PAPER NAME

Dak_merged.pdf

WORD COUNT

8149 Words

CHARACTER COUNT

44453 Characters

PAGE COUNT

60 Pages

FILE SIZE

4.0MB

SUBMISSION DATE

Jul 22, 2022 3:03 PM GMT+5:30

REPORT DATE

Jul 22, 2022 3:04 PM GMT+5:30

37% Overall Similarity

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

- 37% Internet database
- 11% Publications database
- Crossref database

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The development of objective/subjective question sections, the design of test patterns with exam durations, and the digital administration of tests on computers or mobile devices have all been made easier by modern technology.

A scalable, cost-effective technique for converting conventional assessment and scrutiny, as well as the process of creating question banks and question papers, into online formats is the use of online examination management systems. Users may utilize any desktop, laptop, or mobile device that has a browser to complete their tasks. With the help of this system, they can handle their tasks (such as the evaluation of answer sheets, etc.) from any location in the world using a laptop or desktop. Since there are so many users on this connected system, they can do their task online.

This approach aims to modernize the administration of university exams into online mode.

1.2 PROJECT SPECIFICATION

The system includes 4 modules. They are:

1. **University Admin**

University Admin must have a login into the system. They have a provision to add staffs, college, HOD of each College and have the provision to maintenance works on these system

2. **University Exam Cell Staff**

Staff members have the option to add staff to undertake question paper construction, answer sheet evaluation, and scrutiny by logging in to their dashboard. They are granted the right to create the questions used in each exam. In accordance with demand, they can also offer new courses and disciplines.

3. **College Admin (HOD)**

The heads of each department in each college are known as college administrators. They have a login ID and password to access their dashboard and carry out their tasks. The college administration is in charge of adding faculty to the university website, and they have the authority to suggest professors for assignment to the staff of the university's inspection team, question bank building team, or valuation team depending on their expertise.

4. **College Staff (Teachers)**

Teachers from each department at each college make up the college staff. They have a login

ID and password to access their dashboard and carry out their tasks. As stated by individuals can complete their assigned tasks on their dashboard.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

The process of compiling and interpreting data, identifying issues, and recommending system changes is known as system analysis. System users and system developers must communicate extensively during this problem-solving process. Any system development process should start with a system analysis or research. The system is examined and explored in great depth. The system analyst assumes the position of the interrogator and focuses intently on the operation of the existing system. The system's inputs are detected and taken into account as a whole. Several processes are connected to organizational results. Understanding the issue, identifying the pertinent and important variables, evaluating and synthesizing the many elements, and selecting the best or, at the very least, most acceptable course of action is all part of system analysis.

It is important to do a thorough analysis of the process using a variety of methods, including surveys and interviews. To reach a conclusion, the information gathered by multiple sources must be closely reviewed. Understanding how the system functions is the result. The current system is the name of this system. Now, a full analysis of the current system is being conducted, and issues are being rectified. Now when the firm is experiencing problems, the designers act as a system analyst and makes an effort to find solutions. Propositions are used to present the remedies. The best suggestion is then chosen after being systematically compared to the current system. The user is given the opportunity to approve the idea. On user request, the proposal is reviewed and necessary revisions are made. As time the user is content with the suggestion, this cycle breaks.

The process of gathering and analyzing the data in order to use it for later system studies is termed as preliminary study. A problem-solving procedure known preliminary investigation requires proper cooperation between system users and system designers. It conducts a number of feasibility studies. With the use of these studies, it is possible to establish an approximation of the system activities, which can then be used to determine the best study and analysis techniques to use.

2.2 EXISTING SYSTEM

The current system is not entirely automated. Exams are open to students, but the teacher is the only one who analyzes papers. The drawbacks of the existing system are rectified by the proposed system. The current system must always be modified in order to also include new information, enhance its effectiveness, and make it much more adaptable and secure.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Less Teachers want to go for offline evaluation.
- Teachers also want to go for offline scrutiny process.

2.4 PROPOSED SYSTEM

The current system is not entirely automated. The suggested solution is designed to address every drawback of the current system. Here, we develop the system into a more sophisticated system in online mode using contemporary technologies. This system consists of two parts with distinct functionalities. Each component consists of two modules, each with a unique set of functionalities. One section is for college, the other is for university. One module is for administration, while the other is for workers in the university section. Two modules are included in the college portion of the course, one for department heads (who may also serve as department administrators) and the other for staff (this includes teachers of different departments). In the proposed system, the university administrator can oversee maintenance tasks, examine and change staff information as well as college user information, and have access to view and modify course and subject information.

The privilege of creating question papers for comparable subjects and the opportunity to develop subjects according to the courses based on semesters are both privileges enjoyed by university staff. The staff is in charge of delegating tasks to each college staff member. They can delegate tasks like auditing, creating question banks, valuing response sheets, etc. These decisions are made based on the staff's expertise.

Each member of the college staff is required to complete those tasks within the allotted time. These users are maintained by the department HODs; they each have a unique login

credential for their dashboard, which they use to allocate teachers depending on their qualifications for those tasks.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- **It saves cost: -**

It helps to reduce the cost of providing facilities for valuation teams.

- **It saves time: -**

By using this online service, we can save time for valuation and scrutiny etc.

- **It has the ability to create backups: -**

While using this system we recover the data that may loosed. The data is always kept in the database.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1FEASIBILITY STUDY

To ascertain if the project would ultimately fulfil the objectives of the company given the amount of work, time, and effort invested in it, a feasibility study is conducted. The developer is able to project the project's future and benefits thanks to the feasibility study. An examination of a device's practicability is solely concerned with its impact on the organisation, its capacity to fulfil user needs, and its efficient use of resources. As a result, before being approved for development, newly submitted software typically passes through a proof of concept stage.

The project's technical, economic, and operational viabilities, as well as other elements that were carefully taken into account throughout the feasibility study, are all discussed in the paper's definition of the project's viability. The following traits apply to it:

3.1.1 Economical Feasibility

The growing Cost and benefit analyses must be used to support the growing machine. Criteria to make sure that the project is the centre of the work and that the best is being provided date all the way back to the inception. One factor that affects the creation of a new machine is its price.

Some significant financial queries raised during the initial probe include the following:

The costs conduct a full system investigation.

The cost of the hardware and software.

The benefits in the form of reduced costs or fewer costly errors.

There are no manual costs involved with the suggested system because it was developed as part of a project. Furthermore, the system may be built affordably because all of the necessary materials are already on hand.

System costs, capital expenditures, and hosting fees made up the three primary cost categories for the DREAMS project. All estimates show that the project was created at a reasonable cost. As it was entirely developed using open-source software.

3.1.2 Technical Feasibility

The system must first undergo a technical inspection. The assessment of this feasibility must be built around an overview design of the system's requirements in terms of input, output, programmers, and procedures. After identifying an outline system, the inquiry must then advise the type of technology, necessary steps for developing the system, and ways to operate the system once it has been constructed. The following technical issues were encountered during the investigation:

Does the suggested technology work with the current technology?

Can the system grow if it is improved?

The project should be designed in such a way that the necessary functionality and performance are met within the constraints. The project requires a high resolution scanning instrument and uses cryptography techniques. The fact that a newer version of the same software still functions with an older version means that the system can still be used even though the technology may become outmoded with time. Consequently, there aren't many restrictions on this project. The system was developed with PHP for the front end and a MySQL server for the back end; the project can be finished technically. The system was developed with PHP for the front end and a MySQL server for the back end; the project can be finished technically. The computer used has a potent Intel i3 core processor, 4GB of RAM, and a 1TB hard drive.

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

When The project would be useful once it is designed and put into action since it accomplishes the objectives. After carefully analysing all behavioural parameters, it is concluded that the project is behaviorally feasible. The graphical user interface of Online Examination Management System v2 makes it simple for users to use.

3.2 SYSTEM SPECIFICATION

3.1.3 Hardware Specification

Processor	- Intel core i5
RAM	- 4 GB
Hard disk	- 1 TB
Android	- Android 10 and above
Ios	- 10 and above

3.1.4 Software Specification

Front End	-	HTML, FLUTTER
Backend	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, PHP, CSS

3.2 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server-side scripting language that may be used for general programming as well as web development. Over 244 million websites and 2.1 million web servers presently use PHP. The PHP group, which Rasmus Ledorf formed in 1995, now creates the standard PHP implementation. The acronym PHP, which originally stood for personal home page, is now used to refer to PHP:HypertextPreprocessor. PHP code is interpreted by a web server's PHP processor module, which then creates the web page. PHP commands can be directly put into an HTML source document to handle data instead of interacting with an external file.

3.3.2 AJAX

The abbreviation "Ajax" stands for Asynchronous Javascript and XML. Without refreshing the web page, it is used to connect with the server, improving user experience and speed. Ajax employs the Document Object Model and JavaScript for dynamic content display, coupled

with XHTML for content and CSS for presentation. Traditional web applications use

synchronous requests to send and receive data to and from the server. It denotes that after submitting a form, you will be taken to a new page that has fresh data from the server. When you submit a form using AJAX, JavaScript will send a request to the server, analyse the responses, and modify the current screen. The user would not, in the strictest sense, be aware that anything was even sent to the server. Any format, including plain text, can be used to receive server data; however, XML is frequently utilised. AJAX is a web browser feature that runs independently of a web server.

3.3.3 JQuery

A quick, compact, and feature-rich JavaScript library is jQuery. It greatly simplifies tasks like traversing and manipulating HTML documents, handling events, creating animations, and using Ajax thanks to an intuitive API that is compatible with a wide range of browsers. The way that millions of people write JavaScript has altered as a result of jQuery's adaptability and extensibility. There are many other JavaScript libraries available, but jQuery is likely the most widely used and extensible.

3.3.4 MySQL

The most widely used Open Source SQL database management system, MySQL, is developed, distributed, and supported by Oracle Corporation. Current details about MySQL software are available on the MySQL website.

- **MySQL is a database management system.**

A database is a set of data that has been structured well. A simple grocery list, a photo gallery, or the enormous amounts of information in a business network could all be examples. Data included in a computer database must be added to, accessed, and processed using a database model, such as MySQL Server. Because computers are so good at computing massive amounts of data, database management systems—whether used as independent utilities or as parts of other initiatives a significant role in computing.

- **MySQL databases are relational.**

Instead of storing data in a single big storehouse, a relational database does so in different tables. Physical files made of database structures are organised into groups that are optimised

for performance. A flexible programming environment is offered by the logical model, which comprises objects like database, tables, views, rows, and columns. You specify "points" between various tables as well as the relationships between various data fields, including one, one-to-many, unique, required, or optional. These constraints are upheld by the database, making sure that your application never comes across inconsistent, duplicate, orphan, outdated, or missing data. "Structured Query Languages" is referred to as SQL. The most popular standard language for accessing data is SQL. Depending on your programming environment, you might enter SQL right away (for instance, to generate reports), incorporate SQL statements into programmes written in other languages, or use a language-specific API to mask the SQL syntax. The SQL Standard by ANSI/ISO outlines SQL. The SQL standard has been evolving since 1986, and several variations are now available. In this document, references to "SQL92," "SQL: 1999," and "SQL: 2003" are to the 1992, 1999, and current versions of the standard, respectively. The version of the SQL Standard that is now in effect is referred to as "the SQL standard" in this document.

- **MySQL software is Open Source.**

Considering that the software is open source, anyone can use and alter it. The MySQL software is available for free download from the Internet which can be used by anyone. You are free to examine the source code and change this as necessary. MySQL refers to the GPL (GNU General Public License) when describing what you can and cannot do with the application in specific circumstances. You can get a commercially licenced version from us if you don't like the GPL or need MySQL code to be included in a business application. See the MySQL Licensing Overview for more details.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

You should give it a try if that's what you're searching for. On a desktop or laptop, MySQL Server can run alongside your other software, web servers, and other applications with little to no maintenance. You can utilise all of the memory, CPU, and I/O resources on a system if it is solely used for MySQL.

- **MySQL Server works in client/server or embedded systems.**

A multi-threaded SQL server that supports many backends, a variety of client programmes and libraries, administration tools, and a wide range of application programming interfaces are all included in the client/server system known as MySQL Database Software (APIs). You can use MySQL Server's embedded multi-threaded library into your software to build a standalone solution that is more compact, quick, and manageable.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step in the creation of any designed system or product. The process of design is artistic. For a system to be effective, a proper design is necessary. According to one definition, "design" is "the process of using various methodologies and concepts to specify a process or a system in sufficient detail to enable its physical realisation." It is the process of using a variety of approaches and concepts to thoroughly describe a device, process, or system in order to enable physical implementation. The technical foundation of the software engineering process is software design, which is used regardless of development technique. Through system design, the architectural detail needed to create a system or product is developed. Like any systematic approach, This programme has undergone the best design process conceivable, with every level of effectiveness, performance, and accuracy being fine-tuned. The transition from a document for users to a document for programmers or database employees occurs during the design phase. Logical design and physical design are the two phases of system design.

4.2 UML DIAGRAM

For defining, visualising, creating, and documenting software system artefacts, UML is a standard language. A draught of the UML 1.0 definition was submitted to the Object Management Group (OMG), which created UML, in January 1997.

The acronym UML stands for Unified Modeling Language. Compared to other popular programming languages like C++, Java, COBOL, and others, UML is unique. Software designs are made using the graphical language UML. An all-purpose visual modelling language called UML is used to plan, create, and describe software systems. Although software systems are a typical application for UML, it is not limited to this. Simulated non-software systems are also used. Take the process flow in a manufacturing facility, as an illustration. Although UML is not a programming language, it may be used to write code in a number of other languages using UML diagrams. Analysis and design that are focused on objects are intrinsically related to UML.

UML has been standardised to the point where it is now an OMG standard. A comprehensive UML diagram that depicts a system is made up of all the elements and relationships. The most crucial aspect of the entire procedure is the UML diagram's visual impact. It is completed by using all the additional components. The following nine diagrams are part of UML.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

A use case diagram is a visual representation of how system components interact. In system analysis, a use case is a method for locating, delineating, and organising system needs. In this context, the word "system" refers to something that is being built or run, such a Web site for the sales and delivery of mail-order goods. Use case diagrams are used in UML (Unified Modeling Language), a standard language for modelling actual items and systems.

System objectives include, but are not limited to, planning overall requirements, validating a hardware design, testing and debugging a software product in development, creating an online help reference, or finishing a task focused on customer care. For instance, use cases in a scenario involving product sales would include ordering of items, catalogue updates, payment processing, and customer contacts. Four components make up a use case diagram.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.

- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are created to depict a system's functional requirements. To create an effective use case diagram after identifying the aforementioned things, we must adhere to the following rules.

- The name of a use case is very important. The name should be chosen in such a wayso that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagramis to identify the requirements.
- Use notes whenever required to clarify some important points.

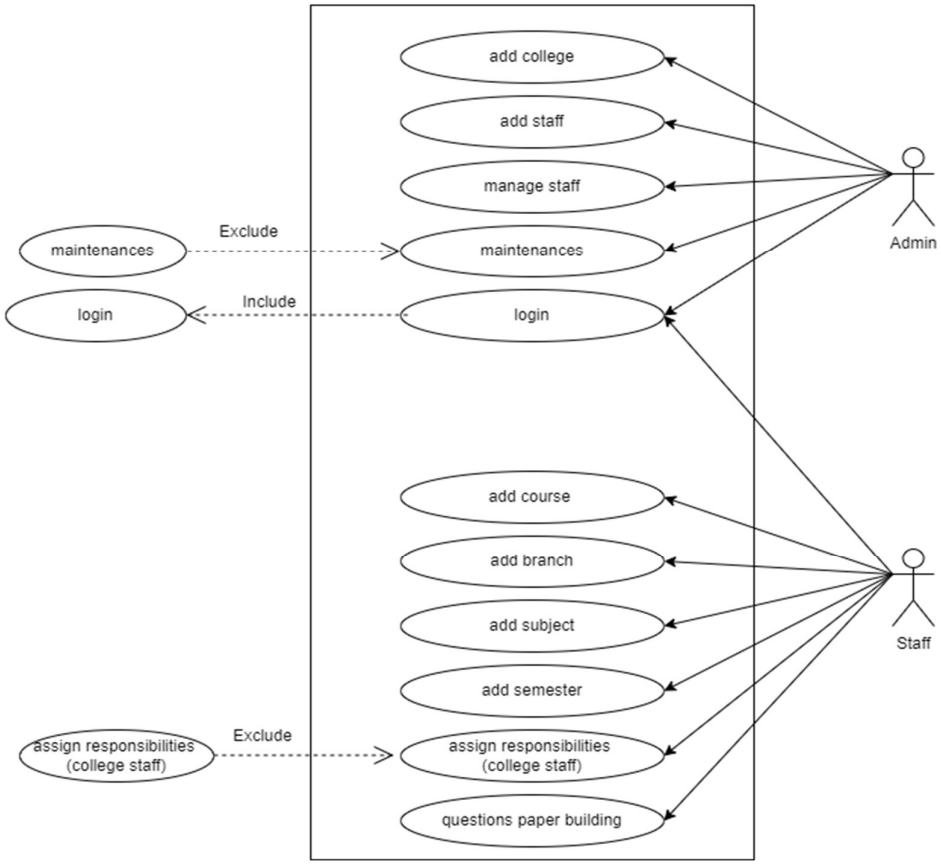


Fig 1: University Use case Diagram

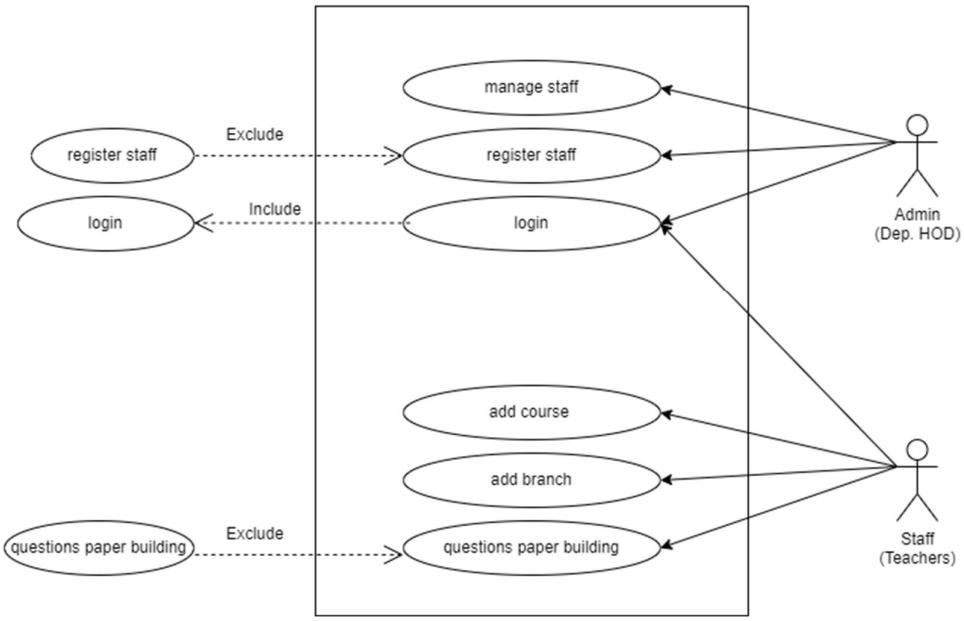


Fig 1: University Use case Diagram

4.2.2 SEQUENCE DIAGRAM

A sequence diagram only shows how various elements interact in a logical order. Specifically, the sequence in which these interactions take place. The terms event diagrams and event scenarios can also be used to describe a sequence diagram. Sequence diagrams demonstrate how and in what order various system components interact. These diagrams are frequently used by businesspeople and software engineers to document and comprehend requirements for both new and existing systems.

Sequence Diagram Notations –

i. **Actors** – In a UML diagram, an actor represents a type of role that interacts with the system and its objects. It is vital to note that an actor is always beyond the scope of the system that we are attempting to depict using the UML diagram. Actors play a variety of roles, including human users and other external subjects. A stick person notation is used to represent an actor in a UML diagram. A sequence diagram can have several actors.

ii. **Lifelines** – A lifeline is a named element in a sequence diagram that represents an individual participant. So, in a sequence diagram, each incident is represented by a lifeline. A sequence diagram's lifeline elements are at the top.

iii. **Messages** – Messages are used to illustrate communication between items. The messages appear on the lifeline in chronological sequence. Messages are represented by arrows. A sequence diagram is built around lifelines and messages.

Messages can be broadly classified into the following categories:

- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message

- Lost Message

iv. Guards – In UML, we utilise guards to model circumstances. They are used to restrict the flow of messages under the guise of a condition being met. Guards play a significant function in informing software developers about the limits associated with a system or a certain procedure.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.

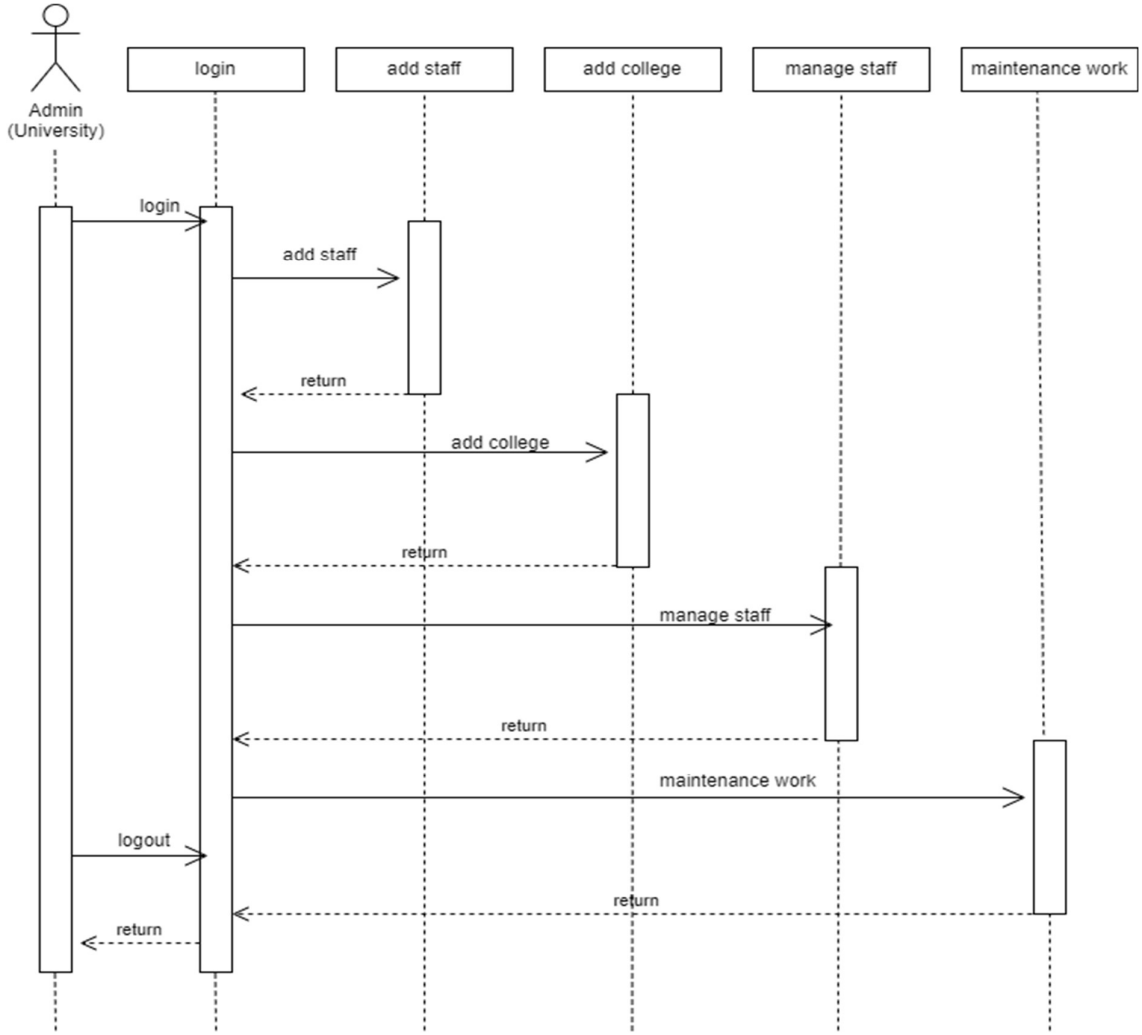


Fig 1: University Admin Sequence Diagram

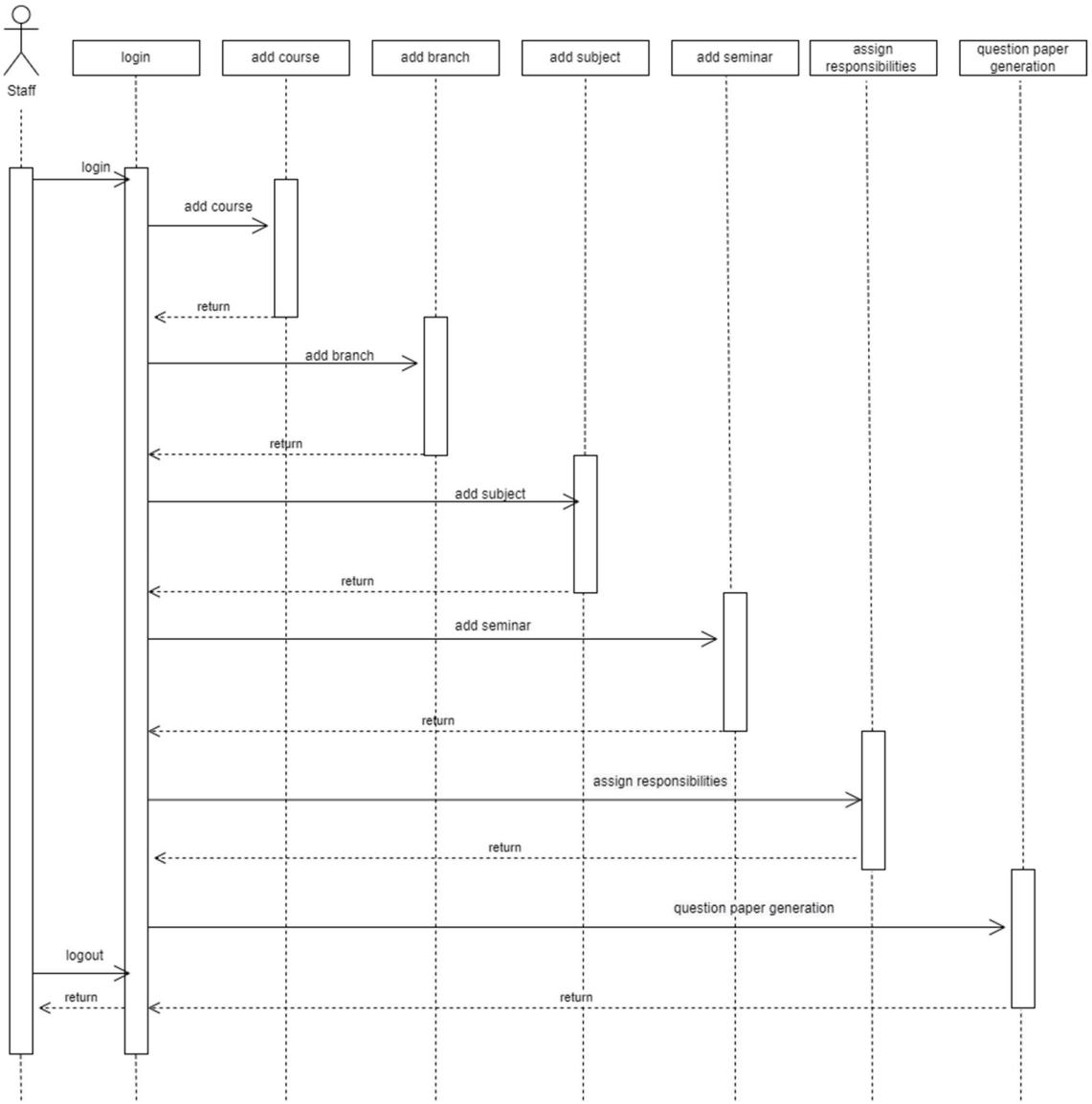


Fig 2: University Staff—Sequence Diagram

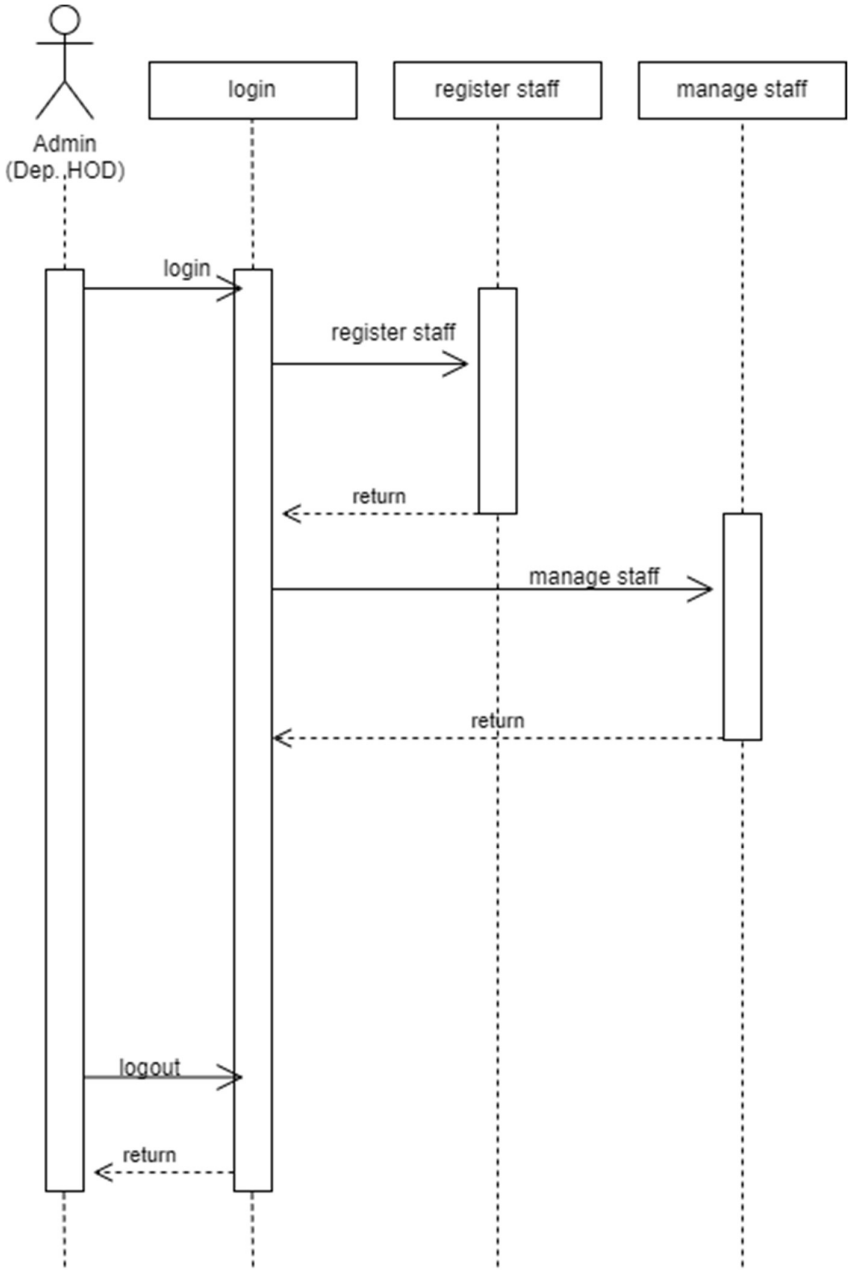


Fig 3: College Admin – Sequence Diagram

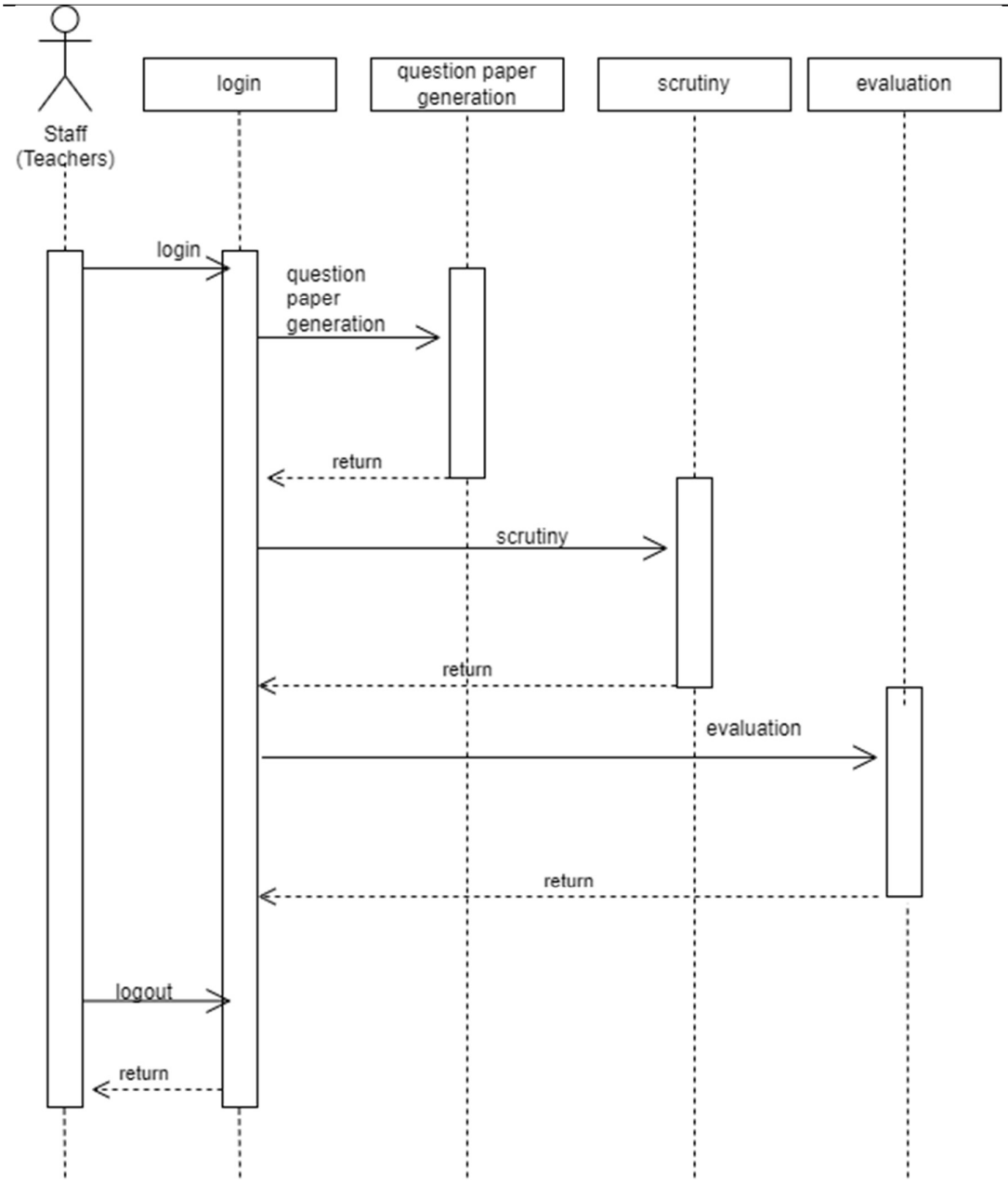


Fig 4: College Staff—Sequence Diagram

4.2.3 State Chart Diagram

The behaviour of a software system is represented by a state diagram. A class, a subsystem, a package, or a complete system's behaviour can be modelled using state machine diagrams in UML. It is also known as a state transition diagram or a state chart. State chart diagrams are a useful tool for simulating interactions or communication within a system as well as between external entities. These illustrations illustrate the event-based system. The status of an item is managed by an event. In application systems, state chart diagrams are used to depict the multiple states of an entity.

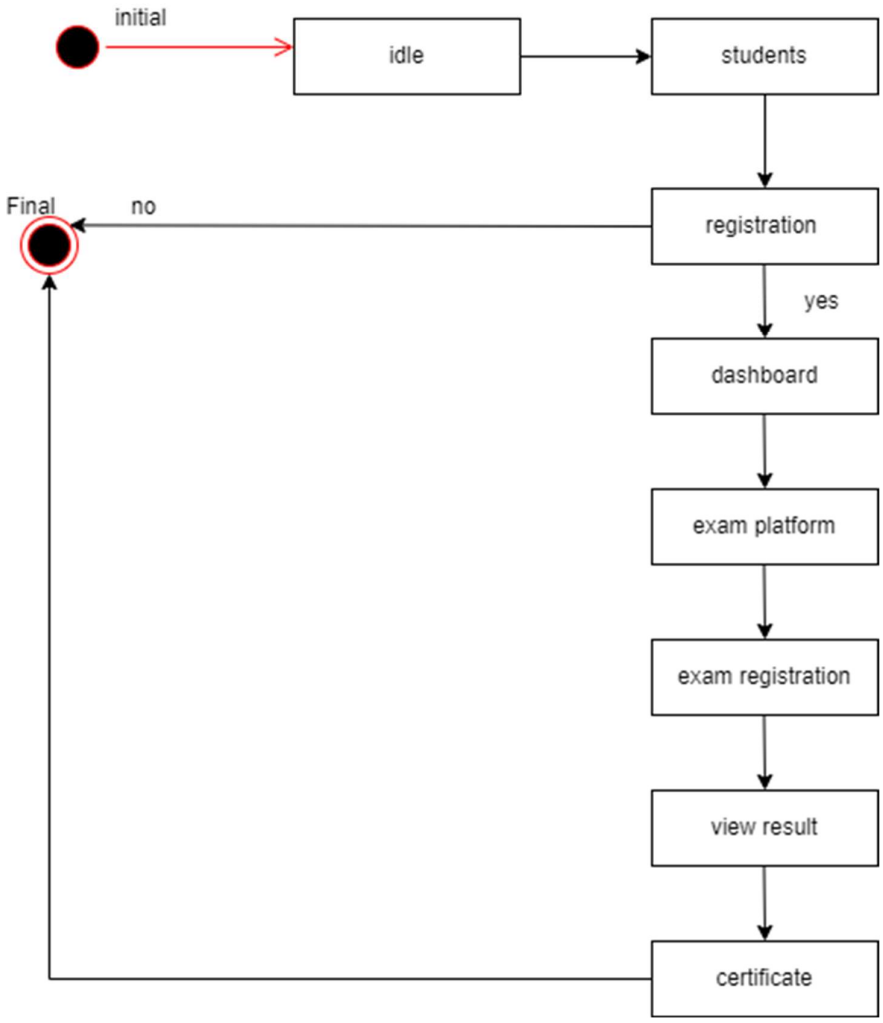


Fig 1: State Diagram

4.2.4 Activity Diagram

At different levels of abstraction, activity diagrams show how activities are organised to create a service. A typical event must be completed by some operations, especially when the operation is intended to complete several distinct tasks that call for coordination. Another typical requirement is how the events in a single use case relate to one another, especially in use cases where activities may overlap and require coordination. It is also suitable for illustrating how a collection of use cases operate together to depict business workflows.

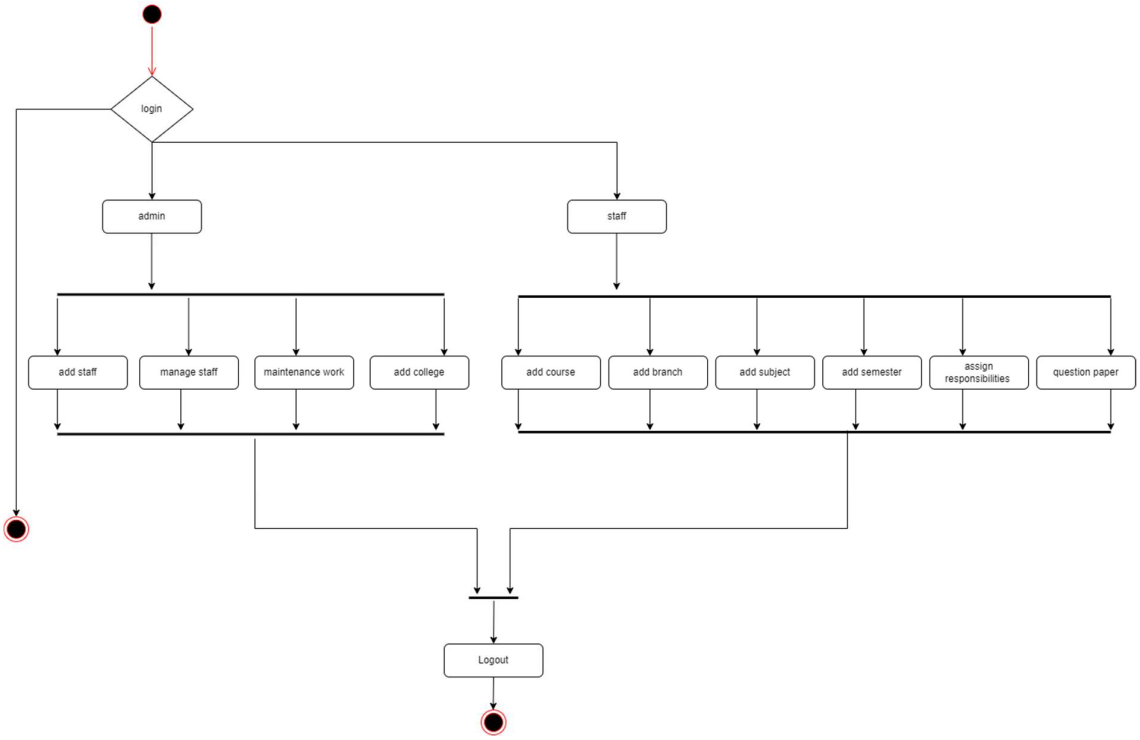


Fig 1: University Activity Diagram

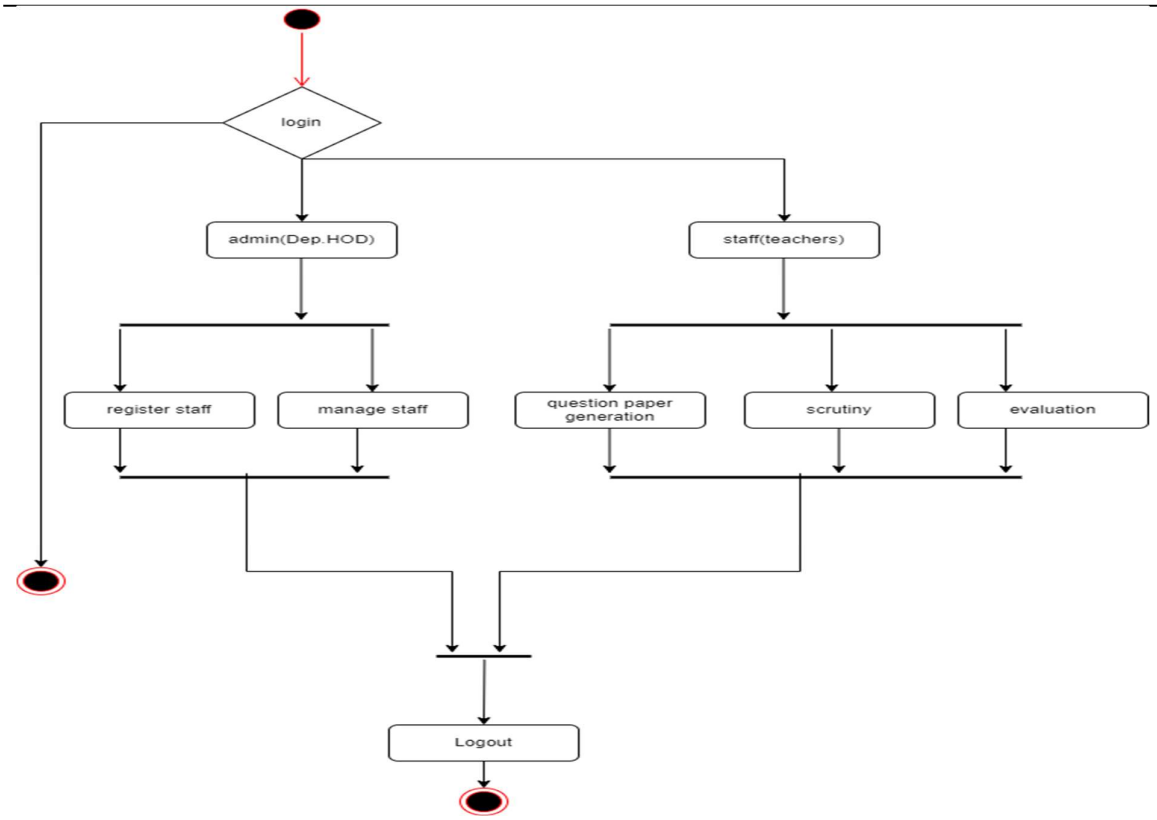
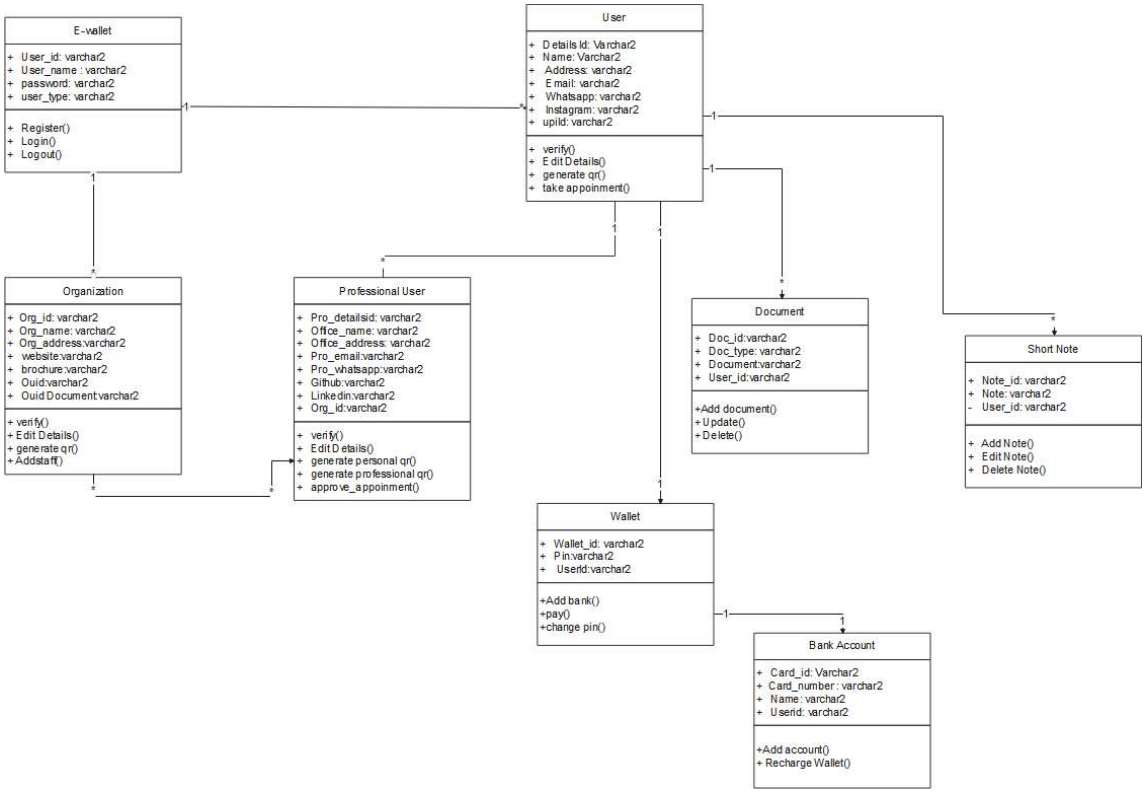


Fig:2 College – Activity Diagram

4.2.5 Class Diagram

A static diagram is a class diagram. It depicts an application's static view. A class diagram is used not only for visualizing, describing, and documenting many parts of a system, but also for building executable code for a software programmer. A class diagram illustrates a class's attributes and operations, as well as the limitations imposed on the system. Because they are the only UML diagrams that can be mapped directly to object-oriented languages, class diagrams are frequently utilised in the modelling of object-oriented systems. A class diagram is a visual representation of a collection of classes, interfaces, affiliations, collaborations, and constraints. A structural diagram is another name for it.



4.2.6 Object Diagram

Since class diagrams are the ancestors of object diagrams, they are dependent on them. An object diagram is a visual representation of a class diagram. Class and object diagrams share the same essential ideas. The static view of a system is also reflected in object diagrams, but this static view represents a snapshot of the system at a particular time. To represent a group of items and their connections, object diagrams are employed.

4.2.7 Component Diagram

Different component diagrams exhibit different characteristics. A system's physical properties are represented using component diagrams. The components that are physically present in a node include executables, libraries, files, documents, and so forth. Component diagrams are used to show how system components are arranged and related to one another. The development of executable systems also uses these representations.

4.2.8 Deployment Diagram

The topology of a system's physical components as well as the locations of its software components are shown in deployment diagrams. The static deployment view of a system is described using deployment diagrams. Deployment diagrams depict nodes and their connections.

4.3 USER INTERFACE DESIGN

4.3.1-INPUT DESIGN

Form Name : Login Page

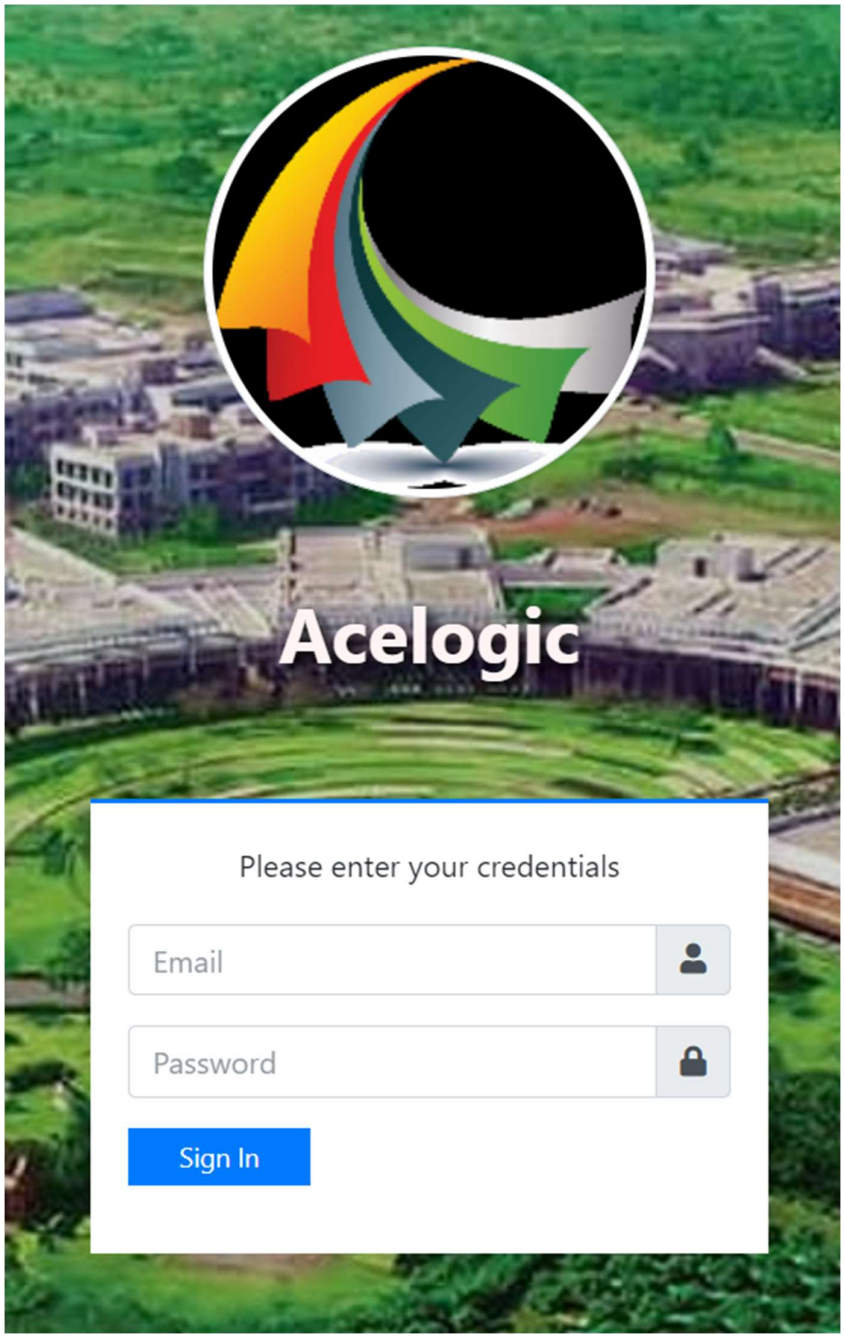


Fig: 1 Login

Form Name : User Registration

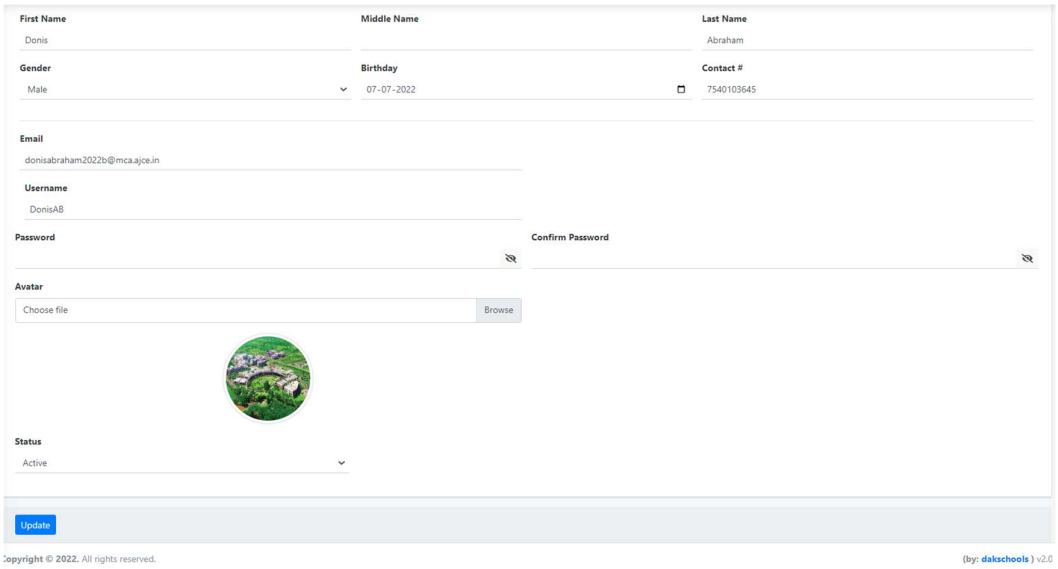


Fig2: User registration

4.3.2 OUTPUT DESIGN

User Login

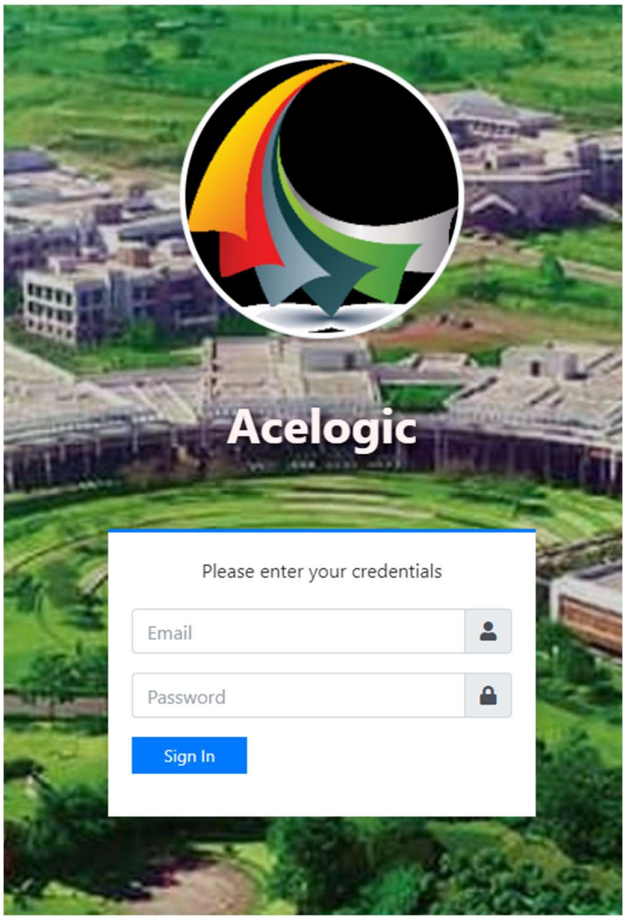


Fig3: User Login

4.4 DATABASE DESIGN

A database is a method of organizing information storage that makes it possible for users to retrieve stored data quickly and effectively. Any database's main goal is to hold data, which must be protected.

There are two phases to the database design process. Gathering user needs is the first step, followed by building a database that as clearly as possible satisfies those demands. This is done independently of any particular DBMS and is referred to as information level design.

The process of converting this information-level design into a design for the particular DBMS that will be used to implement the system in question is done in the second stage. Physical Level Design is the process in question, and it deals with the characteristics of the intended DBMS. Concurrent with a system design is a database design. These two main objectives described below are what the database's data organisation tries to accomplish.

- Data Integrity
- Data independence

4.6.1 Relational Database Management System (RDBMS)

A database is represented as a set of relationships in a relational paradigm. Each relationship resembles a value table or record file. In the formal language of the relational model, a row is known as a tuple, a column header as an attribute, and the table as a relation. An assortment of tables, each with a unique name, make up a relational database. In a tale, a row corresponds to a group of related values.

Relations, Domains & Attributes

One type of relation is a table. Rows make form a table, or a triple. A tuple is an ordered collection of n elements. Characteristics are called as columns. There are relationships established up between each table in the database. By doing this, the referential and entity relationships' integrity is safeguarded. Atomic value collections make up D domains. Declaring a data type from which the domain's data values are derived is a popular method of defining a domain. The domain should have a name to make it easier to understand its values. Each value in a relation is atomic and cannot be broken down.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.6.2 Normalization

To minimise the impact of any changes on data structures, data are assembled as simply as feasible. The formal method of structuring data structures to lessen redundancy and increase integrity is known as normalisation. The process of normalisation involves breaking up a large table into smaller ones and removing duplicate fields. Additionally, it is employed to stop insertion, deletion, and update problems. Keys and relationships are the two concepts typically used in data modelling. A row in a table can only be uniquely identified by a key. The two types of keys are primary keys and foreign keys. A table component or group of components known as a primary key is used to distinguish between records in the same table. A table column known as a foreign key is one that specifically identifies records from another table. All of the tables have been normalised up to the third normal form.

It refers to restoring anything to its original state, as the name suggests. Through normalisation, the application developer tries to create a reasonable data structure into appropriate tables and columns with names that may be quickly associated with the data by the user. Normalization eliminates repeated data groups, preventing data redundancy, a major resource hog for computers. These consist of:

- Normalize the data.
- Choose proper names for the tables and columns.
- Choose the proper name for the data.

First Normal Form

The First Normal Form states that each attribute in a tuple must have a single value from that attribute's domain and that an attribute's domain may only contain atomic values. The 1NF prohibits "relations within relations" or "relations as attribute values within tuples," to put it differently. For attribute values, 1NF only permits single atomic or indivisible values. The data must first be converted into First Normal Form. This can be achieved by grouping the data into tables that contain comparable data kinds. Depending on the requirements of the project, a Primary Key or a Foreign Key is given to each table. We establish a new relationship for every non-atomic characteristic.

Second Normal Form

No non-key attribute in a relationship where the primary key contains several characteristics should be functionally dependent on a component of the primary key, according to Second Normal Form. In this stage, we break down each partial key and its dependent characteristics into separate relations. Keep in touch with the original primary key and any properties that are entirely dependent on it in terms of functionality. This stage helps with data extraction that only needs a small amount of the key. If a connection satisfies all the requirements for first normal form for the primary key and all of the non-primary key characteristics are solely dependent on the primary key, then and only then is it considered to be in second normal form.

Third Normal Form

According to Third Normal Form, relationships shouldn't have a non-key attribute that is functionally determined by another non-key property or a collection of non-key attributes. That is, there shouldn't be any transitive dependence on the primary key. We break down and create a relationship with non-key qualities that serve as functional dependencies for other non-key attributes. Everything that is not entirely dependent on the primary key gets removed at this step. If a relation is only in second normal form and its non-key properties are independent of one another, it is said to be in third normal form.

TABLE DESIGN

Table No 01

Table Name : tbl_users

Primary Key : id

Table Description : To store admin Login information

Name	Type(size)	Constraints	Description
Id	Int(50)	Primary key	Unique ID
Firstname	Varchar(250)	Null	First name of the user
Lastname	Varchar(250)	Null	Last name of the user
Username	Text	Null	Username for login
Password	Text	Null	Password for login
Avatar	Text	Null	Photo of user
Last_login	Datetime	Null	Last login date and time
Type	Tinyint(1)	Null	Role type
Date_added	Datetime	Null	Date of user registered
Date_updated	Datetime	Null	Date of user data updation

Table No 02

Table Name : tbl_system_info

Primary Key : id

Table Description : To store system information.

Name	Type(size)	Constraints	Description
Id	Int(30)	Primary key	Common id
Meta_field	Text	Null	System name
Meta_value	Text	Null	System version

Table No 03

Table Name : tbl_subjects

Primary Key : sub_id

Foreign Key 1 : Course_id,

Foreign Key 2 : User_id

Table Description : To store subject information.

Name	Type(size)	Constraints	Description
Sub_id	Int(255)	Primary key	Common id
Course_id	Varchar(255)	Foreign key	Courses id from tbl_course
Subject_code	Varchar(255)	Null	Subject code for unique identification
Sub_name	Varchar(255)	Null	Subject name
Sub_semester	Varchar(10)	Null	Semester of subject
Subject_credit	Int(5)	Null	score
Subject_status	Tinyint(2)	Null	Status
Syllabus	Text	Null	Syllabus
Delete_flag	Tinyint(2)	Null	If deleted store 0 else 1
User_id	Varchar(10)	Foreign Key	Foreign key from user table

Table No : 04

Table Name : tbl_registered_user

Primary Key : id

Table Description : To store staff information.

Name	Type(Size)	Constraints	Description
Id	Int(30)	Primary key	Common id
Firstname	Text	Null	User first name
Middlename	Text	Null	User second name
Lastname	Text	Null	User final name
Gender	Varchar(100)	Null	Gender info
DOB	Date	Null	Date of birth of the user
Contact	Text	Null	Contact number
Email	Text	Null	User email
Password	Text	Null	User login password
Avatar	Text	Null	User profile pic
Status	Tinyint(1)	Null	User working status
Delete_flag	Tinyint(1)	Null	If user is deleted then value will be 1
Date_created	Datetime	Null	Date of user where added

Date_updated	Datetime	Null	User detail updated date
Scrutiny	Tinyint(4)	Null	If user is appointed as scrutiny then value will be 1 else 0
Invigilator	Tinyint(1)	Null	If user is appointed as invigilator then value will be 1 else 0
Question_bank	Tinyint(4)	Null	If user is appointed as question bank then value will be 1 else 0
Uname	Varchar(20)	Null	username
Exam_cell	Tinyint(2)	Null	If user is appointed as exam cell coordinator then value will be 1 else 0

Table No : 05

Table Name : tbl_registered_college

Primary Key : C_id

Foreign Key : College_id

Table Description : To store college information.

Name	Type(Size)	Constraints	Description
C_id	Int(100)	Primary key	Common id
Date_created	Date	Foreign key	College registered date
College_id	Varchar(30)	Null	College id
College_name	Text	Null	Store College name
College_address	Text	Null	Store college address
College_phone	Varchar(12)	Null	Store college phone number
College_email	Varchar(50)	Null	Store email id
College_website	Varchar(50)	Null	Store website
Manager	Varchar(30)	Null	Store manager name
Manager_email	Varchar(35)	Null	Store manager email id
Status	Tinyint(2)	Null	Store status
Delete_flag	Tinyint(2)	Null	Set flag 1 if deleted else 0
User_id	Int(11)	Foreign Key	User id user who add this college details

Table No 06

Table Name : tbl_question_paper list

Primary Key : proDetails_id

Foreign Key : User_id

Foreign Key : Class_id

Table Description: To Question paper information

Name	Type(Size)	Constraints	Description
Id	Int(30)	Primary key	Common id
User_id	Int(30)	Foreign key	user id
Class_id	Int(30)	Foreign key	Class id
Title	Text	Null	Title of question paper
Description	Text	Null	Description of question paper
Delete_flag	Tinyint(1)	Null	Delete flag is set to 0 and 1
Status	Tinyint(1)	Null	Status
Date_created	Datetime	Null	Date and time of created
Date_updated	Datetime	Null	Date and time of updated

Table No 07

Table Name : tbl_question_list

Primary Key : id

Table Description: To store question list

Name	Type(Size)	Constraints	Description
Id	Int(30)	Primary key	Common id
Question_paper_id	Int(30)	Null	Question paper id
Question	Text	Null	Question
Mark	Double	Null	Mark
Answer Key	Text	Null	Answer key of each questions
Type	Tinyint(1)	Null	Type of question

Table No 08

Table Name : tbl_HOD

Primary Key : Id

Foreign Key :College_i

Table Description: To store hod details

Name	Type(Size)	Constraints	Description
Id	Int(10)	Primary key	
Firstname	Varchar(20)	Null	
Lastname	Varchar(20)	Null	
Username	Varchar(20)	Null	
Avatar	Text	Null	
Type	Int(4)	Null	
Last_login	Date	Null	
Date_added	Date	Null	
Date_updated	Date	Null	
Password	Varchar(255)	Null	
College_id	Varchar(20)	Foreign key	
Department	Varchar(255)	Null	
Department _id	Varchar(10)	Null	

Table No 09

Table Name : tbl_course_list

Primary Key : Id

Foreign Key : user_id

Table Description: To store course details

Name	Type(Size)	Constraints	Description
Id	Int(30)	Primary key	Common id
User_id	Int(30)	Foreign key	User id of the creator
Name	Text	Null	Course name
Description	Text	Null	Course description
Delete_flag	Tinyint(1)	Null	Flag hold if the is deleted or not
Status	Tinyint(1)	Null	Current status active or inactive
Date_created	Datetime	Null	Dated of created
Date_updated	Datetime	Null	Date of updated
Stream	Varchar(50)	Null	Branch name

Table No 09

Table Name : tbl_course_list

Primary Key : Id

Foreign Key : user_id

Foreign Key : course_id

Table Description: To store class details

Name	Type(Size)	Constraints	Description
id	INT(30)	Primary Key	Common id
User id	INT(30)	Foreign Key	Id of the user who created the class
Course id	INT(30)	Foreign Key	Course id
Name	Text	Null	Semester name
Description	Text	Null	Semester details
Delete flag	Tinyint (1)	Null	Hold 1 if the data is deleted else 0
Status	Tinyint (1)	Null	Hold inactive if we disable the class else active
date_created	Datetime	Null	Date of created
date_updated	Datetime	Null	Date of updated

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software testing is the controlled execution of software in order to answer the question, "Does the software behave as specified?" The terms software testing and verification and validation are sometimes used interchangeably. The inspection or testing of goods, including software, for conformance and consistency with an accompanying specification is known as validation. Software testing is only one type of verification; other methods include reviews, analysis, inspections, and walkthroughs. Validation is the process of ensuring that what has been specified is actually what the user desired.

Static and dynamic analysis are two other processes that are frequently connected with software testing. Static analysis examines the source code of software for issues and metrics without actually running the code. Dynamic analysis examines software behaviour while it is running to offer information such as execution traces, timing profiles, and test coverage.

Testing is a series of activities that can be planned ahead of time and carried out in a methodical manner. Testing begins with modules and progresses to the integration of the full computer-based system. Nothing is complete without testing, as testing objectives are critical to the success of the system. There are various guidelines that might serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to havebeen met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan denotes a series of intended actions to be taken in order to complete various testing methodologies. The Test Plan serves as a blueprint for the action to be taken. Software engineers build computer programmes, documentation, and data structures. The software developers are always in charge of evaluating the individual parts of the programmes to ensure that they accomplish the function for which they were developed. An independent test group (ITG) exists to eliminate the inherent issues involved with allowing the builder to test the finished product. The testing objectives should be specified in measurable ways. As a result, the mean time to failure, the cost of finding and fixing the flaws, the remaining defect density or frequency of occurrence, and test work-hours per regression test should all be specified in the test plan. The levels of testing include:

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

5.2.1 Unit Testing

The smallest unit of software design - the software component or module - is the subject of unit testing. Important control pathways are checked within the module's border using the component level design description as a guide. Unit testing's relative difficulty and uncovered scope are established. Unit testing is white-box in nature, and numerous components can be tested in parallel. The modular interface is checked to ensure that data flows into and out of the software unit under test effectively. The local data structure is inspected to guarantee that data stored temporarily retains its integrity throughout all phases in the execution of an algorithm. To ensure that all statements in a module have been performed at least once, boundary conditions are tested. Finally, all error handling paths are put through their paces.

Before starting any other tests, data flow over a module interface must be tested. All other tests are rendered ineffective if data does not enter and exit appropriately. During the unit test, it is critical to perform selective testing of execution pathways. Error circumstances should be expected and error handling paths should be set up to reroute or cleanly end operations when an error occurs. The final duty of the unit testing process is boundary testing. Software frequently fails at its limits.

In the Sell-Soft System, unit testing was performed by treating each module as a separate entity and testing each one with a wide range of test inputs. Some weaknesses in the internal logic of the modules were discovered and fixed. Following coding, each module is tested and run independently. All extraneous code was deleted, and all modules were tested to ensure that they functioned properly and produced the intended results.

5.2.2 Integration Testing

Integration testing is a methodical technique for building the programme structure while also conducting tests to detect interfacing issues. The goal is to use unit-tested components and develop a programme structure governed by design. The entire programme is tested in its entirety. Correction is difficult since isolating causes is confounded by the program's enormous scope. Once these flaws are addressed, new ones develop, and the process appears to loop indefinitely. Following unit testing in the system, all modules were integrated to test for interface inconsistencies. Furthermore, discrepancies in programme architectures were eliminated, and a single programme structure formed.

5.2 Validation Testing or System Testing

This is the last stage of testing. The complete system was tested in its entirety, including all forms, code, modules, and class modules. This type of testing is sometimes referred to as Black Box testing or System testing.

The Black Box testing method focuses on the software's functional requirements. In other words, Black Box testing allows a software engineer to generate sets of input conditions that fully exercise all functional requirements for a programme. Black Box testing looks for defects in the following categories: erroneous or missing functions, interface errors, data structure or external data access errors, performance errors, initialization mistakes, and termination faults.

5.2.4 Output Testing or User Acceptance Testing

The system under consideration has been tested for user acceptance; it should meet the needs of the firm. The software should maintain contact with the prospective system and user while developing and making changes as needed. This is done in relation to the following points:

- Input Screen Designs,
- Output Screen Designs

The aforesaid testing is carried out using several types of test data. The preparation of test data is critical in system testing. The system under study is tested using the test data that has been prepared. While testing the system, test data mistakes are discovered and fixed using the above testing processes, and the corrections are also recorded for future use.

Test cases for User Registration

Project Name: Hostel management system					
Updation Test Case					
Test Case ID: registration			**Test Designed By: Antony Scaria**		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 17-05-2022		
Module Name: Login Screen			**Test Executed By: Ms.Sona Maria Sebastian**		
Test Title: User Registration Details			**Test Execution Date:** 18-05-2022		
Description: Register to system and Registration is completed then login , if someerror occurs, test will fail					
Pre-Condition: User has valid user name and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation to Register Page		Register Page shouldbe	Registrtion page displayed	Pass
2	Provide Valid Registration details	User Name: 99473 15321	User should be ableto Register	User registrion Completed after go to the login page	Pass
3					
4	Click on Login button				
5	Provide profile details	Input profile details	User will be redirected to Login page	Use will be redirected to Login page	Pass
7	Click on register button				
8	Provide invalid information	Input invalid profile details.	User will be stay in register page	User will be stay on that page showing error message	Pass
9	Click on register button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database.					

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class sample {
 public static void main(String[] args) {
System.setProperty("webdriver.chrome.driver","C:\\Users\\antony\\Downloads\\chrom
edriver_win32\\chromedriver.exe" );
  WebDriver driver=new ChromeDriver();
  driver.get("http://localhost/Wallet_web/user/html/registration.php");
  driver.findElement(By.id("firstname")).sendKeys("nithin");
  driver.findElement(By.id("lastname")).sendKeys("george");
driver.findElement(By.id("address")).sendKeys("kanappilly");
driver.findElement(By.id("street")).sendKeys("koonammavu");
driver.findElement(By.id("pincode")).sendKeys("687809");
driver.findElement(By.id("email")).sendKeys("nithingeorgekanappilly@gmail.com");
driver.findElement(By.id("whatsapp")).sendKeys("6689436709");
driver.findElement(By.id("instagram")).sendKeys("Antony.Scaria");
driver.findElement(By.id("upiid")).sendKeys("Scaria@fbl");
driver.findElement(By.id("pin")).sendKeys("!A1234");
driver.findElement(By.id("password")).sendKeys("!Nithin123");
driver.findElement(By.id("conpassword")).sendKeys("!Nithin123");
driver.findElement(By.id("submit")).click();
 }
}
```

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The process of turning a theoretical design into a functional system is known as implementation. It is the most crucial phase in creating a successful new system because it inspires users' confidence in the system's functionality, efficiency, and accuracy. Documentation and user training are its main priorities. Conversion typically happens after or concurrently with user instruction. The process of converting a new, enhanced system design into an operational one is known as implementation.

The user department is currently in charge of most of the work, most of the interruption, and most of the impact on the current system. If the rollout is not well planned and managed, chaos and confusion may occur.

Implementation includes all actions taken to transition from the old to the new system. The new system may be entirely new, take on the role of an existing human or automated system, or make minor adjustments to the latter. To create a dependable system that satisfies the needs of the organisation, proper implementation is necessary. The process of putting the created system into practical use is referred to as system implementation. This includes all steps necessary to switch from the old to the new system. Only after thorough testing and verification that the system complies with the standards can it be installed. The greater the complexity of the system being installed, the more work will be required to complete the three essential components of education and training, system testing, and changeover.The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

The final setup of the software package in its actual environment, as required to carry out the system's functions and reach its objectives, is referred to as software implementation. Many businesses will employ a software designer but not operate the programme themselves. They initially had doubts about the coder, but we must persevere.
ensuring that resistance does not build up, since it is necessary to ensure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

The user should be informed before using the system that the server software must be running on the server in order to access the results. If the server object is not active and functioning on the server, the actual process will not occur.

6.2.1 User Training

The goal of user training is to get the user ready for system conversion and testing. The participants must have faith in their ability to contribute to the new system in order for it to deliver the desired results and advantages. The need for training is increasing as the system gets more complex. The user is trained on how to enter data, respond to error messages, query the database, and call routines that produce reports and perform other crucial tasks.

6.2.2 Training on the Application Software

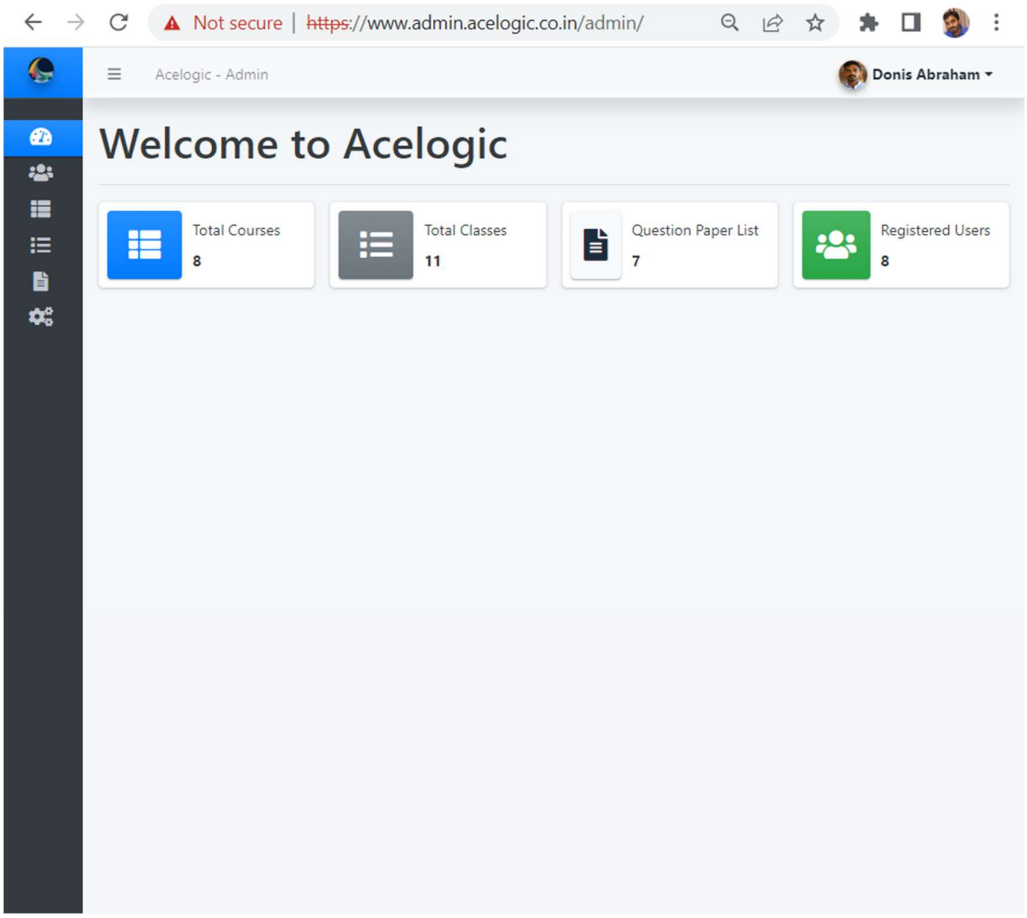
The user will need to be trained on the new application software after getting the fundamental computer awareness training that is very necessary. This will outline the basic tenets of how the new system should be used, including the screen flow, screen design, type of help provided on the screen, types of errors that can occur when entering data, the associated validation check at each entry, and techniques for correcting dates that have been entered incorrectly. Then, while delivering programme training on the application, it should cover the information needed by the particular user/group to utilise the system or a certain system component. Depending on the user group and the organisational level, this training may change.

6.2.3 System Maintenance

Maintenance is the system development conundrum. When a software product performs useful work, it is in the maintenance phase of the software life cycle. A system needs to be properly maintained after it has been implemented successfully. The life cycle of software development must include system maintenance. Making the system more adaptable to changes in the system environment is the goal of system maintenance. Most importantly, software maintenance involves a lot more than just "Finding Mistakes."

6.2.4 Hosting

To host this project I purchased domain name Acelogic.co.in from domain india and purchased host from same company. Which include direct admin panel to manage the data.



CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

With this project now finished, I can say that it served its intended function. The entire concept offers a foundation for students to take exams online while allowing professors to add questions and solutions to the system. The database contains the data, which were saved using the PHP programming language.

This system helps to automated the traditional work in online mode. So we can reduce the cost. Also the whole system will become a perfect system.

7.2 FUTURE SCOPE

The online Examination management system can be upgraded with online examination system with protracting system. This system help automate complete examination process in online. Also, we can develop a system for build question paper with semantic similarity algorithms. This type of algorithms helps to increase the efficacy of the concept.

CHAPTER 8

BIBLIOGRAPHY

37% Overall Similarity

Top sources found in the following databases:

- 37% Internet database
- Crossref database
- 11% Publications database

TOP SOURCES

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

1	issuu.com Internet	7%
2	G.S. Ajith, M.G. Girija, Jinson Devis. "Poly House Environment Monitori... Crossref	6%
3	docshare.tips Internet	5%
4	ia801708.us.archive.org Internet	2%
5	geeksforgeeks.org Internet	2%
6	slideshare.net Internet	2%
7	pdfcoffee.com Internet	1%
8	coursehero.com Internet	<1%
9	cse.anits.edu.in Internet	<1%

10	codingninjas.com Internet	<1%
11	whatis.techtarget.com Internet	<1%
12	medium.com Internet	<1%
13	essaysauce.com Internet	<1%
14	technodocbox.com Internet	<1%
15	docplayer.net Internet	<1%
16	dspace.daffodilvarsity.edu.bd:8080 Internet	<1%
17	automationtestings.com Internet	<1%
18	repository.stcloudstate.edu Internet	<1%
19	studymode.com Internet	<1%
20	silo.pub Internet	<1%
21	scribd.com Internet	<1%

22	kupdf.net Internet	<1%
23	seminarprojects.com Internet	<1%
24	seminartopicsforcomputerscience.com Internet	<1%
25	192.192.246.204 Internet	<1%
26	repository.dinamika.ac.id Internet	<1%
27	tutorialspoint.com Internet	<1%
28	Matteo Biagiola, Andrea Stocco, Ali Mesbah, Filippo Ricca, Paolo Tonel... Crossref	<1%
29	gitlab.doc.gold.ac.uk Internet	<1%
30	Muthukumarasamy Karthikeyan, Renu Vyas. "Chapter 1 Open-Source T... Crossref	<1%
31	dinus.ac.id Internet	<1%
32	en.wikipedia.org Internet	<1%
33	atelier-mc.com Internet	<1%

34	George E. Westlake. "Establishing and maintaining the data-base requi... Crossref	<1%
35	classle.net Internet	<1%
36	comparetrap.com Internet	<1%
37	digilib.uinsby.ac.id Internet	<1%
38	onlyengineering.wordpress.com Internet	<1%
39	guru99.com Internet	<1%
40	itsourcecode.com Internet	<1%
41	itprojectsforyou.com Internet	<1%
42	Nadja Damij, Talib Damij. "Process Management", Springer Science an... Crossref	<1%
43	es.scribd.com Internet	<1%
44	kidsshop-ekb.ru Internet	<1%
45	library.unisel.edu.my Internet	<1%

46	repository.ihu.edu.gr Internet	<1%
47	slideplayer.com Internet	<1%
48	studentsrepo.um.edu.my Internet	<1%
49	uniassignment.com Internet	<1%
50	Zhimin Zhan. "Selenium WebDriver Recipes in C#", 'Springer Science a... Internet	<1%
51	Information Systems Development, 2010. Crossref	<1%
52	Louis Davidson. "Pro SQL Server 2000 Database Design: Building Quali... Crossref	<1%
53	databasestar.com Internet	<1%