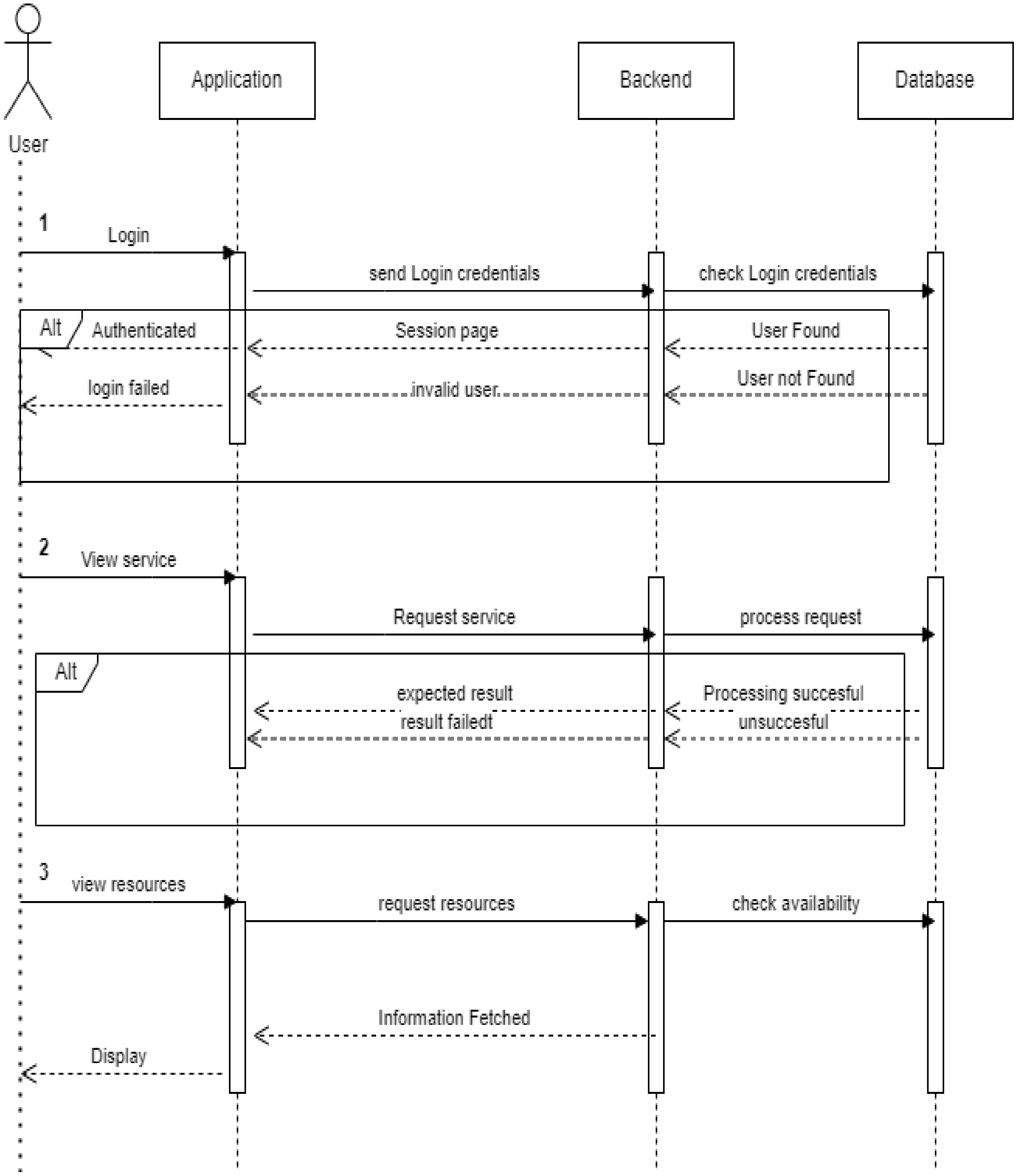


Fig.2 Sequence diagram for Sunday School Management System

4.2.2 State Chart Diagram

A state chart diagram represents a state machine that specifies distinct states of an item and controls them via internal or external events.

Purpose of State chart Diagrams

- State chart diagrams are one of the five types of UML diagrams used to illustrate the dynamic behavior of a system.
- State chart diagrams are utilized to depict reactive systems with multiple states of an entity that change through events.
- Reactive systems react to internal or external events.
- State chart diagrams illustrate the transition of control from one state to another.
- When an item exists, it is in a state that changes when a triggering event occurs.
- The primary goal of a state chart diagram is to illustrate the complete life cycle of an entity, from creation to termination.

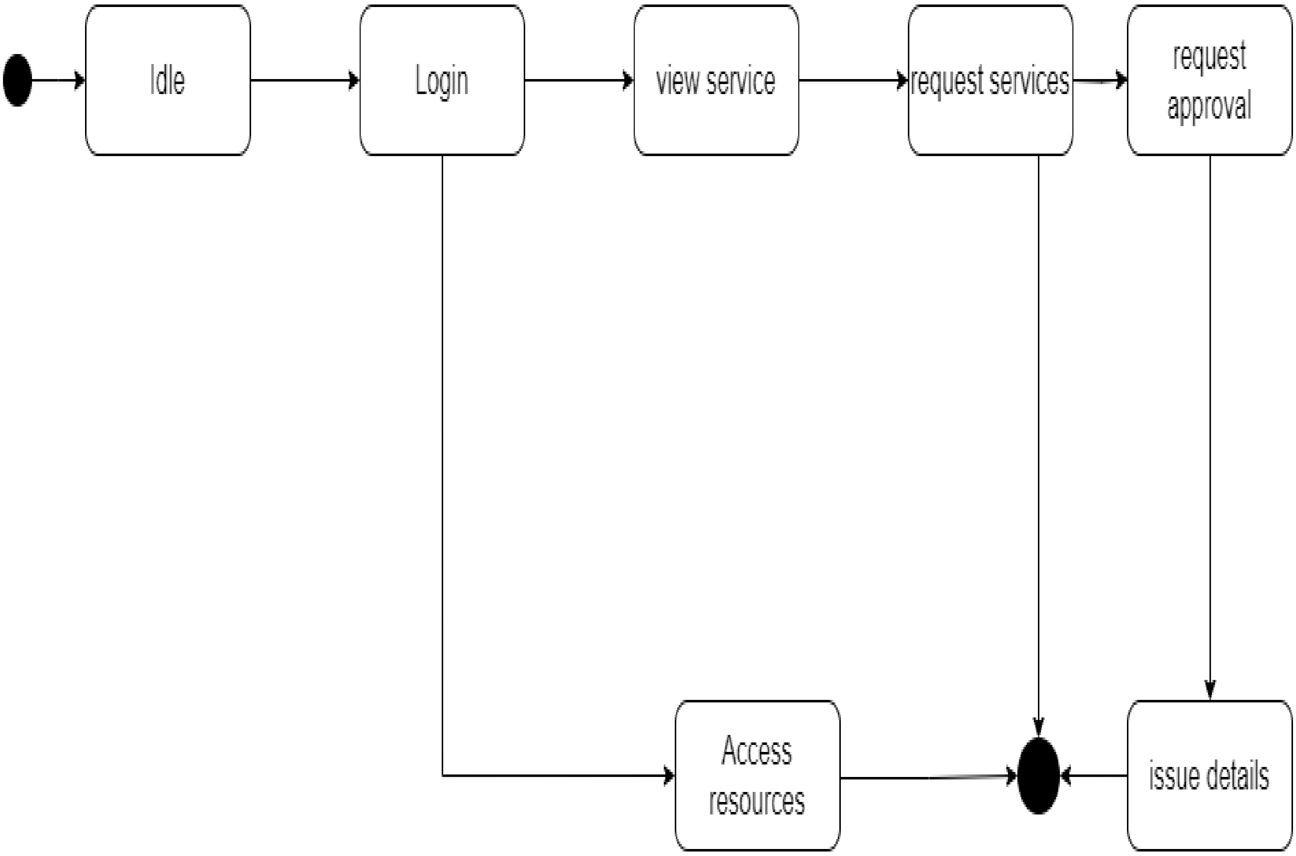


Fig.3 State chart diagram for Sunday School Management System

4.2.2 Activity Diagram

Activity diagram is just a flowchart to illustrate the flow from one activity to another activity. The activity may be regarded as an operation of the system.

The control flow is pulled from one action to another. This flow might be sequential, branching, or concurrent. Activity diagrams deal with every form of flow control by employing various parts such as fork, join.

Purpose of Activity Diagrams

- The main objectives of activity diagrams are identical to other four diagrams. It captures the dynamic behavior of the system.
- The other four diagrams are used to demonstrate the message flow from one object to another, but activity diagram is used to show message flow from one activity to another.
- Action could be a particular operation of the framework.
- Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also utilized to design the executable system by applying forward and reverse engineering approaches.
- The only missing component in the activity diagram is the message section.

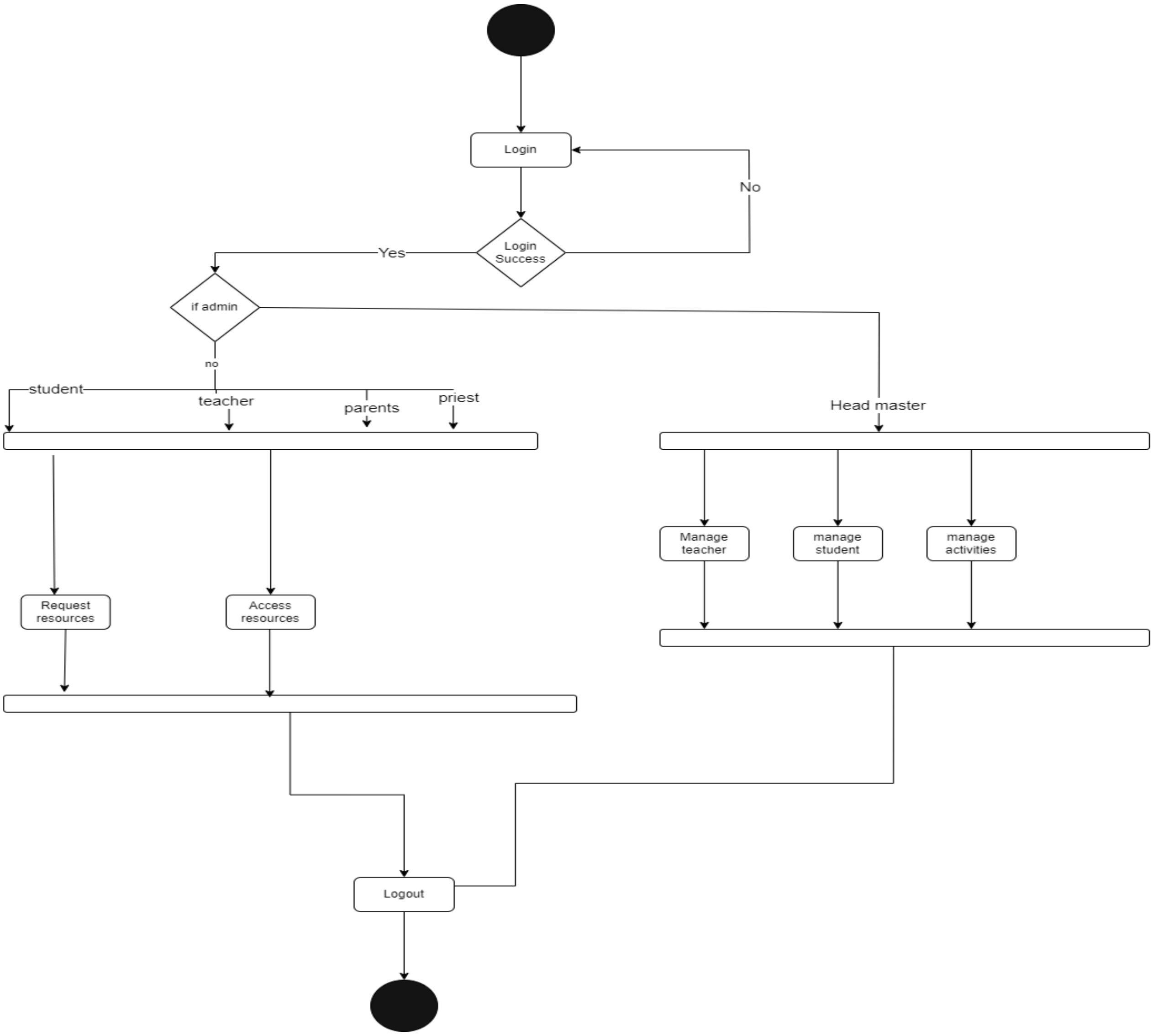


Fig.4 Activity diagram for Sunday school Management system

4.2.3 Class Diagram

Class diagram is a static diagram. It depicts the static view of an application.

Class diagram is not only used for visualizing, describing, and documenting distinct components of a system but also for building executable code of the software program.

Class diagram defines the features and activities of a class and the restrictions put on the system.

The class diagrams are commonly used in the design of object-oriented systems since they are the only UML diagrams that can be directly mapped with object-oriented languages.

Class diagram depicts a collection of classes, interfaces, connections, collaborations, and restrictions. It is also known as a structural diagram.

The objective of the class diagram may be described as −

When designing an application, the static view should be analyzed and designed. This view serves as the basis for component and deployment diagrams and helps describe the duties of a system. Additionally, both forward and reverse engineering can be utilized.

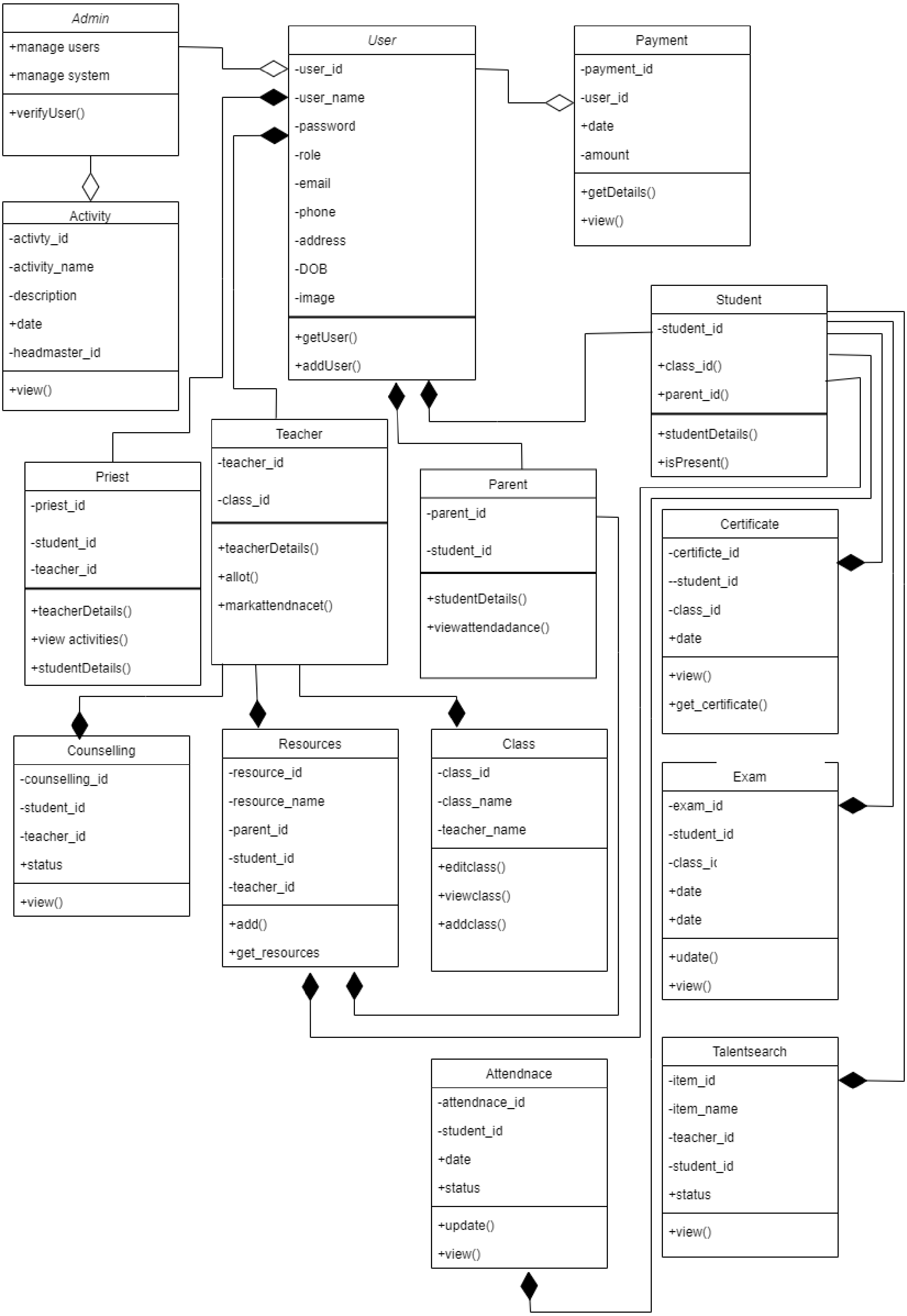


Fig.5 Class diagram for Sunday School Management system

4.2.4 Object Diagram

Object diagrams depict an instance of a class diagram. The core ideas are same for class diagrams and object diagrams. Object diagrams also show the static view of a system but this static view is a snapshot of the system at a given instant.

Object diagrams are used to portray a collection of objects and their connections as an instance.

Purpose of Object Diagrams

- The objective of a diagram should be understood well to apply it practically. The aims of object diagrams are similar to class diagrams.
- The distinction is that a class diagram provides an abstract model consisting of classes and their connections. However, an object diagram reflects an instance at a given instant, which is concrete in nature.
- It suggests the object diagram is closer to the real system behavior. The objective is to capture the static picture of a system at a certain point.

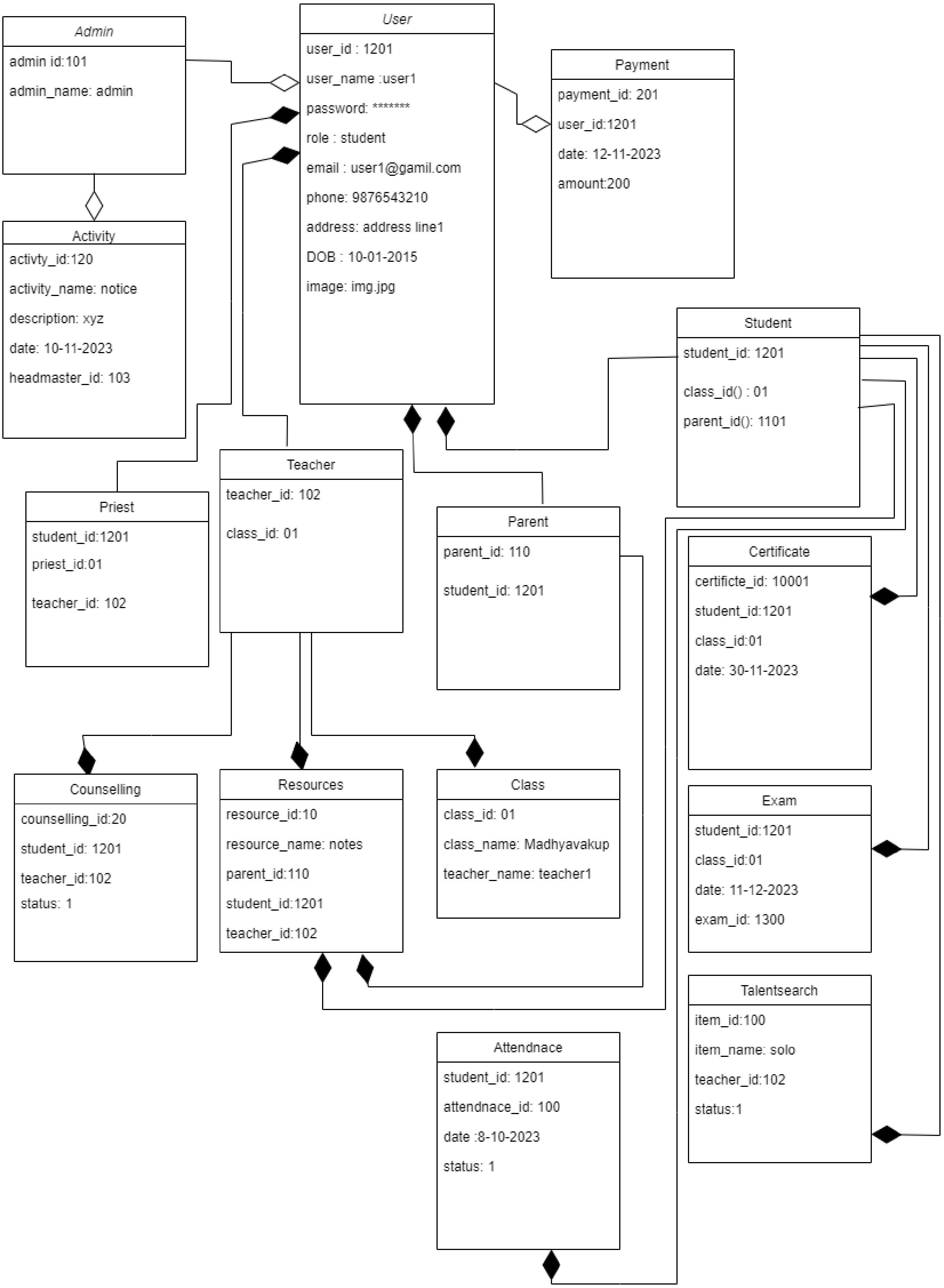


Fig.6 Object diagram for Sunday School Management System

4.2.5 Component Diagram

Component diagrams are varied in terms of nature and behavior. Component diagrams are used to represent the physical features of a system. Now the issue is, what are these physical aspects? Physical aspects are the items like as executables, libraries, files, documents, etc. which live in a node.

Component diagrams are used to depict the arrangement and connections among components in a system. These diagrams are also used to build executable systems.

Purpose of Component Diagrams

- Component diagram is a unique form of diagram in UML. The objective is also distinct from all previous diagrams described thus far.
- It does not define the functionality of the system rather it describes the components needed to produce such functions.
- Component diagrams may alternatively be regarded as a static implementation view of a system. Static implementation reflects the structure of the components at a certain instant.
- Files used in the system.
- Libraries and other artifacts related to the application.
- Relationships among the objects.

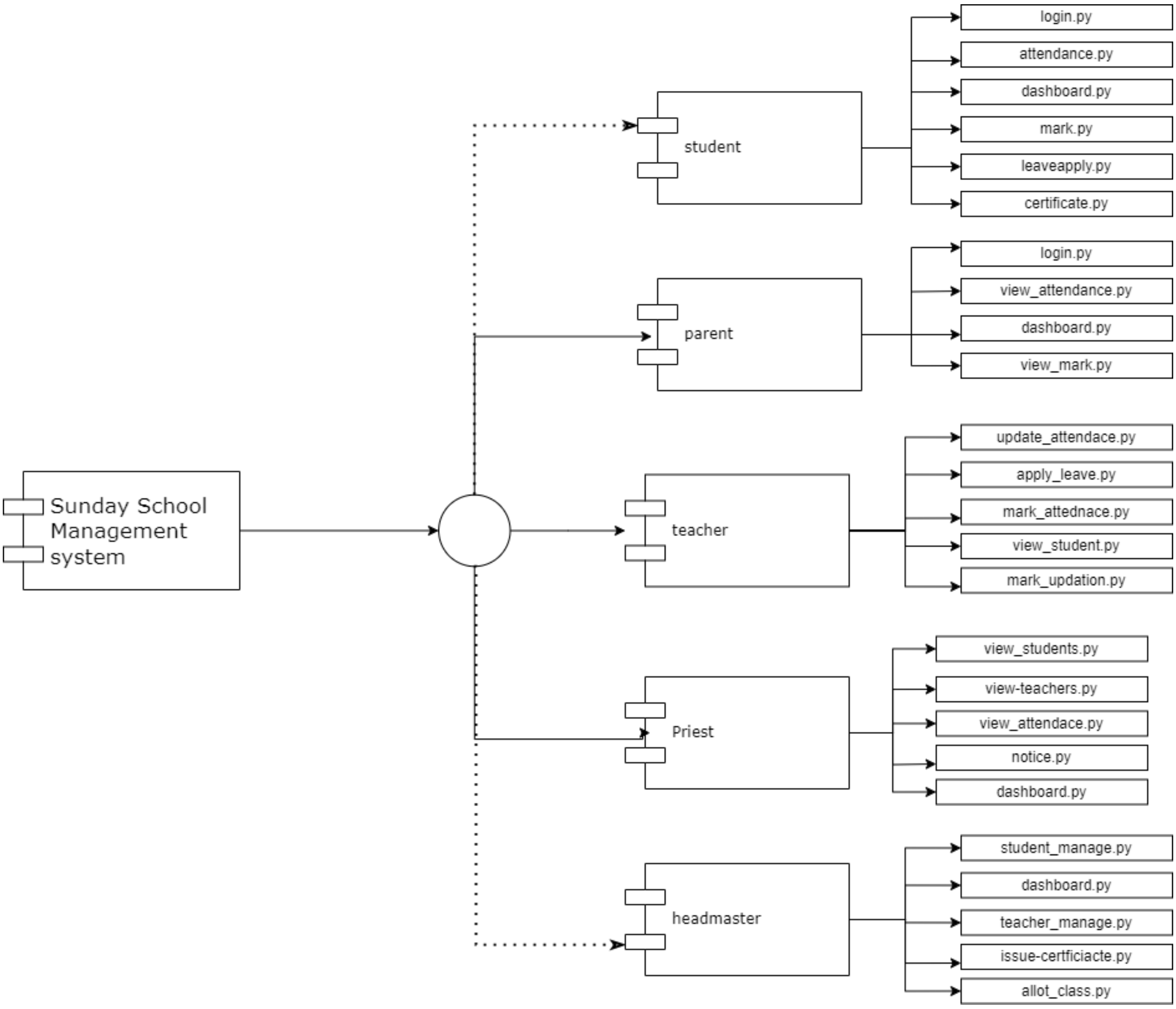


Fig.7 Component diagram for Sunday School Management System

4.2.8 Deployment Diagram

Deployment diagrams are used to depict the topology of the physical components of a system, where the software components are deployed.

Deployment diagrams are used to depict the static deployment perspective of a system. Deployment diagrams consist of nodes and their connections.

Purpose of Deployment Diagrams

- The name Deployment itself explains the aim of the graphic.
- Deployment diagrams are used for depicting the physical components, where software components are deployed.
- Visualize the hardware topology of a system.

- Describe the runtime processing nodes.
- Component graphs and arrangement graphs are closely associated.
- Depict the equipment components utilized to send program components.

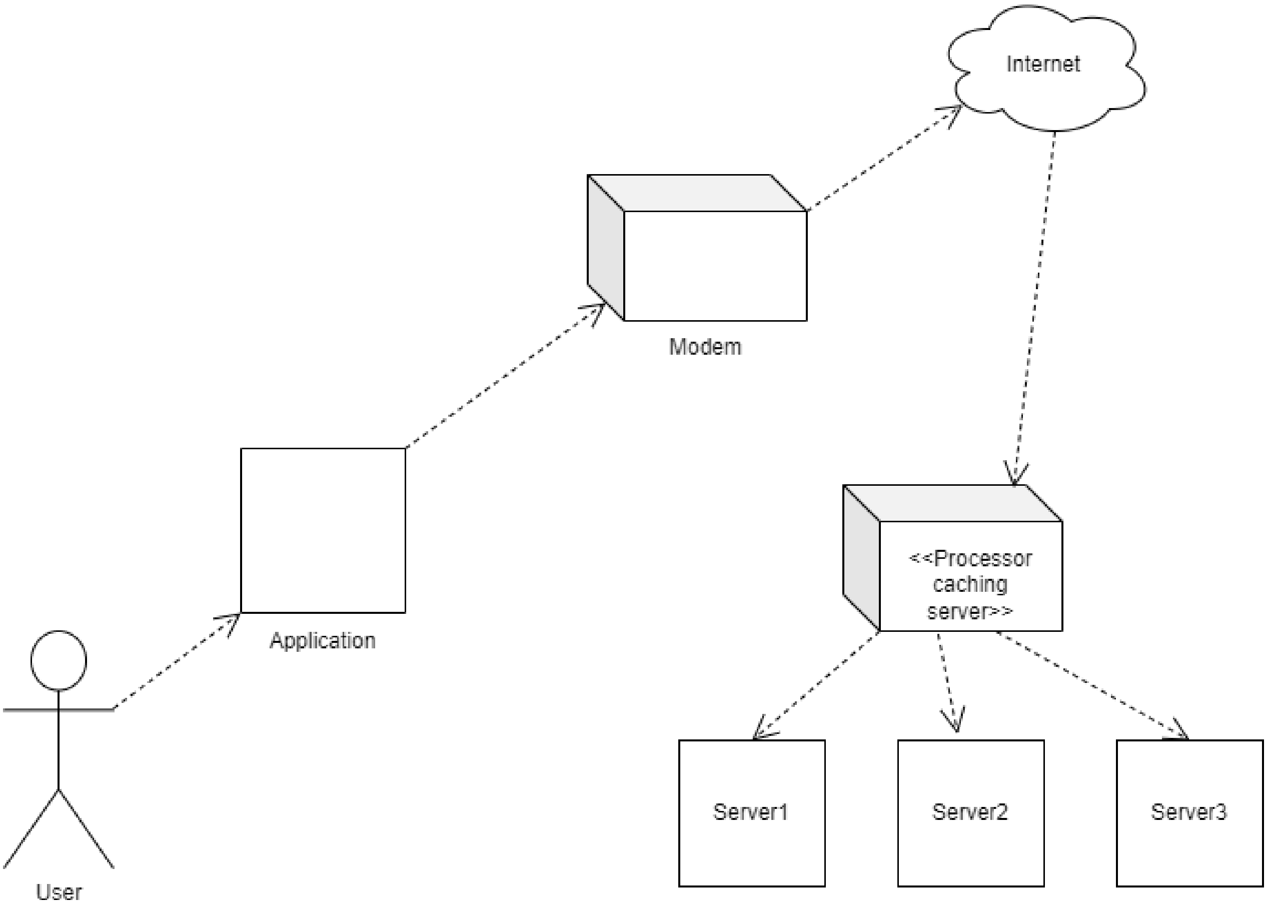


Fig.8 Deployment diagram for Sunday School Management System

4.2.9 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is a depiction of the links and interactions among software objects in the Unified Modeling Language (UML).

- Developers may use these diagrams to show the dynamic behavior of a given use case and specify the function of each component.

- **Objects.** These are displayed as rectangles with name labels within. The naming label follows the standard of object name: class name. If an item has a characteristic or condition that explicitly effects the cooperation, this should also be stated.

- **Actors.** These are examples that activate the interaction in the diagram. Each actor has a name and a role, with one actor beginning the whole use case.

- **Links.** These link things with actors and are portrayed by a solid line between two components. Each link is an instance where messages may be transmitted.

- **Messages between things.** These are represented as a labeled arrow placed near a link. These messages are exchanges between objects that transmit information about the action and might contain the sequence number.

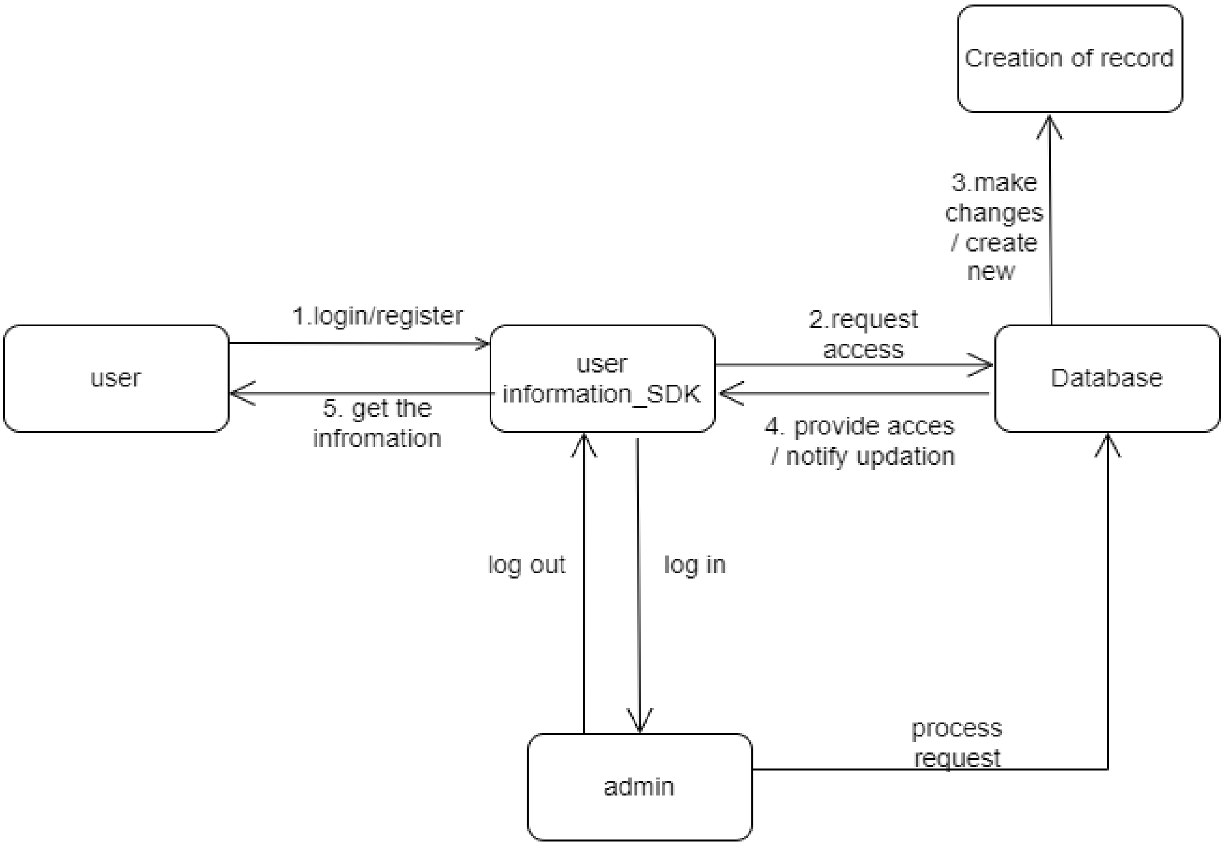
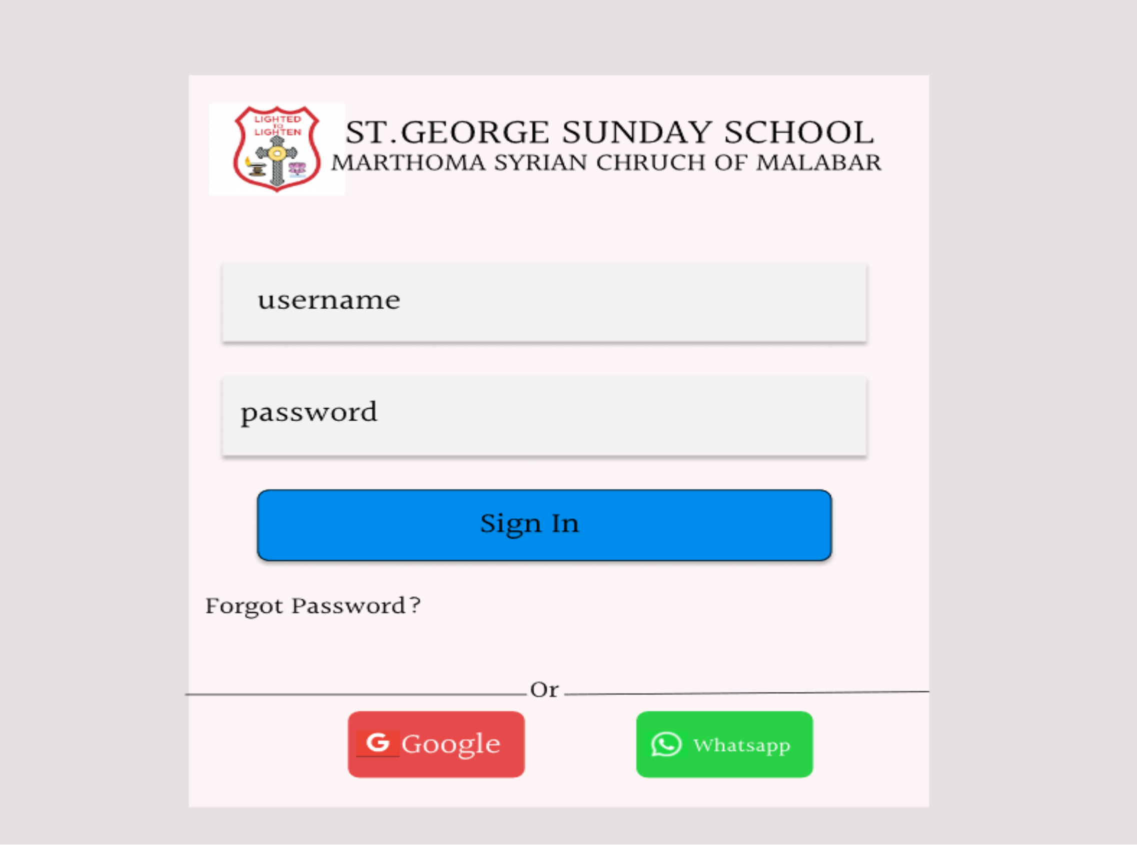


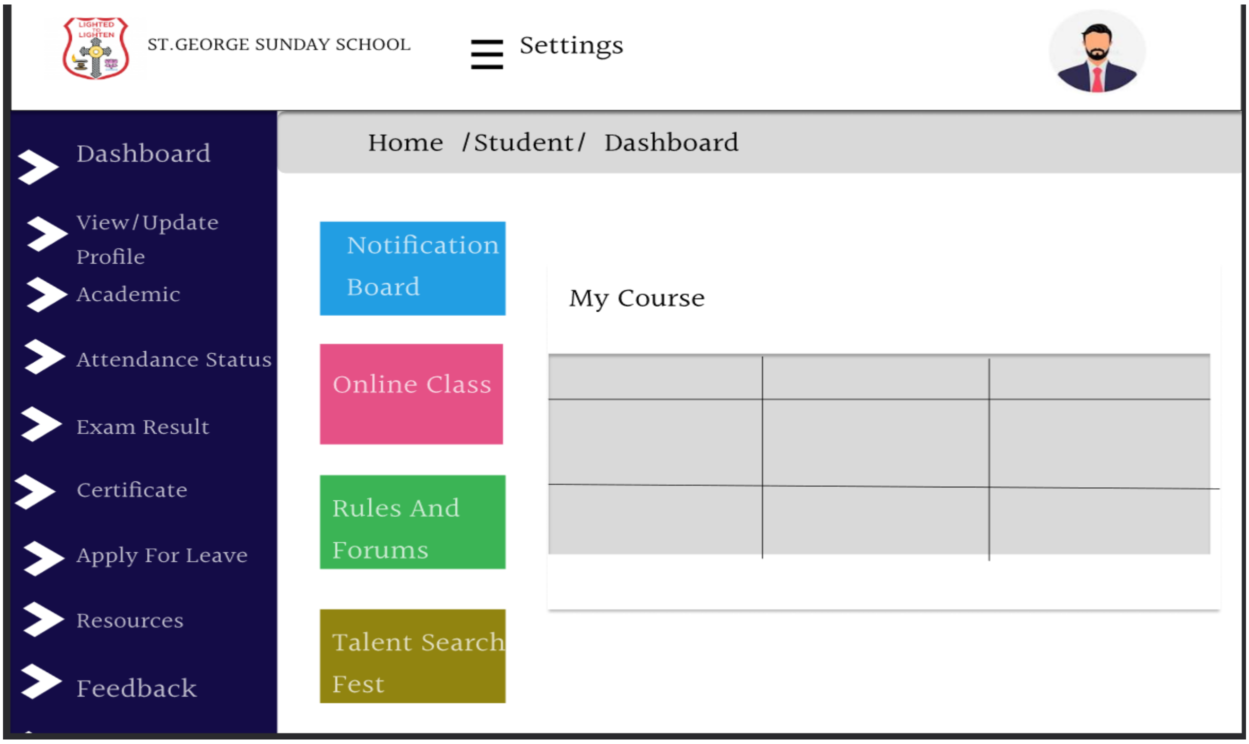
Fig.9 Collaboration diagram for Sunday School Management System

4.3 USER INTERFACE DESIGN USING FIGMA

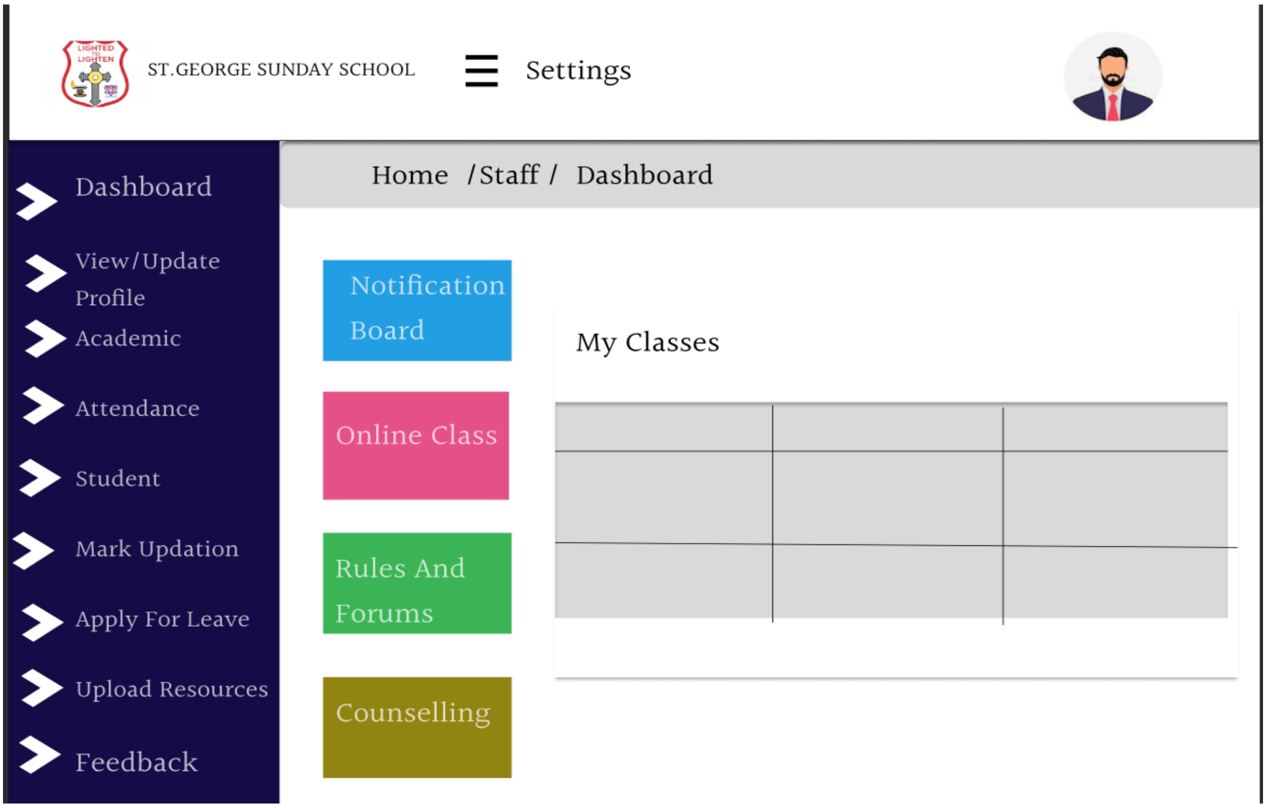
Form Name: login



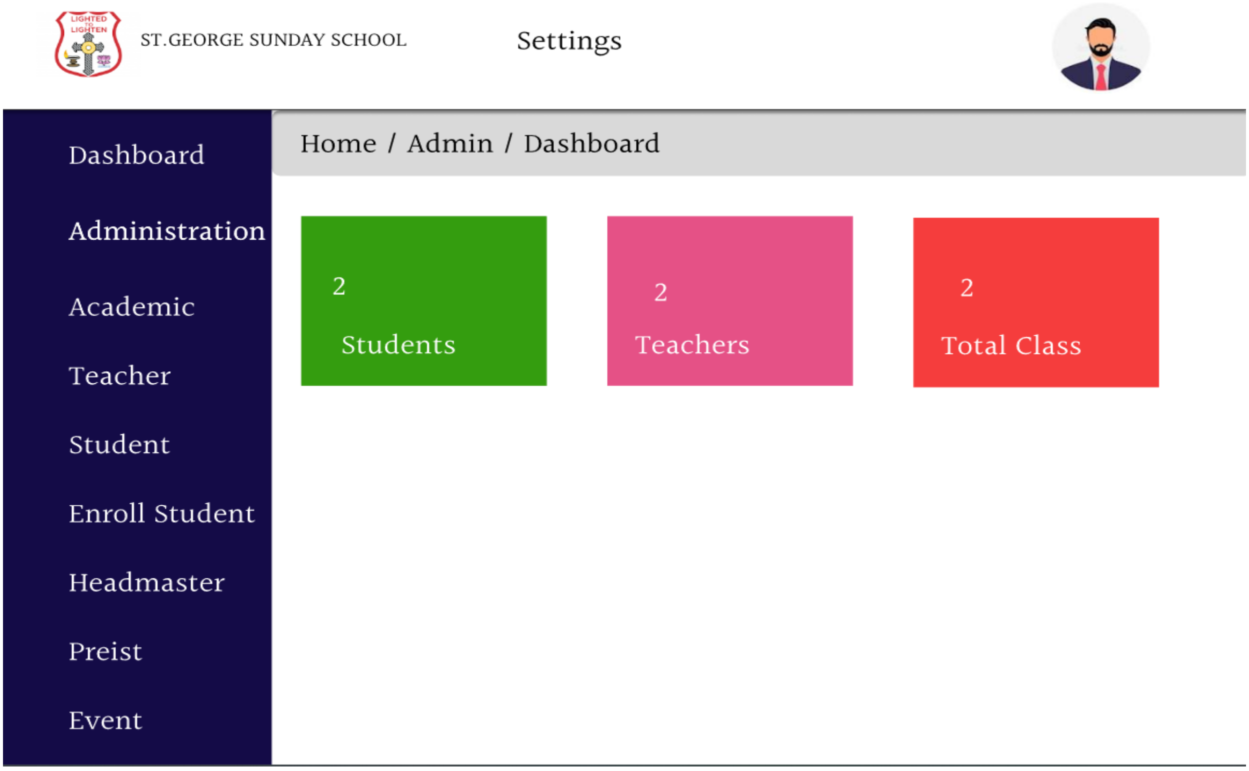
Form Name: student dashboard.



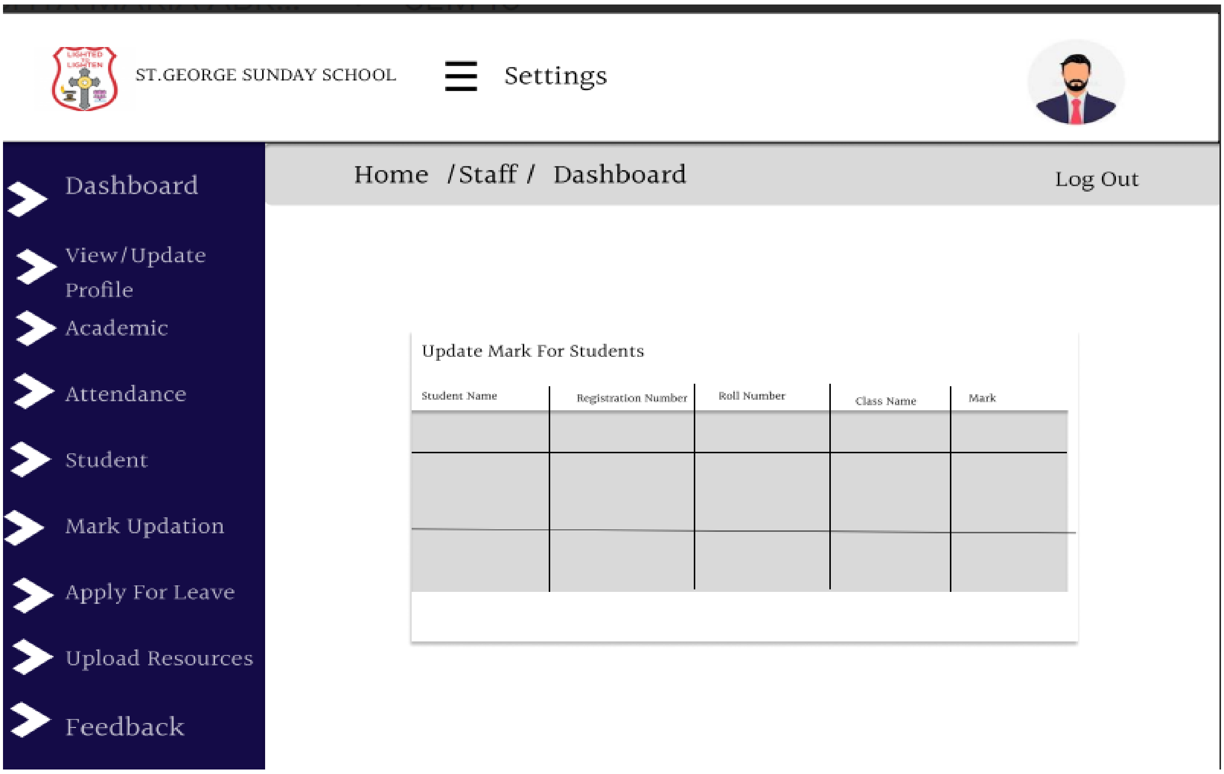
Form Name: teacher dashboard.



Form Name: priest dashboard.



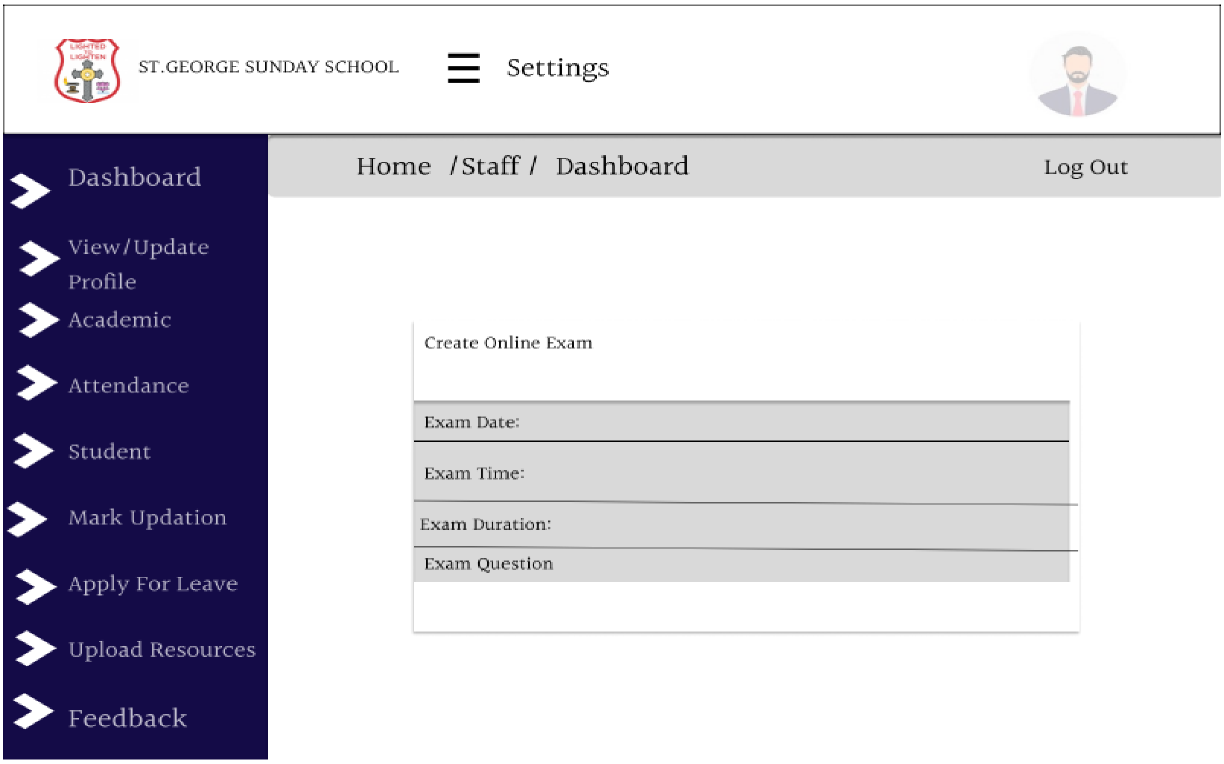
Form Name: Update student marks.



Update Mark For Students

Student Name	Registration Number	Roll Number	Class Name	Mark

Form Name: Create online exam.



Create Online Exam

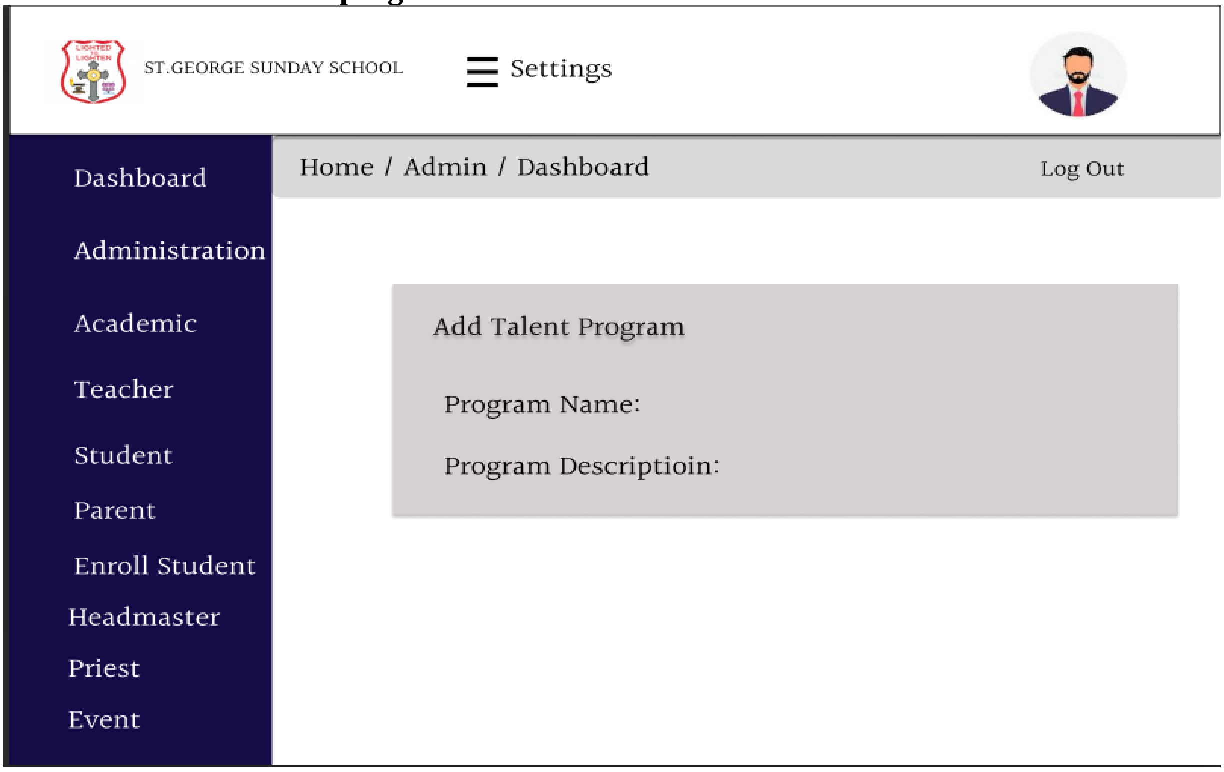
Exam Date:

Exam Time:

Exam Duration:

Exam Question

Form Name: Add talent programs



ST.GEORGE SUNDAY SCHOOL

☰ Settings

Dashboard

Administration

Academic

Teacher

Student

Parent

Enroll Student

Headmaster

Priest

Event

Home / Admin / Dashboard

Log Out

Add Talent Program

Program Name:

Program Descriptioin:

4.4 DATABASE DESIGN

4.4.1 Relational Database Management System (RDBMS)

A Relational Database Management System (RDBMS) is a type of database system that organizes and manages data in a structured manner using tables. Here's a brief summary of its main points:

Tables/Relations: Data in an RDBMS is organized into tables (also known as relations). Each table represents a particular entity type and holds records (rows) that represent instances of that entity. Each record consists of fields (columns) that store data values for attributes of that entity.

In a relational database, connections are already formed between tables to maintain the integrity of both referential and entity relationships. A domain D is a collection of atomic values, and a common way to define a domain is by picking a data type from which the domain's data values are derived.

It is useful to give the domain a name to make it simpler to comprehend the data it contains. Each Value in a relation is atomic and cannot be further split.

In a relational database, table associations are built via keys, with primary key and foreign key being the two most essential ones. Entity integrity and referential integrity connections can be built using these keys. Entity integrity assures that no main key may contain null values,

Whereas referential integrity assures that each different foreign key value must have a matched Primary key value in the same domain. Additionally, there are various sorts of keys such as super Keys and candidate keys.

4.4.2 Normalization

The simplest feasible grouping of data is utilized to bring them together so that future modifications may be made with minimal affect on the data structures. The formal process of normalizing data structures in a manner that eliminates duplication and supports integrity. Using the normalizing procedure, unnecessary fields are deleted and a massive table is broken into multiple smaller ones. Anomalies in insertion, deletion, and updating are also blocked by employing it. Keys and relationships are two concepts utilized in the typical style of data modeling. A row in a table is uniquely recognized by a key. Primary keys and foreign keys are two distinct sorts of keys. Primary key is an element, or collection of components, in a table that serves as a way of identifying between records from the same table. A column in a database called as a foreign key is used to uniquely identify entries from additional tables. Up to the third normal form, all tables have been normalized.

Normalization is a procedure in database architecture that seeks to arrange data into correct tables and columns, making it readily connected to the data by the user. This procedure removes data duplication that may be a drag on computer resources. The primary processes involved in normalizing

Include:

- Normalizing the data
- Choosing acceptable names for tables and columns
- Choosing the suitable names for the data

By following these methods, a developer may design a more efficient and structured database that is simpler to handle and maintain.

First Normal Form

The First Normal Form (1NF) demands that each attribute in a table must include only atomic or indivisible values. It prevents the usage of nested relations or relations inside relations as attribute values inside tulles. To meet 1NF, data must be separated into distinct tables when the data is of comparable type in each table, and each table should have a primary key or foreign key as needed by the project.

Example:

Let's consider a scenario where we have a table named **Student** that stores Student information. Initially, the table is not in First Normal Form (1NF) because some attributes contain multiple values.

Here is the initial table structure:

ID	**Name**	**Talent program**
1	Milan	Solo,group song
2	Alan	Elocution,Solo

In the initial table structure:

Talent programs column contains multiple data separated by commas.

To bring this table into 1NF, we need to separate the multivalued attributes into distinct tables.

First, let's create a table for Students:

ID	**Name**
1	Milan
2	Alan

Then, we create a table for **Talent Programs**:

ID	**Talent Program**
1	Solo
1	Group song
2	Elocution
2	Solo

Now, each attribute in the tables contains only atomic values, and the table structure meets the requirements of the First Normal Form (1NF). Each table has a primary key (**ID**) to uniquely identify each row, and foreign keys can be used to establish relationships between the tables if necessary. This separation allows for easier data management and querying.

Second Normal Form

The second normal form (2NF) is a rule in database normalization that stipulates that non-key attributes should not be functionally reliant on just the portion of the main key of a relation that has a composite main key. In other words, each non-key property should rely on the whole main key, not just a portion of it. To do this, we need to dissect the table and build new associations for each sub key along with their dependent properties. It is crucial to keep the connection with the original main key and any qualities that are entirely functionally dependent on it. A relation is in 2NF only if it meets all the 1NF requirements for the main key.

Student Table:

StudentID	**StudentName**	**Age**	**ClassID**	**ClassName**
101	John	10	1	Grade 5
102	Emily	9	2	Grade 4
103	Michael	11	1	Grade 5
104	Sarah	8	3	Grade 3

Class Table:

ClassID	**ClassName**	**TeacherID**	**TeacherName**
1	Grade 5	201	Mr. Smith
2	Grade 4	202	Mrs. Johnson
3	Grade 3	203	Ms. Brown

In this scenario:

- The **Student** table contains information about students attending Sunday school, including their ID, name, age, and the ID of the class they belong to.
- The **Class** table stores details about each class, including the class ID, name, and the ID and name of the teacher.
- Both tables have a primary key: **StudentID** for the **Student** table and **ClassID** for the **Class** table.
- The **Student** table has a foreign key (**ClassID**) referencing the **Class** table.

This design satisfies Second Normal Form (2NF) because:

1. There are no partial dependencies on the primary key in either table.
2. Each non-key attribute depends on the entire primary key. For example, in the **Student** table, **StudentName** and **Age** depend on **StudentID** (primary key), while **ClassName** depends on **ClassID** (primary key) in the **Class** table.

Third Normal Form

The third normal form (3NF) demands that a relation contains no non-key attribute that is functionally determined by another non-key attribute or group of non-key attributes. This indicates that there should be no transitive reliance on the main key. To accomplish 3NF, we deconstruct the relation and put up a new relation that has non-key qualities that functionally determine other non-key.
attributes. This helps reduce any dependencies that are not solely reliant on the main key. A relation is regarded as a relation in 3NF if it meets the requirements of 2NF and, additionally, the non-key attributes of the relation are not dependent on any other non-key property.

4.4.3 Sanitization

Data sanitization is the process of eliminating any unlawful characters or values from data. In web applications, cleaning user input is a typical operation to avoid security risks. PHP provides a built-in

filter extension that may be used to clean and verify many sorts of external input such as email addresses, URLs, IP addresses, and more. These filters are meant to make data sanitization easier and quicker. For example, the PHP filter extension includes a method that may delete all characters except letters, numerals, and some special characters (!#$%&’*+-=?_`{|}~@.[]), as defined by a flag.
Web applications frequently take external information from numerous sources, including user input from forms, cookies, web services data, server variables, and database query results. It is vital to sanitize all external input to guarantee that it is safe and does not include any dangerous code or data.

Students Table:

StudentID	**StudentName**	**Age**	**Gender**	**ClassID**	**TeacherID**
1	John Doe	10	Male	101	201
2	Jane Smith	11	Female	102	202
3	Michael Lee	10	Male	101	201

Classes Table:

ClassID	**ClassName**	**RoomNumber**	**TeacherID**
101	Grade 5	101A	201
102	Grade 6	102B	202

Teachers Table:

TeacherID	**TeacherName**	**Subject**
201	Mr. Johnson	Math
202	Mrs. Brown	English

Explanation:

- **Students Table**: Contains information about students attending classes. Each student is identified by a unique **StudentID**. Other attributes include **StudentName**, **Age**, **Gender**, **ClassID** (which refers to the class the student attends), and **TeacherID** (which indicates the teacher of the class).
- **Classes Table**: Stores details about classes offered in the Sunday school. Each class has a unique **ClassID**. Attributes include **ClassName** (name of the class), **RoomNumber** (location of the class), and **TeacherID** (the teacher assigned to the class).
- **Teachers Table**: Holds information about teachers. Each teacher is identified by a unique **TeacherID**. Attributes include **TeacherName** (name of the teacher) and **Subject** (the subject they teach).

4.4.4 Indexing

- An index is a database structure that boosts the speed of table operations. Indexes may be created on one or more columns to allow speedy lookups and efficient ordering of information.
- When constructing an index, it is necessary to examine which columns will be utilized in SQL queries and to establish one or more indexes on certain columns. In practice, indexes are a form of table that holds a main key or index field and a reference to each item in the real database. Indexes are invisible to users and are solely utilized by the database search engine to swiftly find entries. The CREATE INDEX statement is used to build indexes in tables.
- When tables include indexes, the INSERT and UPDATE commands take longer because the database needs to insert or update the index values as well. However, the SELECT statements.
- become quicker on such tables because the index helps the database to find records more quickly.

4.5 TABLE DESIGN

1. **Tbl_users_reg**

Primary key: **userid**

Foreign key: **loginid** references table **Tbl_personalinfo, Tbl_teacherpersonalinfo, Tbl_priestpersonalinfo, Tbl_gaudrianinfo**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	userid	Int (10)	PRIMARY KEY	Login Id
2	email	varchar (20)	NOT NULL, UNIQUE	Username of actors
3	password	varchar (20)	NOT NULL	Password
4	name	varchar (20)	NOT NULL	Name of actors
5	mobile	varchar (20)	NOT NULL	Contact number of actors
6	image	imagefield	NOT NULL	Profile picture
7	role	varchar (20)	NOT NULL	Role of actors
8	status	int (10)	NOT NULL	Status-active/blocked

2. **Tbl_personalinfo**

Primary key: **ID**

Foreign key: **id** references table **Tbl_gaurdianfo,Tbl_previous_academic_info, Tbl_previous_academic_certificate**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	Student_id
2	Blood group	Char(2)	NOT NULL	Blood group of student
3	dob	Date()	NOT NULL	Date of birth
4	Gender	Char(2)	NOT NULL	Gender of student
5	Previous_acdemic_info	Int(10)	FOREIGN KEY	Education info id
6	Previous_acdemic_certificate	Int(10)	FOREIGN KEY	Previous education certificate

3. Tbl_teacherpersonalinfo

Primary key: **ID**

Foreign key: **ID** references table **Tbl_education, Tbl_training, Tbl_experinence**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	teacher_id
2	Blood group	Char(2)	NOT NULL	Blood group of teacher
3	dob	Date()	NOT NULL	Date of birth
4	Gender	Char(2)	NOT NULL	Gender of student
5	mobile	varchar (20)	NOT NULL	Contact number of actors
6	Education	Int(10)	FOREIGN KEY	Education info id
7	Training	Int(10)	FOREIGN KEY	traing info id
8	Experience	Int(10)	FOREIGN KEY	Experience id
9	date	Date()	NOT NULL	Date of joining
10	Is_delete	char(2)	NOT NULL	Status

4. Tbl_designation

Primary key: **ID**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID for designation
2	Name	VARCHAR(20)	NOT NULL	Name of designation
3	Date	Date()	NOT NULL	Date of creation

5. Tbl_educationinfo

Primary key: **ID**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID for educationinfo
2	Name_of_exam	VARCHAR(20)	NOT NULL	Name of exam
3	Passing_year	Date()	NOT NULL	Date of passing exam

6. Tbl_traininginfo

Primary key: **ID**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID for traininginfo
2	Traing_name	VARCHAR(20)	NOT NULL	Name of training
3	year	Date()	NOT NULL	Passing year of training
4	Duration	Int(10)	NOT NULL	Time taken to complete training
5	Place	Char(50)	NOT NULL	Place of training

7. Tbl_experienceinfo

Primary key: **ID**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID for experienceinfo
2	Institute_name	VARCHAR(20)	NOT NULL	Name of institution attended
3	Trainer	Char(20)	NOT NULL	Name of trainer

8. Tbl_section

Primary key: **ID**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID for section
2	Name	VARCHAR(20)	NOT NULL	Name of section of class
3	Date	Date()	NOT NULL	Date created

9. Tbl_guideteacher

Primary key: **ID**

Foreign key: **ID** references table **Tbl_teacherpersonalinfo**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Name	VARCHAR(20)	NOT NULL	Name of teacher
3	date	Date()	NOT NULL	Date created

10. Tbl_session

Primary key: **ID**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Name	Int(10)	NOT NULL	year
3	Date	Date()	NOT NULL	Date created

11. Tbl_classinfo

Primary key: **ID**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Name	Char(20)	NOT NULL	Name of class
3	Date	Date()	NOT NULL	Date created
4	Display_name	Char(20)	NOT NULL	Displaying name of class
5	Is_deleted	Int(2)	NOT NULL	status

12. Tbl_classregistration

Primary key: **ID**

Foreign key: **ID** references table **Tbl_guideteacher, Tbl_session, Tbl_section, Tbl_classinfo,**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Name	Char(20)	NOT NULL	Name of class
3	Date	Date()	NOT NULL	Date created
4	Class_name	Char(15)	FOREGIN KEY	Displaying name of class
5	Section	Char(15)	FOREGIN KEY	Section of class
6	Session	Char(20)	FOREGIN KEY	Session of class
7	Guide_teacher	Char(20)	FOREGIN KEY	Teacher to assign in class

13. Tbl_resource

Primary key: **ID**

Foreign key: **ID** references table **Tbl_teacherpersonalinfo, Tbl_classinfo,Tbl_classregistration,**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Class_info	Char(20)	FOREGIN KEY	Name of class
3	Uploaded_date	Date()	NOT NULL	Date of uploading
4	Class_registration	Char(15)	FOREGIN KEY	Registered class details
5	Resource_title	Char(15)	NOT NULL	Title of resource
6	Resource_file	File()	NOT NULL	Type of file
7	Description	Char(20)	NOT NULL	Description of resource

14. Tbl_gaurdianinfo

Primary key: **ID**

Foreign key: **user_id** references table **Tbl_user_reg,**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Father_name	Char(20)	NOT NULL	Father name of student
3	Father_phone_no	Char(20)	NOT NULL	Father’s phone number
4	Mother_name	Char(20)	NOT NULL	Mother name of student
5	Mother_phone_no	Char(20)	NOT NULL	Mother’s phone number
6	User_id	Int(10)	FOREGIN KEY	User_id of user_reg table

15. Tbl_previousacdemicinfo

Primary key: **ID**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Institute_name	Char(20)	NOT NULL	Name of institution
3	Name_of_exam	Char(20)	NOT NULL	Exam attended
4	Board_roll	Char(20)	NOT NULL	Roll_no
5	Passing_year	Char(20)	NOT NULL	Year of passing exam

16. Tbl_previousacdemiccertificate

Primary key: **ID**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	marksheet	File()	NOT NULL	Mark sheet file to upload
3	Other_certificate	File()	NOT NULL	Certificate to upload

17. Tbl_acdemicinfo

Primary key: **ID**

Foreign key: **Class_info** references table **Tbl_class_info, Personal_info_ID** references table**Tbl_personalinfo, Guardianinfo_id** references table **Tbl_gaurdian_info, previousacdemicinfo_id** references table **Tbl_previousacdemicinfo_id, previousacdemiccertificate_id** references table **Tbl_previousacdemiccertificate**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Class_info	Char(20)	FOREIGN KEY	Class_info_id from class_info table
3	Registration_no	Int(10)	UNIQUE	Registration number of student
4	status	Char(5)	NOT NULL	Status of student
5	Personal_info_id	Int(10)	FOREIGN KEY	Personal info id
6	Guardianinfo_id	Int(10)	FOREIGN KEY	Parents details
7	previousacdemicinfo	Int(10)	FOREIGN KEY	Previous education details
8	previousacdemiccertificate	Int(10)	FOREIGN KEY	Previous education certificates

18. Tbl_enrolledstudent

Primary key: **ID**

Foreign key: **Class_name** references table **Tbl_class_info, student_id** references table **Tbl_acdemicinfo**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Class_name	Int(10)	FOREIGN KEY	Class name form class info
3	Student_id	Int(10)	FOREIGN KEY	Student_id from academicinfo
4	Roll	Int(20)	NOT NULL	Roll number of student
5	Date	Date()	NOT NULL	Date created

19. Tbl_attendance

Primary key: **ID**

Foreign key: **Class_info_id** references table **Tbl_class_info, Session year_id** references table **Tbl_session,**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Class_info_id	Int(10)	FOREIGN KEY	Class name form class info
3	Session year_id	Int(10)	FOREIGN KEY	session_id from session
4	Created_at	Date()	NOT NULL	Date created
5	Attendance_date	Date()	NOT NULL	Date of marking attendance

20. Tbl_attendacereport

Primary key: **ID**

Foreign key: **attendace_id** references table **Tbl_attendance, student_id** references table **Tbl_personalinfo**

No	Field name	Datatype (Size)	Key Constraints	Description of the field

:				
1	ID	Int(10)	PRIMARY KEY	ID
2	Attendance_id	Int(10)	FOREIGN KEY	Attendance form attendace table
3	Student_id	Int(10)	FOREIGN KEY	Student_id from academicinfo
4	status	Char(20)	NOT NULL	Status of student
5	Date	Date()	NOT NULL	Date created

21. Tbl_leavereportstudent

Primary key: **ID**

Foreign key: **student_id** references table **Tbl_ personalinfo**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Status	Char(3)	NOT NULL	Status of student
3	Student_id	Int(10)	FOREIGN KEY	Student_id from academicinfo
4	Leave_message	char(50)	NOT NULL	Message
5	Leave_date	Date()	NOT NULL	Date of leave

22. Tbl_notice

Primary key: **ID**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	message	char(50)	NOT NULL	Message
3	date	Date()	NOT NULL	Date of notice

23. Tbl_teachernotice

Primary key: **ID**

Foreign key: **created_by** references table **Tbl_ user_reg**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	message	char(50)	NOT NULL	Message
3	date	Date()	NOT NULL	Date of notice
4	Created_by	Int(10)	FOREGIN KEY	Id of head master

24. Tbl_markupdation

Primary key: **ID**

Foreign key: **student** references table **Tbl_ enrolledstudent**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Student	char(50)	FOREGIN KEY	Id of EnrolledStudent
3	Subject1	Int(100)	NOT NULL	Mark of student

25. Tbl_scheduleonlineclasses

Primary key: **ID**

Foreign key: **teacher** references table **Tbl_ guideteacher, classname** references table **Tbl_classregistration**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Teacher	char(50)	FOREGIN KEY	ID of guideteacher
3	Class name	char(50)	FOREGIN KEY	ID of classregistration
4	Date	Date()	NOT NULL	Date of scheduling class
5	Time	DateTime()	NOT NULL	Time of class
6	Link	Char(50)	NOT NULL	Link to attend class
7	Description	Char(100)	NOT NULL	Details of class

26. Tbl_certificate

Primary key: **ID**

Foreign key: **student** references table **Tbl_ enrolledstudent, class_associated** references table **Tbl_ classregistrations**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Student	Char(50)	FOREGIN KEY	ID of Enrolled Student
3	Class_associated	Char(50)	FOREGIN KEY	ID of class registartion
4	Certificate_file	Char(50)	NOT NULL	Download certificate
5	Issue_date	Date()	NOT NULL	Date of issuing certificate

27. Tbl_talentsearchregistration

Primary key: **ID**

Foreign key: **student** references table **Tbl_ enrolledstudent, student** references table **Tbl_ talent,**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Student	Char(50)	FOREGIN KEY	ID of Enrolled Student
3	Program	Char(50)	FOREGIN KEY	ID of talent
4	Registration_date	Date()	NOT NULL	Date of registration

28. Tbl_talentprogram

Primary key: **ID**

Foreign key:

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Name	Char(50)	NOT NULL	Name of program
3	Description	Char(50)	NOT NULL	Description of program

29. Tbl_examschedule

Primary key: **ID**

Foreign key: **hod** references table **Tbl_personal_info, class_name** references table **Tbl_classregistration,**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Hod	Char(50)	FOREGIN KEY	ID of hod
3	Class_name	Char(50)	FOREGIN KEY	ID of class
4	Subject	Char(50)	NOT NULL	Name of exam
5	Date	Date()	NOT NULL	Date of exam
6	Start time	Datetime()	NOT NULL	Time of exam
7	Duration	Int(10)	NOT NULL	Duration of exam
8	Is_published	Int(2)	NOT NULL	Boolean

30. Tbl_talentprogram

Primary key: **ID**

Foreign key:

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Name	Char(50)	NOT NULL	Name of program
3	Description	Char(50)	NOT NULL	Description of program

31. Tbl_progressreport

Primary key: **ID**

Foreign key:

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Student	Char(50)	FOREGIN KEY	ID of academic info
3	Academic -year	Date()	FOREGIN KEY	ID of session
4	Marks	Int(10)	NOT NULL	ID of mark
5	Remarks	Char(50)	NOT NULL	Remarks of student

32. Tbl_leavereportstaff

Primary key: **ID**

Foreign key: **staff_ID** references table **Tbl_ teacherpersonalinfo**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Status	Char(3)	NOT NULL	Status of staff
3	Staff_ID	Int(10)	FOREIGN KEY	Staff_ID from teacherpersonalinfo

1. Tbl_counsellingsessions

Primary key: **ID**

Foreign key: **student** references table **Tbl_ enrolledstudent**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ID	Int(10)	PRIMARY KEY	ID
2	Student	Char(50)	FOREGIN KEY	ID of EnrolledStudent
3	Teacher	Char(50)	FOREGIN KEY	ID of guideteacher
4	Chat_room_link	Char(50)	NOT NULL	Link to chat room
5	Date	Date()	NOT NULL	Date of conduct counselling

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

System testing tries to assess the system's end-to-end functionality, performance, and behavior under diverse circumstances rather than concentrating on specific units or components of the program. System testing verifies that the various modules or components inside the program interact harmoniously, and that the software operates properly in real-world circumstances. It is often carried out in an environment that closely resembles the production configuration. During this crucial step, potential problems, such as functionality gaps, interface flaws, or integration issues, are discovered and fixed, guaranteeing that the software product is of high quality and ready for deployment until the very end.

Software testing includes running a software program in a controlled way to see whether it acts as intended, generally utilizing verification and validation techniques. Validation entails reviewing a product to verify it meets with requirements, whereas verification might encompass reviews, analyses, inspections, and walkthroughs. Static analysis investigates the software's source code to discover faults, whereas dynamic analysis examines its behavior during runtime to acquire information including execution traces, timing profiles, and test coverage data.

Testing comprises a set of planned and methodical operations that start with individual modules and move to the integration of the full computer-based system. The aims of testing include finding flaws and faults in the program, ensuring that the software operates according to its specifications, and validating that it fulfills performance requirements. Testing may be conducted to check accuracy, implementation efficiency, and computational complexity.

A successful test is one that discovers an undetected mistake, and a good test case has a high possibility of discovering such faults. Testing is vital to attaining system testing goals and may entail numerous methodologies such as functional testing, performance testing, and security testing.

5.2 TESTPLAN

A test plan is a document that defines the needed procedures to fulfill different testing techniques. It gives instructions on the tasks that need to be conducted during testing. Software developers construct computer programs, documentation, and related data structures. They are responsible for testing each component of the software to verify it fits the intended goal. To overcome concerns with self-evaluation, an independent test group (ITG) is typically created.

Testing goals should be described in quantitative terms, such as mean time to failure, cost to detect and repair problems, remaining defect density or frequency of recurrence, and test work-hours per regression test.

The many levels of testing include:

- Unit testing
- Integration testing
- Data validation testing
- Output testing

5.2.1 Unit Testing

Unit testing is a software testing approach that focuses on evaluating individual components or modules of the program architecture. The aim of unit testing is to test the smallest unit of software design and guarantee that it functions as intended. Unit testing is often white-box centric, and numerous components may be evaluated concurrently. The component-level design description is used as a reference during testing to identify essential control routes and probable defects inside the module's perimeter.

During unit testing, the modular interface is checked to verify that data enters and departs the program unit under test appropriately. The local data structure is reviewed to guarantee that data temporarily stored preserves its integrity at each stage of an algorithm's execution. Boundary conditions are verified to guarantee that all statements in a module have been performed at least once, and all error handling routes are evaluated to confirm that the program can handle problems effectively.

Before any additional testing can take place, it is required to test data flow across a module interface. If data cannot enter and leave the system correctly, all other tests are meaningless. Another key responsibility during unit testing is the selective evaluation of execution routes to anticipate probable mistakes and verify that error handling channels are set up to redirect or stop activity when an error occurs. Finally, boundary testing is undertaken to confirm that the program runs appropriately at its bounds.

In the Sell-Soft System unit testing was done out by considering each module as an independent entity and submitting them to a range of test inputs. Any flaws with the underlying logic of the modules were resolved, and each module was tested and ran individually after development. Unused code was deleted, and it was checked that every module was functional and provided the required consequence.

5.2.2 Integration Testing

Integration testing is a systematic approach that involves creating the program structure while simultaneously conducting tests to identify interface issues. The objective is to construct a program structure based on the design that uses unit-tested components. The entire program is then tested. Correcting errors in integration testing can be challenging due to the size of the overall program, which makes it difficult to isolate the causes of the errors. As soon as one set of errors is fixed, new ones may arise, and the process may continue in an apparently endless cycle.

Once unit testing is complete for all modules in the system, they are integrated to check for any interface inconsistencies. Any discrepancies in program structures are resolved, and a unique program structure is developed.

5.2.3 Validation Testing or System Testing

The final stage of the testing process involves testing the entire software system as a whole, including all forms, code, modules, and class modules. This is commonly referred to as system testing or black box testing. The focus of black box testing is on testing the functional requirements of the software. A software engineer can use this approach to create input conditions that will fully test each program requirement. The main types of errors targeted by black box testing include incorrect or missing functions, interface errors, errors in data structure or external data access, performance errors, initialization errors, and termination errors.

5.2.4 Output Testing or User Acceptance Testing

User acceptance testing is conducted to verify that the system fulfills the business objectives and user demands. It is crucial to include the end users throughout the development phase to ensure that the software corresponds with their requirements and expectations. During user acceptability testing, the input and output screen designs are examined with various kinds of test data. The compilation of test data is crucial to guarantee complete testing of the system.

5.2.5 Automation Testing

Automation testing is a software testing technique that leverages specialized automated testing software tools to perform a collection of test cases. Its major objective is to verify that the software or equipment performs exactly as planned. Automation testing discovers faults, bugs, and other problems that may emerge during product development.

While certain forms of testing, such as functional or regression testing, may be conducted manually, there are significant advantages to automating the process. Automation testing may be conducted at any time of day and employs automated procedures to analyze the software. The findings are presented, and this information may be compared to earlier test runs. Automation developers often create code in programming languages such as C#, JavaScript, and Ruby.

5.2.6 Selenium Testing

Selenium is an open-source automated testing framework used to validate web applications across multiple browsers and platforms. Selenium enables for the production of test scripts in many programming languages such as Java, C#, and Python. Jason Huggins, a developer at Thought Works, invented Selenium in 2004 while working on a web application that needed regular testing. He designed a JavaScript tool named "JavaScriptTestRunner" to automate browser activities and boost testing productivity. Selenium has subsequently evolved and continues to be developed by a team of contributors.

In addition to Selenium, another prominent tool used for automated testing is Cucumber. Cucumber is an open-source software testing framework that facilitates behavior-driven development (BDD). It enables for the generation of executable specifications in a human-readable format called Gherkin. One of the benefits of utilizing Cucumber is its ability to bridge the gap between business stakeholders and technical teams. By employing a standard language, Cucumber fosters better communication and cooperation throughout the testing process. It provides a common knowledge of the requirements and helps ensuring that the generated software satisfies the desired business objectives.

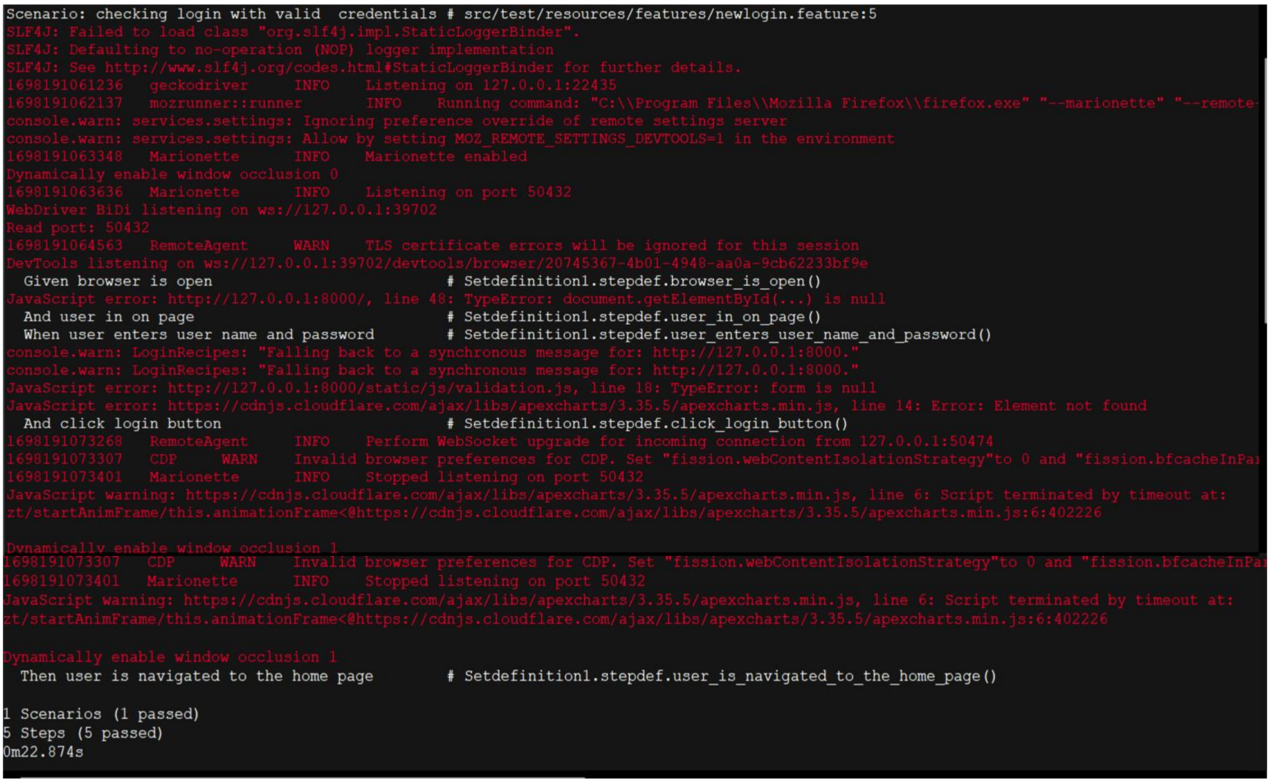
Example:

Test Case 1

Code

```
package Setdefinition1;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import io.cucumber.java.After;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
public class stepdef {
WebDriver driver=null;
@Given("browser is open")
public void browser_is_open() {
System.setProperty("webdriver.gecko.marionette","/A_3/src/test/resources/drivers/geckodriver-v0.32.2-win-aarch64/geckodriver.exe");
driver = new FirefoxDriver();
driver.manage().window().maximize();
}
@And("user in on page")
public void user_in_on_page() throws InterruptedException {
driver.navigate().to("http://127.0.0.1:8000/");
Thread.sleep(2000);
}
@When("user enters user name and password")
public void user_enters_user_name_and_password() {
driver.findElement(By.id("username")).sendKeys("admin@gmail.com");
driver.findElement(By.id("password")).sendKeys("123");
}
@And("click login button")
public void click_login_button() throws InterruptedException {
driver.findElement(By.id("submit")).click();
}
@Then("user is navigated to the home page")
public void user_is_navigated_to_the_home_page() throws InterruptedException {
driver.findElement(By.id("sidebar")).isDisplayed();
//driver.findElement(By.id("btn2")).isDisplayed();
driver.close();
Thread.sleep(2000);
}
}
```

Eg.Screenshot



```
Scenario: checking login with valid  credentials # src/test/resources/features/newlogin.feature:5
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
1698191061236   geckodriver     INFO    Listening on 127.0.0.1:22435
1698191062137   mozrunner::runner       INFO    Running command: "C:\\Program Files\\Mozilla Firefox\\firefox.exe" "--marionette" "--remote-
console.warn: services.settings: Ignoring preference override of remote settings server
console.warn: services.settings: Allow by setting MOZ_REMOTE_SETTINGS_DEVTOOLS=1 in the environment
1698191063348   Marionette      INFO    Marionette enabled
Dynamically enable window occlusion 0
1698191063636   Marionette      INFO    Listening on port 50432
WebDriver BiDi listening on ws://127.0.0.1:39702
Read port: 50432
1698191064563   RemoteAgent     WARN    TLS certificate errors will be ignored for this session
DevTools listening on ws://127.0.0.1:39702/devtools/browser/20745367-4b01-4948-aa0a-9cb62233bf9e
  Given browser is open                             # Setdefinition1.stepdef.browser_is_open()
JavaScript error: http://127.0.0.1:8000/, line 48: TypeError: document.getElementById(...) is null
  And user in on page                               # Setdefinition1.stepdef.user_in_on_page()
  When user enters user name and password           # Setdefinition1.stepdef.user_enters_user_name_and_password()
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
JavaScript error: http://127.0.0.1:8000/static/js/validation.js, line 18: TypeError: form is null
JavaScript error: https://cdnjs.cloudflare.com/ajax/libs/apexcharts/3.35.5/apexcharts.min.js, line 14: Error: Element not found
  And click login button                            # Setdefinition1.stepdef.click_login_button()
1698191073268   RemoteAgent     INFO    Perform WebSocket upgrade for incoming connection from 127.0.0.1:50474
1698191073307   CDP     WARN    Invalid browser preferences for CDP. Set "fission.webContentIsolationStrategy"to 0 and "fission.bfcacheInPa
1698191073401   Marionette      INFO    Stopped listening on port 50432
JavaScript warning: https://cdnjs.cloudflare.com/ajax/libs/apexcharts/3.35.5/apexcharts.min.js, line 6: Script terminated by timeout at:
zt/startAnimFrame/this.animationFrame<@https://cdnjs.cloudflare.com/ajax/libs/apexcharts/3.35.5/apexcharts.min.js:6:402226

Dynamically enable window occlusion 1
1698191073307   CDP     WARN    Invalid browser preferences for CDP. Set "fission.webContentIsolationStrategy"to 0 and "fission.bfcacheInPa
1698191073401   Marionette      INFO    Stopped listening on port 50432
JavaScript warning: https://cdnjs.cloudflare.com/ajax/libs/apexcharts/3.35.5/apexcharts.min.js, line 6: Script terminated by timeout at:
zt/startAnimFrame/this.animationFrame<@https://cdnjs.cloudflare.com/ajax/libs/apexcharts/3.35.5/apexcharts.min.js:6:402226

Dynamically enable window occlusion 1
  Then user is navigated to the home page           # Setdefinition1.stepdef.user_is_navigated_to_the_home_page()

1 Scenarios (1 passed)
5 Steps (5 passed)
0m22.874s
```

Test Report

Test Case 1

Project Name: Lighted To Lighten	
Login Test Case	
Test Case ID: Test_1	**Test Designed By: Adrishya Maria Abraham**
Test Priority(Low/Medium/High):High	**Test Designed Date: 20-03-2024**
Module Name: Login Interface	**Test Executed By: Lisha Varghese**
Test Title: User login	**Test Execution Date:23-03-2024**
Description: checking login with valid credentials	
Pre-Condition: User has valid username and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	browser is open		Browser is open with login interface	Browser is open with login interface	pass
2	user in on page and enters login id	Email: admin@gmail.com	User should be able to Login	User Logged in and navigated to User Dashboard	pass
3	enters password	password: admin			
4	Enter submit button				

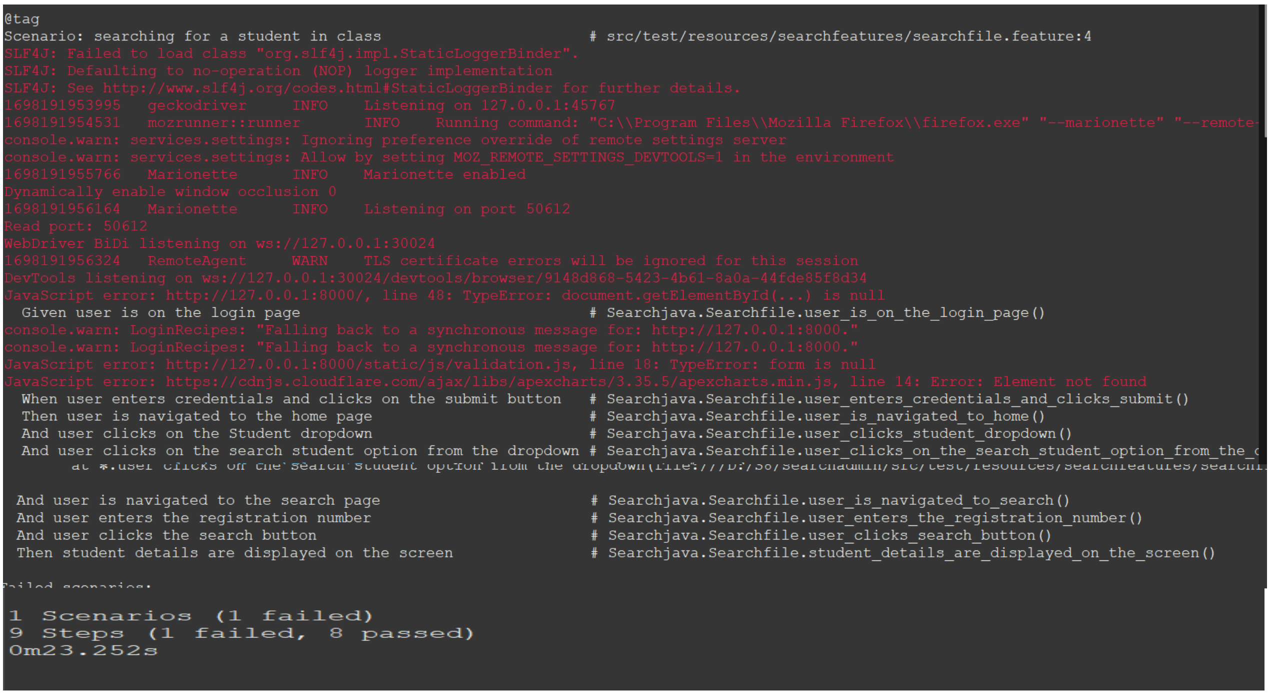
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database

Test Case 2:

Code:

```
package Searchjava;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class Searchfile {
WebDriver driver=new FirefoxDriver();
;

@Given("browser is open")
public void browser_is_open() {

System.setProperty("webdriver.gecko.marionette","/A_3/src/test/resources/drivers/gec
kodriver-v0.32.2-win-aarch64/geckodriver.exe");

driver = new FirefoxDriver();
driver.manage().window().maximize();
}

@Given("user is on the login page")
public void user_is_on_the_login_page() throws InterruptedException {
driver.navigate().to("http://127.0.0.1:8000/");
Thread.sleep(2000);
}


@When("user enters credentials and clicks on the submit button")
public void user_enters_credentials_and_clicks_submit() {
driver.findElement(By.id("username")).sendKeys("admin@gmail.com");
driver.findElement(By.id("password")).sendKeys("admin");
driver.findElement(By.id("submit")).click();
}

@Then("user is navigated to the home page")
public void user_is_navigated_to_home()throws InterruptedException
{
Thread.sleep(2000);
}
@And("user clicks on the Student dropdown")
public void user_clicks_student_dropdown() {
WebElement studentButton = driver.findElement(By.id("student"));
studentButton.click();
}
@And("user clicks on the search student option from the dropdown")
public void user_clicks_on_the_search_student_option_from_the_dropdown() {
WebElement searchOption = driver.findElement(By.id("search"));
searchOption.click();
}
@And("user is navigated to the search page")
public void user_is_navigated_to_search()throws InterruptedException{
Thread.sleep(2000);
}
@And("user enters the registration number")
public void user_enters_the_registration_number()throws InterruptedException{
WebElement registrations =
driver.findElement(By.cssSelector("input[name='registration_no']"));
registrations.sendKeys("294761");
Thread.sleep(2000);}
@And("user clicks the search button")
public void user_clicks_search_button() {
WebElement searchButton =
driver.findElement(By.xpath("//input[@type='submit']"));
searchButton.click();
}

@Then("student details are displayed on the screen")
public void student_details_are_displayed_on_the_screen()throws
InterruptedException{
WebElement settingsLink = driver.findElement(By.linkText("Settings"));

// Assert that the "Settings" link is displayed
if(!settingsLink.isDisplayed()) {
throw new AssertionError("Settings link is not displayed.");
}

Thread.sleep(2000);  // It's better to avoid using Thread.sleep. Consider using
WebDriverWait instead.
driver.close();
driver.quit();
}
}
```

Screenshot



```
@tag
Scenario: searching for a student in class                          # src/test/resources/searchfeatures/searchfile.feature:4
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
1698191953995   geckodriver     INFO    Listening on 127.0.0.1:45767
1698191954531   mozrunner::runner       INFO    Running command: "C:\\Program Files\\Mozilla Firefox\\firefox.exe" "--marionette" "--remote-
console.warn: services.settings: Ignoring preference override of remote settings server
console.warn: services.settings: Allow by setting MOZ_REMOTE_SETTINGS_DEVTOOLS=1 in the environment
1698191955766   Marionette      INFO    Marionette enabled
Dynamically enable window occlusion 0
1698191956164   Marionette      INFO    Listening on port 50612
Read port: 50612
WebDriver BiDi listening on ws://127.0.0.1:30024
1698191956324   RemoteAgent     WARN    TLS certificate errors will be ignored for this session
DevTools listening on ws://127.0.0.1:30024/devtools/browser/9148d868-5423-4b61-8a0a-44fde85f8d34
JavaScript error: http://127.0.0.1:8000/, line 48: TypeError: document.getElementById(...) is null
  Given user is on the login page                                   # Searchjava.Searchfile.user_is_on_the_login_page()
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
JavaScript error: http://127.0.0.1:8000/static/js/validation.js, line 18: TypeError: form is null
JavaScript error: https://cdnjs.cloudflare.com/ajax/libs/apexcharts/3.35.5/apexcharts.min.js, line 14: Error: Element not found
  When user enters credentials and clicks on the submit button   # Searchjava.Searchfile.user_enters_credentials_and_clicks_submit()
  Then user is navigated to the home page                           # Searchjava.Searchfile.user_is_navigated_to_home()
  And user clicks on the Student dropdown                           # Searchjava.Searchfile.user_clicks_student_dropdown()
  And user clicks on the search student option from the dropdown # Searchjava.Searchfile.user_clicks_on_the_search_student_option_from_the_c
        at ✱.user clicks on the search student option from the dropdown(file:///D:/38/searchadmin/src/test/resources/searchfeatures/searchfi

  And user is navigated to the search page                          # Searchjava.Searchfile.user_is_navigated_to_search()
  And user enters the registration number                           # Searchjava.Searchfile.user_enters_the_registration_number()
  And user clicks the search button                                 # Searchjava.Searchfile.user_clicks_search_button()
  Then student details are displayed on the screen                  # Searchjava.Searchfile.student_details_are_displayed_on_the_screen()

Failed scenarios:

1 Scenarios (1 failed)
9 Steps (1 failed, 8 passed)
0m23.252s
```

Test report

Test Case 2

Project Name: Lighted To Lighten	
Search Test Case	
Test Case ID: Test_2	**Test Designed By: Adrishya Maria Abrahan**
Test Priority (Low/Medium/High): Medium	**Test Designed Date:20-03-2024**
Module Name: Search student	**Test Executed By: Lisha Varghese**
Test Title: Search Student	**Test Execution Date:23-03-2024**
Description: searching for a student in class	

Pre-Condition: User has valid username and password

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	user is on the login page		Browser is open with login interface	Browser is open with login interface	pass
2	user in on page and enters login id	Email: admin@gmail.com	User should be able to Login	User Logged in and navigated to User Dashboard	pass
3	enters password	password: admin			
4	Enter submit button				
5	user is navigated to the home page and user clicks on the Student dropdown		Admin dashboard is viewed and click on search student from drop down	Dashboard is open and admin click on serach student option	pass
6	user is navigated to the search page user enters the registration number	Registration number: 360166	User is navigated to search page and when enters the number student details is displayed	User is navigated to search page and when enters the number student details is displayed	pass

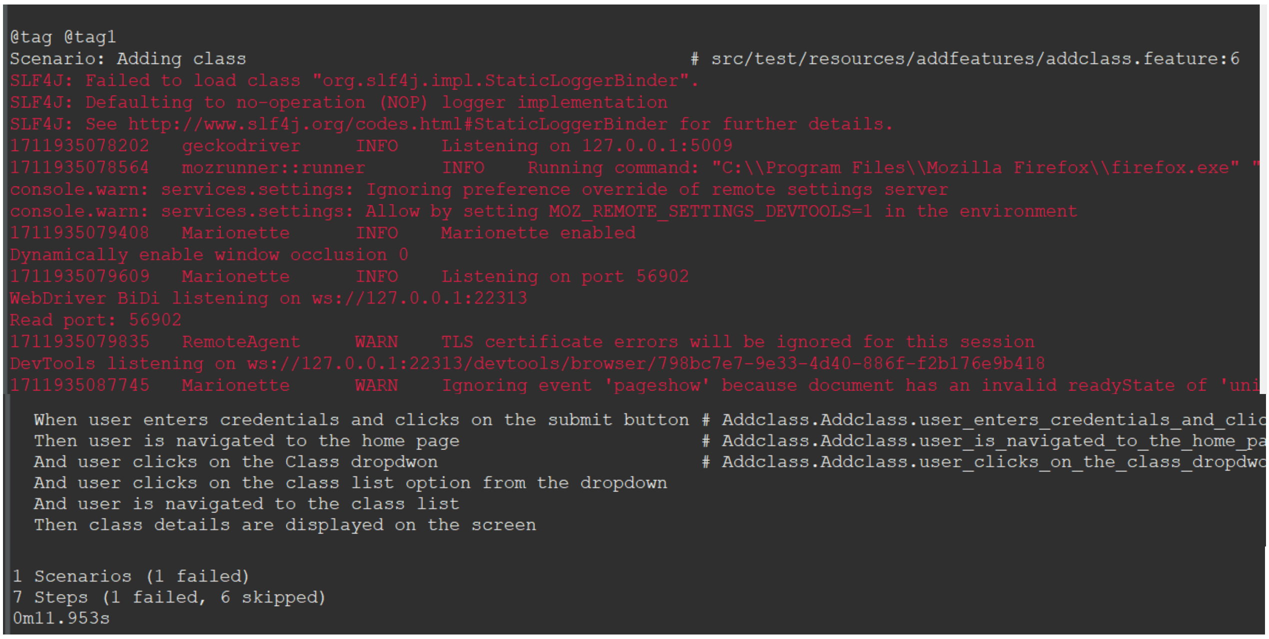
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database and when searching for a student, student details are displayed

Test Case 3

Code:

```
package Addclass;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
public class Addclass {

        WebDriver driver=new FirefoxDriver();

        @Given("browser is open")
        public void browser_is_open() {


System.setProperty("webdriver.gecko.marionette","/A_3/src/test/resources/drivers/geckodriver-
v0.32.2-win-aarch64/geckodriver.exe");

                driver = new FirefoxDriver();
                driver.manage().window().maximize();
        }
@Given("user is on the login page")
public void user_is_on_the_login_page() throws InterruptedException {

           driver.navigate().to("http://127.0.0.1:8000/");
                Thread.sleep(2000);
   // Write code here that turns the phrase above into concrete actions

}

@When("user enters credentials and clicks on the submit button")
public void user_enters_credentials_and_clicks_on_the_submit_button() {
   // Write code here that turns the phrase above into concrete actions
        driver.findElement(By.id("username")).sendKeys("admin@gmail.com");
        driver.findElement(By.id("password")).sendKeys("admin");
        driver.findElement(By.id("submit")).click();
}

@Then("user is navigated to the home page")
public void user_is_navigated_to_the_home_page() throws InterruptedException
{
         Thread.sleep(2000);
}

@And("user clicks on the Class dropdwon")
public void user_clicks_on_the_class_dropdwon() {
            WebElement studentButton = driver.findElement(By.id("class"));
    studentButton.click();
}

@Then("user clicks on the  class list option from the dropdown")
public void user_clicks_on_the_class_list_option_from_the_dropdown() {
            WebElement searchOption = driver.findElement(By.id("classlist"));
    searchOption.click();
}

@Then("user is navigated to the class list ")
public void user_is_navigated_to_the_class_list ()throws InterruptedException {
             Thread.sleep(2000);  // It's better to avoid using Thread.sleep. Consider using
WebDriverWait instead.
                driver.close();
                driver.quit();
}}
```

Screenshot



```
@tag @tag1
Scenario: Adding class                                           # src/test/resources/addfeatures/addclass.feature:6
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
1711935078202   geckodriver     INFO    Listening on 127.0.0.1:5009
1711935078564   mozrunner::runner       INFO    Running command: "C:\\Program Files\\Mozilla Firefox\\firefox.exe" "
console.warn: services.settings: Ignoring preference override of remote settings server
console.warn: services.settings: Allow by setting MOZ_REMOTE_SETTINGS_DEVTOOLS=1 in the environment
1711935079408   Marionette      INFO    Marionette enabled
Dynamically enable window occlusion 0
1711935079609   Marionette      INFO    Listening on port 56902
WebDriver BiDi listening on ws://127.0.0.1:22313
Read port: 56902
1711935079835   RemoteAgent     WARN    TLS certificate errors will be ignored for this session
DevTools listening on ws://127.0.0.1:22313/devtools/browser/798bc7e7-9e33-4d40-886f-f2b176e9b418
1711935087745   Marionette      WARN    Ignoring event 'pageshow' because document has an invalid readyState of 'uni
  When user enters credentials and clicks on the submit button # Addclass.Addclass.user_enters_credentials_and_clic
  Then user is navigated to the home page                      # Addclass.Addclass.user_is_navigated_to_the_home_pa
  And user clicks on the Class dropdwon                        # Addclass.Addclass.user_clicks_on_the_class_dropdwo
  And user clicks on the class list option from the dropdown
  And user is navigated to the class list
  Then class details are displayed on the screen

1 Scenarios (1 failed)
7 Steps (1 failed, 6 skipped)
0m11.953s
```

Test report

Test Case 3

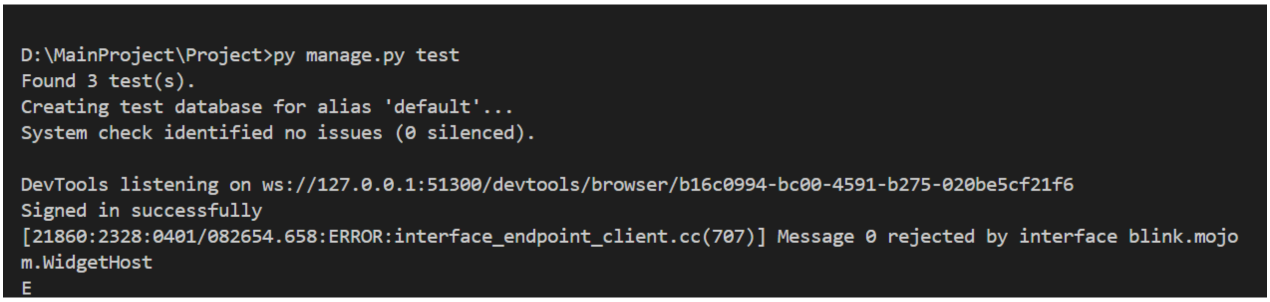
Project Name: Lighted To Lighten					
Add Class Test Case					
Test Case ID: Test_3			**Test Designed By: Adrishya Maria Abraham**		
Test Priority (Low/Medium/High): Medium			**Test Designed Date:20-03-2024**		
Module Name: Add class			**Test Executed By: Lisha Varghese**		
Test Title: Add class			**Test Execution Date:23-03-2024**		
Description: Add class by head master					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	user is on the login page		Browser is open with login interface	Browser is open with login interface	pass
2	user in on page and enters login id	Email: admin@gmail.com	User should be able to Login	User Logged in and navigated to User Dashboard	pass
3	enters password	password: admin			
4	Enter submit button				
5	user is navigated to the home page and user clicks on the class option		Admin dashboard is viewed and click on add class from drop down	Dashboard is open and admin click on add class option	pass
6	user is navigated to the search page user enters the class name	Class name:Class one	User is navigated to class page and when enters class name and details are displayed	User is navigated to class page and when enters class name and details are displayed	pass
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database and when searching for a student, student details are displayed					

Test Case 4

Code:

```
from django.test import TestCase
from datetime import datetime
from django.test import TestCase
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.alert import Alert
from selenium.webdriver.support import expected_conditions as EC
class Hosttest(TestCase):

    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.implicitly_wait(10)
        self.live_server_url = 'http://127.0.0.1:8000/'

    def tearDown(self):
        self.driver.quit()

    def test_01_login_page(self):
        driver = self.driver
        driver.get(self.live_server_url)
        driver.maximize_window()
        time.sleep(1)
        # Log in to the website
        username_input = driver.find_element(By.CSS_SELECTOR, '#username')
        username_input.send_keys("ishta@gmail.com")
        password_input = driver.find_element(By.CSS_SELECTOR, '#password')
        password_input.send_keys("STGEORGE")
        submit_button = driver.find_element(By.CSS_SELECTOR, '#submit')
        submit_button.click()
        time.sleep(2)
        print("Signed in successfully")

        # Navigate to the student leave list
        student_dropdown = driver.find_element(By.ID, 'student')
        student_dropdown.click()
        time.sleep(2)
        leave_list_link = driver.find_element(By.ID, 'leave_list')
        leave_list_link.click()
        time.sleep(2)
        print("Navigated to student leave list page")

        # Extract the student ID from the sidebar
                student_id_element = driver.find_element(By.CSS_SELECTOR, '.dropdown-
         content#student p')
        student_id = student_id_element.text
        print("Student ID:", student_id)
        # Example: Searching for student details using the student ID
        search_input = driver.find_element(By.CSS_SELECTOR, '#search_input')
        search_input.send_keys(student_id)
        search_button = driver.find_element(By.CSS_SELECTOR, '#search_button')
        search_button.click()
        time.sleep(2)

        # Example: Asserting that the student details are displayed
        student_details_element = driver.find_element(By.CSS_SELECTOR, '.student-details')
        assert student_id in student_details_element.text, "Student details not found"

        logout_button = driver.find_element(By.CSS_SELECTOR, ".logout-button")
        logout_button.click()
        time.sleep(2)
        print("Logged out successfully")
```

Screenshot



```
D:\MainProject\Project>py manage.py test
Found 3 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).

DevTools listening on ws://127.0.0.1:51300/devtools/browser/b16c0994-bc00-4591-b275-020be5cf21f6
Signed in successfully
[21860:2328:0401/082654.658:ERROR:interface_endpoint_client.cc(707)] Message 0 rejected by interface blink.mojo
m.WidgetHost
E
```

Test report

Test Case 4

Project Name: Lighted To Lighten

Student leave list Test Case

Test Case ID: Test_4	**Test Designed By: Adrishya Maria Abraham**
Test Priority (Low/Medium/High): Medium	**Test Designed Date:20-03-2024**
Module Name: student leave view	**Test Executed By: Lisha Varghese**
Test Title: student leave view	**Test Execution Date:23-03-2024**
Description: Teacher view students leave details	

Pre-Condition: User has valid username and password

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	user is on the login page		Browser is open with login interface	Browser is open with login interface	pass
2	user in on page and enters login id	Email: ishta@gmail.com	User should be able to Login	User Logged in and navigated to User Dashboard	pass
3	enters password	password: STGEORGE			
4	Enter submit button				
5	user is navigated to the home page and user clicks on student leave list		Teacher dashboard is viewed and click on student leave list	Dashboard is open and teacher click on student leave list	pass
6	user is navigated to the student leave list		User is navigated to student leave list page and details are displayed	User is navigated to student leave list page and details are displayed	pass

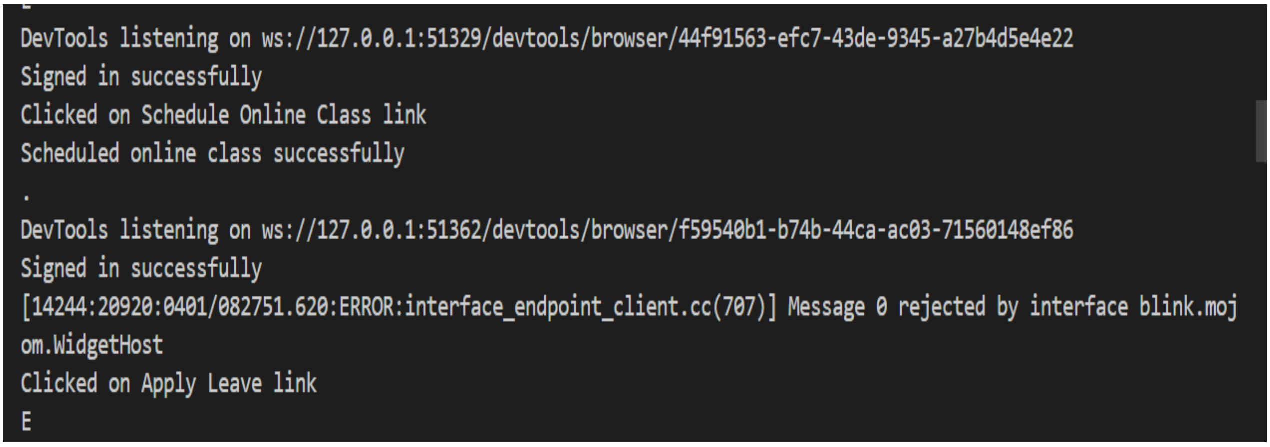
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database and when viewing student leave list, student details are displayed

Test Case 5

Code:

```
from django.test import TestCase
from datetime import datetime
from django.test import TestCase
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.alert import Alert
from selenium.webdriver.support import expected_conditions as EC
class Hosttest(TestCase):
    def test_03_schedule_online_class(self):
        driver = self.driver
        driver.get(self.live_server_url)
        driver.maximize_window()
        time.sleep(1)

        username_input = driver.find_element(By.CSS_SELECTOR, '#username')
        username_input.send_keys("ishta@gmail.com")
        password_input = driver.find_element(By.CSS_SELECTOR, '#password')
        password_input.send_keys("STGEORGE")
        submit_button = driver.find_element(By.CSS_SELECTOR, '#submit')
        submit_button.click()
        time.sleep(2)
        print("Signed in successfully")

        # Click on the link to schedule online class
        schedule_link = driver.find_element(By.ID, "onlineclass")
        schedule_link.click()
        time.sleep(2)
        print("Clicked on Schedule Online Class link")

        # Fill out the form to schedule the class
        enrolled_class_dropdown = driver.find_element(By.ID, 'selectclass')
        enrolled_class_dropdown.send_keys("Sisuvakup")  # Replace "Class Name" with the desired class name

        teacher_dropdown = driver.find_element(By.ID, 'teacher')
        teacher_dropdown.send_keys("Ishta")  # Replace "Ishta" with the desired teacher's name

        date_input = driver.find_element(By.ID, 'datePicker')
        date_input.send_keys("23-03-2024")  # Replace "2024-03-25" with the desired date

        time_input = driver.find_element(By.ID, 'timePicker')
        time_input.send_keys("16:00")  # Replace "10:00 AM" with the desired time
        platform_link_input = driver.find_element(By.ID, 'id')
        platform_link_input.send_keys("https://meet.google.com/mcv-mnxf-uan")  # Replace
"https://meet.google.com/mcv-mnxf-uan" with the platform link

        # Submit the form to schedule the class
        schedule_button = driver.find_element(By.ID, 'submit')
        schedule_button.click()
        time.sleep(2)
        print("Scheduled online class successfully")
```

Screenshot



```
DevTools listening on ws://127.0.0.1:51329/devtools/browser/44f91563-efc7-43de-9345-a27b4d5e4e22
Signed in successfully
Clicked on Schedule Online Class link
Scheduled online class successfully
.
DevTools listening on ws://127.0.0.1:51362/devtools/browser/f59540b1-b74b-44ca-ac03-71560148ef86
Signed in successfully
[14244:20920:0401/082751.620:ERROR:interface_endpoint_client.cc(707)] Message 0 rejected by interface blink.moj
om.WidgetHost
Clicked on Apply Leave link
E
```

Test report

Test Case 5

Project Name: Lighted To Lighten	
Schedule Online Class Test Case	
Test Case ID: Test_5	**Test Designed By: Adrishya Maria Abraham**
Test Priority (Low/Medium/High): Medium	**Test Designed Date:20-03-2024**
Module Name: schedule online class	**Test Executed By: Lisha Varghese**
Test Title: Schedule online class	**Test Execution Date:23-03-2024**
Description: Teacher can schedule online class for students	
Pre-Condition: User has valid username and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	user is on the login page		Browser is open with login interface	Browser is open with login interface	pass
2	user in on page and enters login id	Email: ishta@gmail.com	User should be able to Login	User Logged in and navigated to User Dashboard	pass
3	enters password	password:STGEROGE			
4	Enter submit button				
5	user is navigated to the home page and user clicks on the schedule online class option		teacher dashboard is viewed and click on schedule class	Dashboard is open and admin click on online class option	pass
6	user is navigated to the online class user enters the class name, teacher name, date, time, link	Class name:sisuvakup Teacher name: ishta Date: 5-04-24 Time: 10:00 am	User is navigated to online class page and details are added	User is navigated to online class page and details are added	pass

Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database and when adding online class, details are displayed.

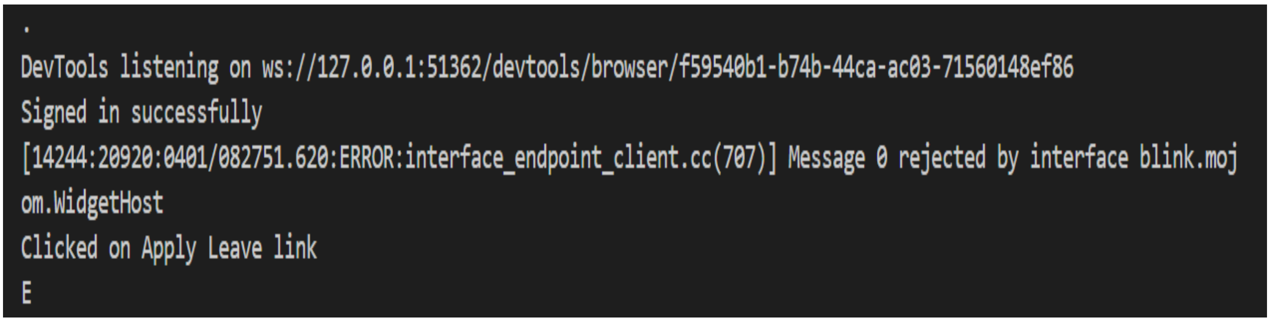
Test Case 6

Code:

```
from django.test import TestCase

# Create your tests here.
from datetime import datetime
from django.test import TestCase
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.alert import Alert
from selenium.webdriver.support import expected_conditions as EC
class Hosttest(TestCase):

    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.implicitly_wait(10)
        self.live_server_url = 'http://127.0.0.1:8000/'

    def tearDown(self):
        self.driver.quit()

    def test_01_login_page(self):
        driver = self.driver
        driver.get(self.live_server_url)
        driver.maximize_window()
        time.sleep(1)
        # Log in to the website
        username_input = driver.find_element(By.CSS_SELECTOR, '#username')
        username_input.send_keys("ishta@gmail.com")
        password_input = driver.find_element(By.CSS_SELECTOR, '#password')
        password_input.send_keys("STGEORGE")
        submit_button = driver.find_element(By.CSS_SELECTOR, '#submit')
        submit_button.click()
        time.sleep(2)
        print("Signed in successfully")

    def test_apply_leave(self):
        driver = self.driver
        driver.get(self.live_server_url)
        driver.maximize_window()
        time.sleep(1)

        # Log in to the teacher dashboard
        username_input = driver.find_element(By.CSS_SELECTOR, '#username')
        username_input.send_keys("ishta@gmail.com")
```

```
password_input = driver.find_element(By.CSS_SELECTOR, '#password')
password_input.send_keys("STGEORGE")
submit_button = driver.find_element(By.CSS_SELECTOR, '#submit')
submit_button.click()
time.sleep(2)
print("Signed in successfully")

# Navigate to the apply leave page
apply_leave_link = driver.find_element(By.XPATH, "//a[contains(text(),'Apply Leave')]")
apply_leave_link.click()
time.sleep(2)
print("Clicked on Apply Leave link")

# Fill out the leave application form

start_date_input = driver.find_element(By.CSS_SELECTOR, '#id_leave_date')
start_date_input.send_keys("17-03-2024")  # Replace with the desired start date


reason_input = driver.find_element(By.CSS_SELECTOR, '#id_leave_message')
reason_input.send_keys("Feeling unwell")  # Replace with the reason for leave

# Submit the leave application form
submit_button = driver.find_element(By.ID, 'submit')
submit_button.click()
time.sleep(2)
print("Leave applied successfully")
```

Screenshot



```
.
DevTools listening on ws://127.0.0.1:51362/devtools/browser/f59540b1-b74b-44ca-ac03-71560148ef86
Signed in successfully
[14244:20920:0401/082751.620:ERROR:interface_endpoint_client.cc(707)] Message 0 rejected by interface blink.moj
om.WidgetHost
Clicked on Apply Leave link
E
```

Test report

Test Case 6

Project Name: Lighted To Lighten	
Apply Leave Test Case	
Test Case ID: Test_6	**Test Designed By: Adrishya Maria Abraham**
Test Priority (Low/Medium/High): Medium	**Test Designed Date:20-03-2024**
Module Name: apply leave	**Test Executed By: Lisha Varghese**
Test Title: apply leave	**Test Execution Date:23-03-2024**
Description: teacher apply leave	
Pre-Condition: User has valid username and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	user is on the login page		Browser is open with login interface	Browser is open with login interface	pass
2	user in on page and enters login id	Email: ishta@gmail.com	User should be able to Login	User Logged in and navigated to User Dashboard	pass
3	enters password	password: STGEORGE			
4	Enter submit button				
5	user is navigated to the home page and user clicks on attendance		Teacher dashboard is viewed and click on attendance and select apply leave from dropdown	Teacher dashboard is viewed and click on attendance and select apply leave from dropdown	pass
6	user is navigated to the apply leave and enter leave date, leave message	Leave date:06-05-24 Leave message: headache	User is navigated to apply leave page and details are entered.	User is navigated to apply leave page and details are entered.	pass

Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database and leave is applied successfully by the user.

6.1INTRODUCTION

The implementation phase of a project is when the design is translated into a working system. It is a key step in guaranteeing the success of the new system, since it entails earning user trust that the system will perform successfully and precisely. User training and documentation are significant priorities during this period. Conversion may occur simultaneously with user training or at a later point. Implementation entails the translation of a freshly updated system design into an operating system.

During this stage, the user department bears the principal burden, undergoes the most severe upheaval, and has the most profound influence on the current system. Poorly planned or regulated implementation may produce confusion and turmoil. Whether the new system is wholly new, replaces an old manual or automated system, or alters an existing system, appropriate implementation is vital to satisfy the organization's goals. System implementation covers all operations necessary to convert from the old to the new system. The system can only be installed if rigorous testing is done and proven to be operating according to requirements. System professionals analyze the viability of the system. Implementation takes substantial work in three primary areas: education and training, system testing, and changeover. The implementation phase comprises meticulous planning, examining system and limitations, and creating techniques to accomplish changeover.

6.2 IMPLEMENTATION PROCEDURES

Software implementation is the process of deploying the program in its real environment and verifying that it fulfills the intended usage and performs as anticipated. In certain companies, the software development project may be commissioned by someone who will not be utilizing the program themself. During the beginning phases, there may be reservations regarding the program, but it's crucial to guarantee that opposition does not build up. This may be done by:

Ensuring that active users are aware of the advantages of the new system, boosting their trust in the program.

Providing sufficient instructions to the users so that they feel comfortable using the program. Before seeing the system, users need know that the server software must be operating on the server. Without the server object up and running, the planned procedure will not take place.

6.2.1 User Training

User training is aimed to prepare the user for testing and converting the system. To achieve the aim and advantages anticipated from computer-based system, it is vital for the people who will be participating to be sure of their position in the new system. As system grows more sophisticated, the necessity for training is increasingly critical. By user training the user begins to know how to input data, reply to error alerts, probe the database and call up routine that will make reports and do other essential duties.

6.2.2 Training on the Application Software

After delivering the required fundamental training on computer awareness, it is crucial to offer training on the new application software to the user. This training should include the underlying philosophy of using the new system, such as the flow of screens, screen design, the type of help available on the screen, the types of errors that may occur while entering data, and the corresponding validation checks for each entry, and ways to correct the data entered. Additionally, the training should contain knowledge particular to the user or group, which is important to utilize the system or portion of the system efficiently. It is vital to remember that this training may change across various user groups and levels of organization.

6.2.3 System Maintenance

The maintenance phase is a critical component of the software development cycle, since it is the period when the program is really put to use and fulfills its intended duties. Proper maintenance is vital to guarantee that the system stays functional, dependable, and responsive to changes in the system environment. Maintenance operations go beyond merely discovering and repairing problems or defects in the system. It may entail upgrades to the program, improvements to its features, and enhancements to its performance, among other things. In essence, software maintenance is an ongoing process that includes regular monitoring, review, and enhancement of the system to satisfy changing user demands and expectations.

6.2.4 Hosting

The project "Lighted to Lighten" is hosted on Amazon Web Services (AWS) Elastic Compute Cloud (EC2) platform. This decision was made after careful consideration of various hosting providers due to the robust and scalable nature of AWS services. By choosing AWS EC2, the project benefits from a reliable infrastructure that allows for easy scaling of compute resources based on demand. Additionally, AWS offers a wide range of features and services that cater to different needs, such as security, storage, database management, and networking. The flexibility of EC2 instances enables the project to efficiently handle varying workloads, ensuring optimal performance and resource utilization. Furthermore, AWS provides a global network of data centers, ensuring low latency and high availability for users across different geographical regions. Overall, hosting "Lighted to Lighten" on AWS EC2 aligns with the project's requirements for scalability, reliability, and performance, offering a competitive advantage in delivering a seamless and responsive user experience.

6.2.5 AWS EC2

Amazon Elastic Compute Cloud (EC2) is a web service offered by Amazon Web Services (AWS) that provides resizable compute capacity in the cloud. It allows users to easily rent virtual servers, known as instances, to run their applications and workloads. EC2 offers a wide range of instance types with varying CPU, memory, storage, and networking capabilities to meet different workload requirements. Users can choose from pre-configured Amazon Machine Images (AMIs) or create their own custom AMIs to launch instances with specific operating systems and software configurations.

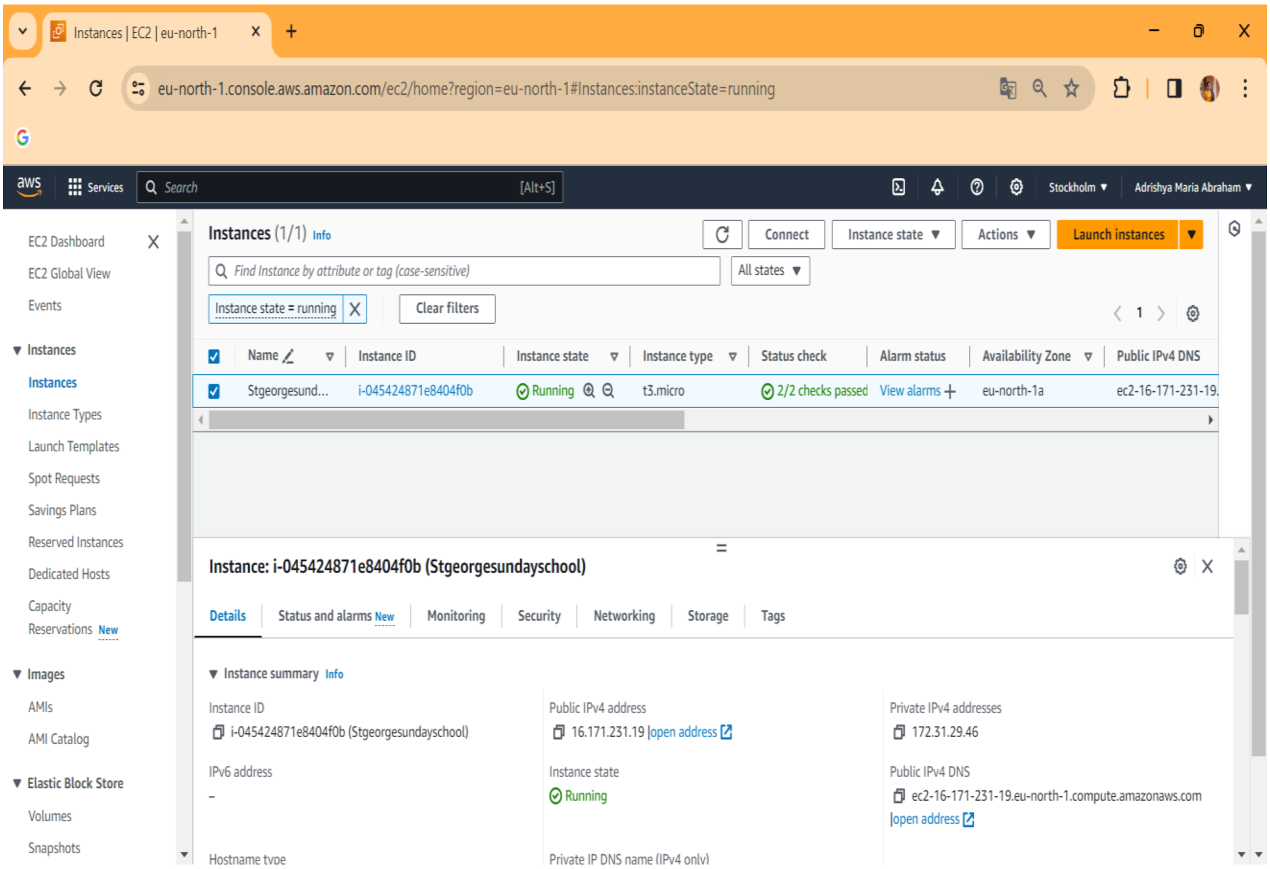
Users can quickly scale their compute capacity up or down based on demand, allowing them to handle fluctuating workloads efficiently. EC2 also provides features such as auto-scaling, which automatically adjusts the number of instances based on predefined scaling policies.

Another advantage of EC2 is its reliability and availability. AWS operates data centers in multiple geographic regions, allowing users to deploy instances closer to their end-users for reduced latency and improved performance. Additionally, EC2 instances are built on top of AWS's highly resilient infrastructure, which includes redundant components and data replication to ensure high availability and data durability.

EC2 offers various pricing options, including on-demand, reserved, and spot instances, allowing users to choose the most cost-effective pricing model based on their usage patterns and budget requirements. Overall, Amazon EC2 provides a flexible, reliable, and scalable cloud computing platform that empowers users to build and deploy a wide range of applications and services with ease.

Procedure for hosting website on AWS EC2

Hosted website:



Instances | EC2 | eu-north-1

eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#Instances:instanceState=running

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☑	Stgeorgesund...	i-045424871e8404f0b	Running	t3.micro	2/2 checks passed	View alarms +	eu-north-1a	ec2-16-171-231-19.

Instance: i-045424871e8404f0b (Stgeorgesundayschool)

Instance ID: i-045424871e8404f0b (Stgeorgesundayschool)

Public IPv4 address: 16.171.231.19 |open address

Private IPv4 addresses: 172.31.29.46

IPv6 address: –

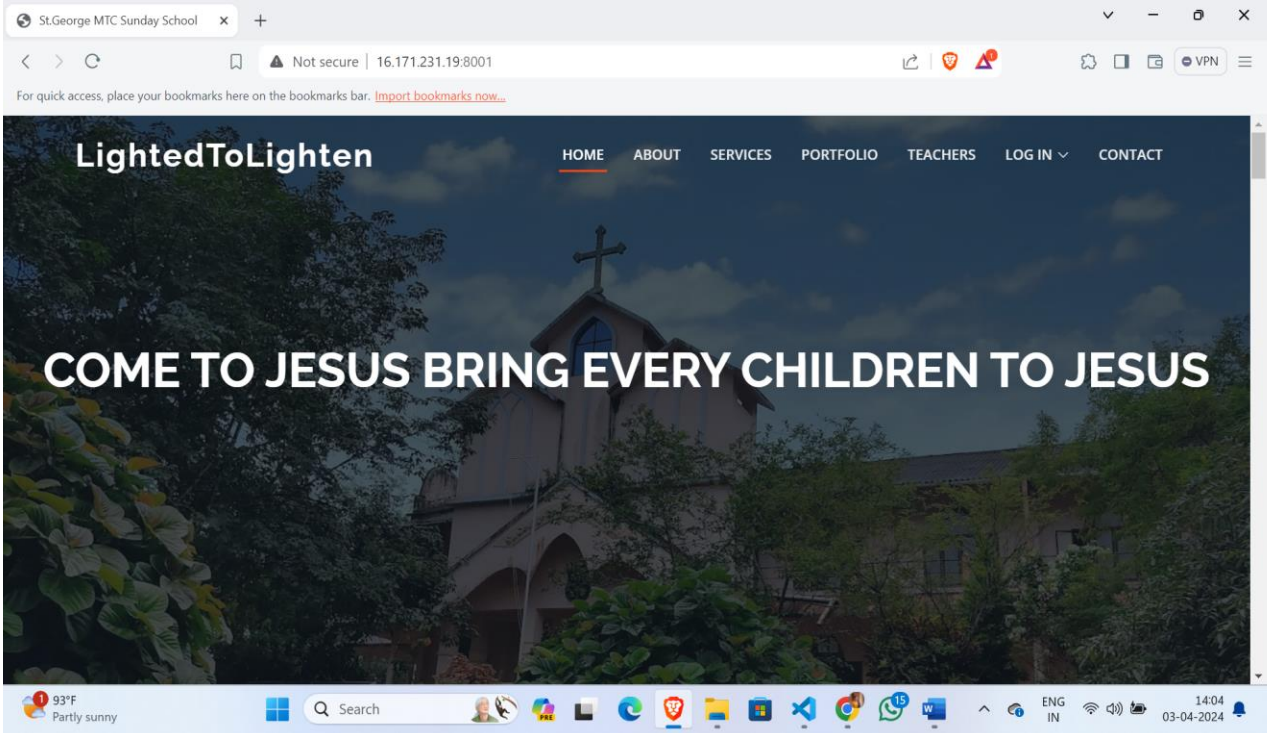
Instance state: Running

Public IPv4 DNS: ec2-16-171-231-19.eu-north-1.compute.amazonaws.com |open address

Hostname type

Private IP DNS name (IPv4 only)

Hosted link: http://16.171.231.19:8001
Screen Shot:



St.George MTC Sunday School

Not secure | 16.171.231.19:8001

For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

LightedToLighten

COME TO JESUS BRING EVERY CHILDREN TO JESUS

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The Sunday School Management System (SSMS) marks a big leap forward in the digitalization of religious education. By blending current technology breakthroughs with traditional religious teachings, SSMS aspires to increase the quality and accessibility of Sunday school experiences. Not only does the system strive to simplify administrative processes, but it also provides a more engaging, customized, and efficient learning method for students. By offering a consolidated platform for teachers, kids, parents, and priests, it facilitates greater communication and smoother administration of resources.

7.2 FUTURE SCOPE

Integration with Advanced Technologies: Future versions of SSMS may interface with artificial intelligence (AI) to better tailor the learning experience, utilizing data analytics to identify student performance trends and offer changes.

Mobile Application: Developing a mobile application version of the SSMS would further expand its accessibility and utilization. Instant alerts, simple access to information, and real-time communication might be enabled with a specialized app.

Virtual Reality and Augmented Reality Modules: These technologies might be utilized to reproduce religious tales or events, giving an immersive learning experience for pupils.

Multilingual Support: To appeal to a worldwide audience, SSMS may add multi-language support, enabling users to browse and study in their favorite language.

Community Features: Introducing features such as discussion forums, virtual meet-ups, and online community activities might further improve the SSMS experience.

Sustainability and Environmental Impact: As the system minimizes the demand for tangible resources, such paper, its contribution to sustainability may be recognized and improved upon.

Enhanced Security Features: With increased worries about data privacy and cyber threats, future versions should concentrate on boosting the security mechanisms, ensuring that user data is safeguarded and the system is resistant to possible intrusions

● 40% Overall Similarity

Top sources found in the following databases:

- 26% Internet database
- 8% Publications database
- Crossref database
- Crossref Posted Content database
- 39% Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	St. Ignatius High School on 2022-07-19 Submitted works	3%
2	St. Ignatius High School on 2022-07-19 Submitted works	2%
3	docshare.tips Internet	2%
4	github.com Internet	2%
5	Purdue University on 2023-05-10 Submitted works	1%
6	Open University of Mauritius on 2021-12-03 Submitted works	1%
7	ABES Engineering College on 2019-04-22 Submitted works	1%
8	techtarget.com Internet	<1%

9	coursehero.com Internet	<1%
10	tutorialspoint.com Internet	<1%
11	Arab Open University on 2018-05-08 Submitted works	<1%
12	Asia Pacific International College on 2023-12-19 Submitted works	<1%
13	numpyninja.com Internet	<1%
14	Universidad TecMilenio on 2024-01-29 Submitted works	<1%
15	br.teradata.com Internet	<1%
16	Imperial College of Science, Technology and Medicine on 2017-06-04 Submitted works	<1%
17	issuu.com Internet	<1%
18	Staffordshire University on 2021-05-15 Submitted works	<1%
19	University of Sydney on 2021-05-12 Submitted works	<1%
20	University of Hertfordshire on 2023-09-18 Submitted works	<1%

21	ir.nmu.org.ua Internet	<1%
22	College of Banking and Financial Studies on 2023-04-19 Submitted works	<1%
23	Segi University College on 2014-04-21 Submitted works	<1%
24	Pentecost University College on 2022-01-25 Submitted works	<1%
25	University of San Francisco on 2023-09-14 Submitted works	<1%
26	RK University on 2016-04-28 Submitted works	<1%
27	Midlands State University on 2014-05-16 Submitted works	<1%
28	University of Teesside on 2015-11-03 Submitted works	<1%
29	testingdocs.com Internet	<1%
30	Trident University International on 2016-05-22 Submitted works	<1%
31	Chandigarh University on 2022-12-20 Submitted works	<1%
32	Wawasan Open University on 2022-11-17 Submitted works	<1%

33	Visvesvaraya Technological University on 2014-11-05 Submitted works	<1%
34	gist.github.com Internet	<1%
35	NCC Education on 2020-12-07 Submitted works	<1%
36	HELP UNIVERSITY on 2022-01-05 Submitted works	<1%
37	Nizwa College of Technology on 2022-12-29 Submitted works	<1%
38	The University of Wolverhampton on 2023-05-16 Submitted works	<1%
39	Leeds Beckett University on 2021-06-09 Submitted works	<1%
40	docplayer.net Internet	<1%
41	University of Winchester on 2022-07-04 Submitted works	<1%
42	University of Wales central institutions on 2012-09-04 Submitted works	<1%
43	Higher Education Commission Pakistan on 2019-10-04 Submitted works	<1%
44	download.51testing.com Internet	<1%

45	es.scribd.com Internet	<1%
46	De Montfort University on 2014-06-09 Submitted works	<1%
47	Griffith College Dublin on 2024-01-26 Submitted works	<1%
48	University of Greenwich on 2014-06-03 Submitted works	<1%
49	Baze University on 2015-11-22 Submitted works	<1%
50	QA Learning on 2024-03-15 Submitted works	<1%
51	Universiti Sains Islam Malaysia on 2017-12-11 Submitted works	<1%
52	University of Maryland, Global Campus on 2023-06-17 Submitted works	<1%
53	Westcliff University on 2024-03-26 Submitted works	<1%
54	The University of Manchester on 2023-12-04 Submitted works	<1%
55	University of Greenwich on 2013-04-23 Submitted works	<1%
56	University of Huddersfield on 2020-09-24 Submitted works	<1%

57	University of Mauritius on 2023-07-28 Submitted works	<1%
58	Info Myanmar College on 2024-02-28 Submitted works	<1%
59	Manipal University on 2023-09-27 Submitted works	<1%
60	QA Learning on 2022-04-01 Submitted works	<1%
61	intellipaat.com Internet	<1%
62	lambdatest.com Internet	<1%
63	Colorado Technical University on 2022-03-03 Submitted works	<1%
64	Manipal University on 2023-09-05 Submitted works	<1%
65	mitpolytechnic.ac.in Internet	<1%
66	automation99.com Internet	<1%
67	Dhirubhai Ambani Institute of Information and Communication on 202... Submitted works	<1%
68	Galgotias University, Greater Noida on 2020-06-09 Submitted works	<1%

69	University of London External System on 2020-02-14 Submitted works	<1%
70	cse.anits.edu.in Internet	<1%
71	ijates.com Internet	<1%
72	Kingston University on 2020-09-13 Submitted works	<1%
73	University of Hertfordshire on 2023-08-28 Submitted works	<1%
74	Mohammed, Nabil Mohammed Abdo. "Exploring Software Security App... Publication	<1%
75	slideshare.net Internet	<1%
76	Liverpool John Moores University on 2023-12-07 Submitted works	<1%
77	TMC Education Group on 2023-04-30 Submitted works	<1%
78	browserstack.com Internet	<1%
79	Asia Pacific University College of Technology and Innovation (UCTI) on... Submitted works	<1%
80	Bedford College on 2021-12-06 Submitted works	<1%

81	Brunswick Community College on 2011-06-23	Submitted works	<1%
82	City University on 2017-04-27	Submitted works	<1%
83	Kennesaw State University on 2023-07-03	Submitted works	<1%
84	Queensland University of Technology on 2022-09-23	Submitted works	<1%
85	Runshaw College, Lancashire on 2017-04-07	Submitted works	<1%
86	Te Pūkenga trading as the Open Polytechnic on 2024-02-15	Submitted works	<1%
87	University of Lancaster on 2018-05-18	Submitted works	<1%
88	goodfood-project.org	Internet	<1%
89	softwareautomata.wordpress.com	Internet	<1%
90	Colorado State University, Global Campus on 2021-06-05	Submitted works	<1%
91	Imperial College of Science, Technology and Medicine on 2023-06-20	Submitted works	<1%
92	San Diego Community College District on 2024-02-13	Submitted works	<1%

93	South West College on 2023-11-24 Submitted works	<1%
94	University Tun Hussein Onn Malaysia on 2016-06-27 Submitted works	<1%
95	University of Durham on 2006-10-18 Submitted works	<1%
96	University of Mauritius on 2019-03-31 Submitted works	<1%
97	University of Southern Mississippi on 2022-12-11 Submitted works	<1%
98	classle.net Internet	<1%
99	Arab Open University on 2012-04-27 Submitted works	<1%
100	Chester College of Higher Education on 2023-05-25 Submitted works	<1%
101	Griffth University on 2022-10-13 Submitted works	<1%
102	NACIT Lilongwe on 2023-11-16 Submitted works	<1%
103	Singapore Institute of Technology on 2023-03-17 Submitted works	<1%
104	Technological University Dublin on 2023-07-26 Submitted works	<1%

105	Victoria University on 2023-09-14 Submitted works	<1%
106	vdocuments.mx Internet	<1%
107	devopsschool.com Internet	<1%
108	Griffith College Dublin on 2020-04-05 Submitted works	<1%
109	Kennesaw State University on 2023-07-04 Submitted works	<1%
110	Monash University on 2023-10-22 Submitted works	<1%
111	repository.tudelft.nl Internet	<1%
112	Asia Pacific University College of Technology and Innovation (UCTI) on... Submitted works	<1%
113	Imperial College of Science, Technology and Medicine on 2023-10-25 Submitted works	<1%
114	Köse, Simge Yelkenci. "Hybrid Meta-Heuristic Approaches for Single a... Publication	<1%
115	Middle East College of Information Technology on 2023-01-20 Submitted works	<1%
116	Scott Tilley, Brianna Floss. "Hard Problems in Software Testing", Sprin... Crossref	<1%

117	Swinburne University of Technology on 2021-06-01 Submitted works	<1%
118	Universiteit van Amsterdam on 2023-06-15 Submitted works	<1%
119	University of East London on 2023-09-11 Submitted works	<1%
120	University of Sydney on 2021-05-11 Submitted works	<1%
121	University of Warwick on 2023-05-01 Submitted works	<1%
122	ia803404.us.archive.org Internet	<1%
123	isiarticles.com Internet	<1%
124	studentsrepo.um.edu.my Internet	<1%
125	Adventist University of Central Africa on 2023-07-25 Submitted works	<1%
126	Amity University on 2015-05-13 Submitted works	<1%
127	Asia Pacific International College on 2023-10-11 Submitted works	<1%
128	Birkbeck College on 2015-09-16 Submitted works	<1%

129	Hong Kong Baptist University on 2024-03-04 Submitted works	<1%
130	Midlands State University on 2021-08-24 Submitted works	<1%
131	Submitted on 1692893744231 Submitted works	<1%
132	University of Ulster on 2023-05-09 Submitted works	<1%
133	University of Wales Institute, Cardiff on 2023-07-10 Submitted works	<1%
134	cyfuture.cloud Internet	<1%
135	pdfcookie.com Internet	<1%
136	Asia Pacific University College of Technology and Innovation (UCTI) on... Submitted works	<1%
137	Colorado Technical University Online on 2006-08-29 Submitted works	<1%
138	Colorado Technical University Online on 2009-01-21 Submitted works	<1%
139	Higher Education Commission Pakistan on 2016-08-15 Submitted works	<1%
140	Kampala International University on 2020-01-08 Submitted works	<1%

141	Kingston University on 2022-07-22 Submitted works	<1%
142	Manipal University on 2023-11-23 Submitted works	<1%
143	NCC Education on 2014-03-04 Submitted works	<1%
144	Sujay Raghavendra. "Python Testing with Selenium", Springer Science ... Crossref	<1%
145	University of London External System on 2013-04-12 Submitted works	<1%
146	University of Strathclyde on 2023-09-02 Submitted works	<1%
147	University of Wales Swansea on 2020-07-04 Submitted works	<1%
148	University of Wales central institutions on 2012-11-30 Submitted works	<1%
149	University of West Florida on 2017-02-04 Submitted works	<1%
150	Vision West Nottinghamshire College on 2024-04-03 Submitted works	<1%
151	ceur-ws.org Internet	<1%
152	pdfcoffee.com Internet	<1%

153	rolandblogsite.wordpress.com Internet	<1%
154	iare.ac.in Internet	<1%
155	yumpu.com Internet	<1%
156	Adventist University of Central Africa on 2022-09-02 Submitted works	<1%
157	Imperial College of Science, Technology and Medicine on 2017-08-29 Submitted works	<1%
158	Royal Holloway and Bedford New College on 2018-08-24 Submitted works	<1%
159	SHAPE (VTC college) on 2015-06-30 Submitted works	<1%
160	Troy University on 2016-10-04 Submitted works	<1%
161	Ganpat University on 2019-04-12 Submitted works	<1%
162	Kampala International University on 2019-12-12 Submitted works	<1%
163	RMIT University on 2020-09-08 Submitted works	<1%
164	Trust Academy on 2015-04-29 Submitted works	<1%