

**WOLKITE UNIVERSITY**

**COLLEGE OF COMPUTING AND INFORMATICS**

**DEPARTMENT OF INFORMATION SYSTEM**

**PROJECT TITLE: WEB BASED DISTANCE EDUCATION**

**MANAGEMENT SYSTEM FOR GREAT VISION COLLEGE**

TEAM MEMBER

NAME ID.NO.

1.YESHINEH ABEBAW…………….…...NSR/2201/13

2.ENDALEW SHUMET………………… NSR/0753/13

3.YESUF ASSEFA………………...……..NSR/2204/13

Advisor Name: Mr Badimaw T.

Wolkite University, Wolkite, Ethiopia

May 10, 2024

# **DECLARATION**

This is to declare the project work which is done under the supervision of Mr. Kabtamu D. and having the title “Web Based Distance Education Management system” is the sole contribution of: Yeshineh Abebaw, Endalew Shumet,Yesuf Assefa No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Group Members:

Yeshineh Abebaw \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Endalew Shumet \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Yesuf Assefa \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **APPROVAL FORM**

This is to confirm that the project report entitled web-Based distance education management system for Great vision college submitted to Wolkite University, College of Computing and Informatics Department of Information System by: Yeshineh Abebaw, Endalew shumet, and Yesuf Assefa is oved for submission.

Advisor Name Signature Date

----------------------- ------------------- ---------------

Department Head Name Signature Date

----------------------- -------------------- ---------------

Examiner 1 Name Signature Date

----------------------- -------------------- ----------------

Examiner 2 Name Signature Date

----------------------- -------------------- ---------------

Examiner 3 Name Signature Date

----------------------- -------------------- ----------------

# **ACKNOWLEDGEMENT**

First and foremost, we would like to thank almighty God that permits us to do and enable us to be successful in our final year project and to be complete without any problem as his permission. We also greatly thanks to our advisor Mr.Kabtamu.D ,who have given timely advice as well as encouragement to the completion of our project and his continuous support in every phase of this project. Finally we would like to thank our classmates and friends.

**Table of content**

[DECLARATION I](#_Toc14824)

[APPROVAL FORM II](#_Toc25456)

[ACKNOWLEDGEMENT III](#_Toc30817)

[List of Table VI](#_Toc4774)

[list of figure VII](#_Toc20346)

[ABSTRACT X](#_Toc28991)

[CHAPTER ONE 1](#_Toc20894)

[1.1. Introduction 1](#_Toc13353)

[1.2. Background of Great Vision College 1](#_Toc17281)

[1.3. Statement of the Problem 2](#_Toc27327)

[1.4. Objectives of the Project 2](#_Toc18085)

[1.4.1. General Objective 2](#_Toc2501)

[1.4.2. Specific objectives 3](#_Toc6366)

[1.5.Scope and limitations 3](#_Toc2905)

[1.5.1 Scope of the Project 3](#_Toc6405)

[1.5.2 Limitations 3](#_Toc13146)

[1.6.Feasibility Study 3](#_Toc576)

[1.6.1 Economic Feasibility 4](#_Toc17424)

[1.6.2 Technical Feasibility 4](#_Toc6789)

[1.6.3 Operational feasibility 4](#_Toc3363)

[1.6.4 Political Feasibility 4](#_Toc13895)

[1.7 .Significance and beneficiary of the project 4](#_Toc20245)

[1.7.1 Significance of the Project 4](#_Toc17001)

[1.7.2 Beneficiaries of the Project 4](#_Toc32744)

[1.8 .Methodology of the Project 5](#_Toc20202)

[1.8.1 Data Gathering Method 5](#_Toc8597)

[1.8.2 System Analysis and Design 5](#_Toc10224)

[1.8.3 System Development Model 6](#_Toc8565)

[1.8.4 Development Tools 6](#_Toc14358)

[1.9 .Testing Procedure 7](#_Toc12246)

[1.9.1 Unit Testing 7](#_Toc1569)

[1.9.2 Integration Testing 8](#_Toc14668)

[1.9.3 System Testing 9](#_Toc29079)

[1.10. Project Plan and Budget 10](#_Toc14210)

[1.10.1 Project Plan 10](#_Toc32151)

[1.10.2 Project budget 11](#_Toc10300)

[1.10.3 Project Team Organization 11](#_Toc1091)

[Chapter Two 12](#_Toc699)

[2. The Existing System 12](#_Toc7264)

[2.1 Overview of Existing System 12](#_Toc27190)

[2.2 Users of the Existing System 12](#_Toc25231)

[2.3 Major Functions of Existing System 13](#_Toc32097)

[2.5 Drawbacks of the Existing System: 13](#_Toc25860)

[2.6 Business Rule of the Existing System 14](#_Toc31426)

[CHAPTER THREE 15](#_Toc13510)

[3. PROPOSED SYSTEM 15](#_Toc12661)

[3.1 Functional Requirement 15](#_Toc9701)

[3.2 Non-Functional Requirement 16](#_Toc2312)

[3.2.2 User interface and human factors: 16](#_Toc24186)

[3.2.3 Reliability: 16](#_Toc24840)

[3.2.4 Scalability: 17](#_Toc4704)

[3.2.5 Security: 17](#_Toc20034)

[CHAPTER FOUR 18](#_Toc31245)

[4. SYSTEM ANALYSIS 18](#_Toc10949)

[4.1. System Model 18](#_Toc17663)

[4.2. Object Model 27](#_Toc22304)

[4.3. Dynamic Model 29](#_Toc3202)

[CHAPTER Five 48](#_Toc15993)

[5. SYSTEM DESIGN 48](#_Toc12783)

[5.1 Design Goals 48](#_Toc31851)

[5.2 Current System Architecture 48](#_Toc11149)

[5.3Proposed System Architecture 50](#_Toc15059)

[5.5. Packages 56](#_Toc7604)

[5.6. Algorithm Design 57](#_Toc18924)

[5.7. User Interface Design 58](#_Toc6248)

[CHAPTER SIX 60](#_Toc27675)

[6. IMPLEMENTATION AND TESTING 60](#_Toc31362)

[6.1. Implementation of database 60](#_Toc7674)

[6.2. Implementation of the Class Diagram 65](#_Toc17913)

[6.3. Configuration of the Application Server 65](#_Toc23408)

[6.4. Configuration of Application Security 66](#_Toc25760)

[6.5. Implementation of user interface 66](#_Toc22146)

[6.6. Testing 66](#_Toc23033)

[CHAPTER SEVEN 69](#_Toc32754)

[7. CONCLUSION AND RECOMMENDATION 69](#_Toc2524)

[7.1. Conclusion 69](#_Toc4561)

[7.2. Recommendation 69](#_Toc32074)

[APPENDEX: QUESTIONAIRE 71](#_Toc4996)

[Reference 72](#_Toc24278)

# **List of Table**

[Table 1- 1: Project budget 10](#_Toc8754)

[Table 1- 2:Project Team Organization 10](#_Toc26242)

[Table 4- 1: Use case of the System 21](#_Toc27548)

[Table 4- 2 : Use case Description for Login 24](#_Toc13019)

[Table 4- 3 : Use case documentation of logout 24](#_Toc13158)

[Table 4- 4 : Use case documentation of download module 25](#_Toc24863)

[Table 4- 5 : Use case documentation of view Result 25](#_Toc29522)

[Table 4- 6 :Use case documentation of view course 26](#_Toc11859)

[Table 4- 7 :Use case documentation of Register Instructor 27](#_Toc7540)

[Table 4- 8 :Use case documentation of Assign Instructor 27](#_Toc20686)

[Table 4- 9 : Use case documentation of Add course 28](#_Toc20732)

[Table 4- 10 :Use case documentation of Add student 28](#_Toc27761)

[Table 4- 11 :Use case documentation of create account 29](#_Toc17925)

[Table 4- 12 :Use case documentation of deactivate user account 30](#_Toc30115)

[Table 4- 13 :Use case documentation of activate user account 30](#_Toc12589)

[Table 4- 14 :Use case documentation of Add student result 31](#_Toc24848)

[Table 4- 15 :Use case documentation of upload learning material 31](#_Toc9425)

[Table 4- 16 : Use case documentation of add department 32](#_Toc11903)

[Table 4- 17:Data Dictionary 34](#_Toc25873)

[Table 5- 1: Access Control and Security of the System 56](#_Toc3899)

# **list of figure**

[Figure 1- 1 :Gant chart 9](#_Toc25537)

[Figure 4- 1:Use Case Diagram 23](#_Toc7890)

[Figure 4- 2 :Class Diagram 33](#_Toc24556)

[Figure 4- 3 :Sequence diagram login 35](#_Toc16927)

[Figure 4- 4: Sequence diagram download 35](#_Toc9360)

[Figure 4- 5 :Sequence diagram create account 36](#_Toc4914)

[Figure 4- 6 :Sequence diagram upload learning material 36](#_Toc25887)

[Figure 4- 7 :Sequence diagram add dept. 37](#_Toc14028)

[Figure 4- 8 :Sequence diagram add course 37](#_Toc18580)

[Figure 4- 9 :Sequence diagram register Instructor 38](#_Toc10048)

[Figure 4- 10 : Sequence diagram Assign Instructor 39](#_Toc5606)

[Figure 4- 11 :Sequence diagram view notice 39](#_Toc5603)

[Figure 4- 12 : Sequence diagram view result 39](#_Toc11849)

[Figure 4- 13: Activity Diagram login 40](#_Toc26484)

[Figure 4- 14 :Activity Diagram logout 40](#_Toc4753)

[Figure 4- 15 : Activity Diagram add student 41](#_Toc13395)

[Figure 4- 16 : Activity Diagram upload module 42](#_Toc12529)

[Figure 4- 17: Activity Diagram download module 42](#_Toc29655)

[Figure 4- 18 : Activity Diagram view notice 42](#_Toc9617)

[Figure 4- 19 : Activity Diagram view course result 43](#_Toc29455)

[Figure 4- 20:Activity Diagram view course 43](#_Toc11455)

[Figure 4- 21 :Activity Diagram Assign instructor 44](#_Toc31395)

[Figure 4- 22 : Activity Diagram deactive account 44](#_Toc15311)

[Figure 4- 23 : Activity Diagram create account 45](#_Toc25783)

[Figure 4- 24: Activity Diagram add course 46](#_Toc7657)

[Figure 4- 25 : Activity Diagram Register Instructor 46](#_Toc11010)

[Figure 4- 26 :State chart diagram for login 47](#_Toc7253)

[Figure 4- 27: State chart diagram for upload module 47](#_Toc30324)

[Figure 5- 1 : Proposed System Architecture 50](#_Toc13823)

[Figure 5- 2 : Subsystem Decomposition Diagram. 52](#_Toc17689)

[Figure 5- 3: Development diagram Hardware/Software Mapping 53](#_Toc18507)

[Figure 5- 4: Detailed Class Diagram 54](#_Toc8228)

[Figure 5- 5 : Persistent Data Management Diagram 55](#_Toc23419)

[Figure 5- 6 : Packages 58](#_Toc25372)

[Figure 5-7 : User Interface for Home Page 59](#_Toc984)

[Figure 5-8: User Interface for Login Page 60](#_Toc13929)

**List of Abbreviations**

|  |  |
| --- | --- |
| BR | Business Rule |
| JS | Java script |
| DB | Database |
| GUI | Graphical User Interface |
| HTML | Hyper Text Markup Language |
| ID | Identification |
| XAMP | Platform Apache MySQL Perl PHP |
| MYSQL | My Structure Query Language |
| RAM | Random Access Memory |
| UML | Unified Modeling Language |
| ADMIN | Administrator |
| HD | Hard disk |
| CSS | Cascade Style Sheet |
| UC | Use Case |

# **ABSTRACT**

The project entitled as develop web-based distance education management system for Great Vision College. Currently Great Vision College uses a manual way of giving services to the students. Because of this, there are a number of different challenges that are available in this organization such as lack of space to store records human and computational error, and loss of data, hard to retrieve information, maintenance and so on. So that, to solve the problems we overcome with the new developed system that enhances the existing system which is computerized web-based system. This system is implemented using the programming language HTML, CSS, PHP, uses the MySQL database and WAMP server. This system is user friendly and contains all the necessary details of the College. To conduct the project in effective and efficient manner we are gathered information by using different data collection methodology such as interview, document analysis. As the result of developing new system, we can get effective management of College activities.

# **CHAPTER ONE**

## 1.1. Introduction

This proposal presents the concept of a Web Based Distance Education Management System, which aims to address the challenges faced by educational institutions in managing and delivering distance learning programs. With the increasing demand for flexible and accessible education, especially in recent times, there is a growing need for an efficient and comprehensive system to support Web Based distance education.

The proposed system will serve as a centralized platform for educational institutions to manage various aspects of distance learning, including course administration, student enrollment, content delivery, assessments, and communication. By leveraging technology, this system aims to streamline processes, enhance collaboration, and improve the overall learning experience for both students and educators.

Implementing a Web Based Distance Education Management System will enable educational institutions to overcome geographical barriers and reach a wider audience. It will provide students with the flexibility to access educational resources at their convenience, regardless of their location. Through virtual classrooms, discussion forums, and multimedia content, students will engage in interactive learning experiences.

## 1.2. Background of Great Vision College

Great Vision College, founded in 2013 E.C in the city of Wolkite, the college introduced diploma programs in key departments such as accounting, management, and HRM. These programs were designed to prepare students with practical skills and knowledge, preparing them for the workforce.

As time progressed and the college gained valuable insights into the evolving needs of its students, it decided to take a significant step forward. To further empower its students and provide them with more extensive academic opportunities, the college expanded its department offerings.

In this expansion, Great Vision College introduced new departments like accounting, management, and economics, along with an enhanced HRM program at the degree level. This decision was rooted in the college commitment to nurturing well-rounded, highly skilled graduates who are not only equipped with practical knowledge but are also capable of at

## 1.3. Statement of the Problem

Great Vision College in Wolkite town currently relies on a manual distance education management system, which poses several challenges and limitations. The absence of an efficient and automated system creates obstacle the college's ability to effectively manage and deliver distance learning programs. This statement of problem aims to highlight the key issues associated with the manual distance education management system at Great Vision College. The manual system results in time consuming and labor-intensive administrative tasks. The process of enrolling students, managing course registrations, and maintaining accurate records is prone to errors and inefficiencies. This not only consumes valuable staff resources but also leads to delays and inaccuracies in student data management. Communication between students, instructors, and administrative staff is hindered by the lack of a centralized platform. Important announcements, course materials, and assignment submissions are often exchanged through leading to confusion and miscommunication Traditional education methods have limitations, restricting students to physical classrooms and fixed schedules.

* Managing student data and academic information manually is time consuming and prone to errors.
* The need for a better system to track and analyze student progress.
* These problems are uses manual system to process data.

## 1.4. Objectives of the Project

### **1.4.1. General Objective**

**The main objective of this project is to design and implement web-based distance education management system for Great Vision College in Wolkite.**

### **1.4.2. Specific objectives**

To achieve the general objective mentioned above the following are specific objective:

* Study the existing manual system
* Gathering required information for proposed system
* Analyse the requirements of the proposed system
* Design and build a particular model of this proposed system
* Implementing the model with the flexible user interface
* Test and evaluate the proposed system

## 1.5.Scope and limitations

### **1.5.1 Scope of the Project**

The following scopes are including in the project.

* Upload modules.
* Download module.
* View courses present from department.
* Viewing result online.
* Assign instructor.
* Posting announcement and information.
* Adding, updating and searching student information.
* Online registration
* Online payment
* Online Examination
* Grading system

### **1.5.2 Limitations**

* Security Concerns.
* User Training and Support.

## 1.6.Feasibility Study

To determine the feasibility of the project, we have considered economic,technical,operational,

and political factors:

### **1.6.1 Economic Feasibility**

The project is economically feasible, requiring a reasonable investment to provide significant benefits. It offers cost effective solutions and efficiency gains for students and administrators.

### **1.6.2 Technical Feasibility**

The system to be developed by using technologically system development techniques such as HTML,CSS,JS ,PHP and also database without any problems and the group members have enough capability to develop the project. So the system will be technically feasible.

### **1.6.3** **Operational feasibility**

This Web Based system for distance education in Great Vision College will attain its desired objectives. It can solve the problems in distributing module and tutorials; therefore it will minimize the amount of effort to do all through manually.

### **1.6.4 Political Feasibility**

The project aligns with the goal of providing accessible, efficient, and quality education, and it does not conflict with any government directives. It is politically feasible.

## 1.7 .Significance and beneficiary of the project

### **1.7.1 Significance of the Project**

This system has the following benefits:

* Access of student detail information through internet.
* Reduce resource wastage.
* Support to distribute information.
* Easy to search and manages the student information.
* Reduce employees work load.
* The proper and ordered files of employees which has stability means which is not easily lost.
* Enabling students to learn at their own pace and convenience.

### **1.7.2 Beneficiaries of the Project**

The project will benefit various stakeholders, including:

* Educational Institutions: Streamlined operations, improved course quality, and reduced administrative overhead.
* Students: Access to user-friendly, interactive online learning platforms.
* Teachers: Tools for efficient course management and effective communication with students.
* Administrators: Data-driven insights for curriculum development and quality enhancement.
* IT Teams: Opportunities to contribute to a cutting edge educational infrastructure.

## 1.8 .Methodology of the Project

### **1**.**8.1 Data Gathering Method**

There are different data gathering technique which helps us to understand the present systems

General activities and its procedures. We used three methodologies to gather the needed information. The methods we use for data collection are:

#### **1.8.1.1 Interview**

As a method for the collection of data about the activities in College of distance education we use

interviewing method to understand peoples who belongs to the current system also we raised

questions that helps us to develop the new system.

* **Ask the background of the organization?**
* **Ask how to work the current system?**
* **Ask the problem of current system?**

#### **8.1.1.2. Document Analysis:**

* **we analyze some documents and forms found in the office to get information about the current system.**
* **Assessment Review: Ensuring that assessments, such as quizzes, exams, and assignments, are aligned with learning outcomes and are fair and unbiased.**
* **Financial Document Review: Assessing financial documents, such as budgets and expenditure reports, to ensure efficient resource allocation.**

### **1.8.2** **System Analysis and Design**

In the system analysis and design phase of a project we will use the object oriented approach that examines requirements from the perspective of the class and objects found in the problem domain. The reasons that we will use the object oriented approaches are:

* We can reuse methods for avoiding redundancy.
* To make it easier for teams of designers and programmers to work in a single software project
* The data and functions are encapsulated in the objects that help us for easily debugging purpose.
* It will increase consistency among analysis, design and programming activities.
* It will improve communication among users, analysis, design and programming
* Modification of the object implementation is easy because objects are loosely coupled.
* Understanding of the structure is easy because object oriented modelling represents real world entities.

### **1.8.3 System Development Model**

In the system development model to develop good software we will use agile model because we have different reason such:

* We will iterate and increment with in each phase.
* We can easily control it, and it is flexible for developers.
* It primarily concentrates on the deliverable and fewer about paperwork.

### **1.8.4 Development Tools**

#### **1.8.4.1 Hardware tools**

* Computer with internet connection
* Secondary storage device
* Flash disk (8 GB)
* Memory :8GB RAM
* Hard disk:300GB

#### **1.8.4.2 Software tools**

The different kind of software we used for developing the project is:

* MySQL server – to develop database application
* Microsoft office word 2013 for documentation.
* Edraw Max—to design Use case Diagram, design Sequence Diagram, design
* Activity Diagram and. Class Diagram, Deploy Diagram and other diagram
* Programming Language: - PHP, CSS, JavaScript and HTML

## 1.9 .Testing Procedure

Developing software for an Web Based Distance Education Management System is a complex process. No matter how diligently we progress through the phases of requirements, analysis, design, specification, and implementation, it's essential to ensure that significant faults are prevented from arising in the first place. To achieve this, a dedicated testing phase is required to identify and rectify any remaining issues before the system is ready for use. The testing process involves different levels, including unit testing, integration testing, and system testing.

### **1.9.1 Unit Testing**

During unit testing, developers of the Web Based Distance Education Management System will assess individual components, sub-procedures, and functions.

Test Scenario:

Scenario 1: Testing User Authentication and Examination Submission

Test whether the user authentication function correctly authenticates valid users.

Test whether the system allows students to submit examinations and records their submissions accurately.

Test Plan:

Identify the components and functions related to user authentication and examination submission.

Create test cases for valid and invalid user credentials.

Create test cases for submitting examinations.

Execute the tests by providing different inputs.

Verify that authentication and examination submission processes work as expected.

Test Cases:

Test Case 1:

Verify a valid user can log in successfully.

Input: Valid username and password.

Expected Output: User is logged in.

Test Case 2:

Verify an invalid user cannot log in.

Input: Invalid username and password.

Expected Output: Authentication failure.

Test Case 3:

Verify that a student can submit an examination.

Input: Student selects an examination and submits answers.

Expected Output: Examination submission is recorded.

### **1.9.2 Integration Testing**

In integration testing, we assess how different system components work together to achieve the intended goals of the subsystems in the Web Based Distance Education Management System.

Test Scenario:

Scenario 2: Testing Course Registration, Payment, Examination Integration, and Grading

Test how the course registration module integrates with the payment system.

Ensure that students can register for a course, make a payment, take examinations, and receive grades accurately.

**Test Plan:**

* Identify the components related to course registration, payment, examination submission, and grading.
* Create test cases for different scenarios of registration, payment, examination submission, and grading.
* Execute the tests by simulating the entire process.
* Verify that the integration works as intended.

**Test Cases:**

**Test Case 1:**

* Verify that the system handles payment failures gracefully.
* Input: Student's payment fails.
* Expected Output: Registration is not confirmed, and an error message is displayed.

**Test Case 2:**

* Verify that a student can register for a course, pay for it, take an examination, and receive a grade.
* Input: Student registers, pays, takes an examination, and receives a grade.
* Expected Output: Registration, payment, examination submission, and grading are successful.

## 1.9.3 System Testing

System testing evaluates how well the various subsystems of the complete Web Based Distance Education Management System function cohesively to achieve the desired educational outcomes. This phase ensures that the system operates seamlessly and efficiently when used by students, instructors, and administrators.

**Test Scenario:**

**Scenario 1:**

* Testing End-to-End Educational Outcome, Examination, and Grading
* Evaluate how the entire system functions to support the educational process, including examinations and grading.
* Ensure that students, instructors, and administrators can use the system efficiently to achieve their goals.

**Test Plan:**

* Identify key user roles (students, instructors, administrators) and their goals, including examination and grading processes.
* Create test cases to cover common user journeys for each role, including examination and grading tasks.
* Execute the tests by simulating the actions of users in different roles.
* Verify that the system operates seamlessly and efficiently for all users in the context of examinations and grading.

**Test Cases:**

**Test Case 1:**

* Verify that an instructor can create and manage courses, grade examinations, and communicate with students.
* Input: Instructor logs in, creates a course, grades examinations, and communicates with students.
* Expected Output: All actions are completed successfully, including examination grading.

**Test Case 2:**

* Verify that an administrator can manage user accounts, oversee course enrollment, generate reports, and manage the grading system.
* Input: Administrator logs in, manages user accounts, checks course enrollment, generates reports, and manages grading.
* Expected Output: All actions are completed successfully, including grading tasks.

## 1.10. Project Plan and Budget

### **1.10.1 Project Plan**



Figure 1- 1 :Gant chart

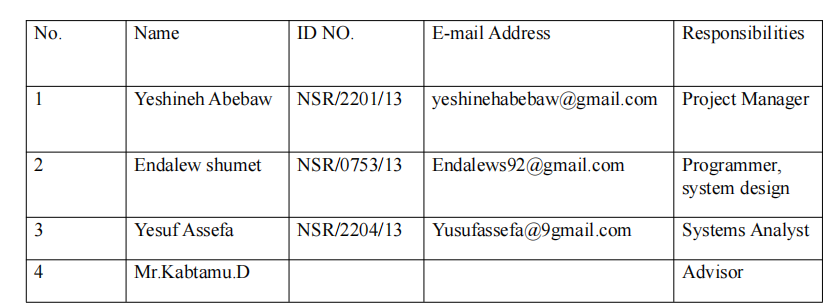
### **1.10.2 Project budget**

Table 1- 1: Project budget

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Material** | **Unit** | **Price per unit** | **Total price** |
| 1 | Flash Disk | 3(16 GB) | 3\*400 | 1200 |
| 2 | Paper | 1(‘desta’) | 500 | 500 |
| 3 | Pen | 3 | 3\*20 | 60 |
| 4 | Pc(hp) | 1 | 40000 | 40,000 |
| 5 | Total |  |  | 41,760 |

### **1.10.3 Project Team Organization**

Table 1- 2:Project Team Organization



# **Chapter Two**

## 2. The Existing System

### **2.1 Overview of Existing System**

The existing system in Great Vision College is manually operated.Different activities are performed as follow The department gives education service for the students at these sectors. First the students apply to register by coming personally . Then the department has its own criteria to approve the student’s application. registrar checks the registration form filled by the student weather it satisfies the criteria or not. If the student satisfies these criteria he/she will be approved to be registered. Then after the approval, the student will pay the payment for the courses that he/she will take. Then the student will be registered and start the class by the schedule posted by the department. The department handles this information manually. And the file management system is paper based. So, the information recording or filling and retrieval system is too tedious, error prone, and also time taking. Distributing modules, allocating assignments, announcing registration and examination dates and some other activities are performed by there responsibilities.

### **2.2 Users of the Existing System**

The user of existing system is students,, Instructors,Admin,.Their activities in the existing systems are:-

* **Instructors:**
* Give teaching, tutorials and makeup services.
* Make grade and sends grade to the dep’t head
* Students:
* Apply to be registered.
* Collect modules or reading materials from instructor.
* Collect assignments from the instructor.
* Submit assignments.
* View information and announcements.
* Admin:
* add department
* assign instructor
* delete department
* Assign Course
* update notice

### **2.3 Major Functions of Existing System**

* Module Distribution: Instructors play a pivotal role in distributing learning modules to students, a task that is currently part of the manual operations within the system.
* Tutorial Sessions: Instructors are responsible for conducting tutorial sessions, adding to the manual workload and potentially contributing to inefficiencies in the teaching process.
* Course Result View:Instructors are also tasked with view course results, requiring students to interact directly with them to access their grades, quizzes, and test outcomes.
* Registration Process:The registrar's office manages the registration process, encompassing responsibilities such as preparing grade reports, transcripts, and registering students for courses.
* Verification of Academic Schedule:The academic vice president verifies the academic schedule prepared by the registrar, ensuring coherence and compliance with the college's academic objectives.

### **2.5 Drawbacks of the Existing System:**

* Manual Module Distribution: The reliance on instructors for the manual distribution of learning modules contributes to inefficiencies, potential errors, and increased workload on teaching staff.
* Student Result Retrieval: Students having to go back to instructors to view their course results introduces a time-consuming and cumbersome process, adversely affecting the overall student experience.
* Registrar Office Workload: The registration process, managed by the registrar's office, involves various tasks such as preparing grade reports, transcripts, and managing academic schedules. The manual nature of these tasks may result in delays and errors.
* Resource and Time Wastage: The overall system exhibits a significant wastage of college resources and time, particularly during the preparation, duplication, and distribution of study modules. This not only impacts operational efficiency but also contributes to unnecessary costs.
* Work Overload on College Employees: The duplication and distribution of modules for each course, coupled with manual tasks, create an overburden on college employees, potentially affecting their productivity and job satisfaction.
* Student Dissatisfaction: The manual processes, coupled with the identified drawbacks, contribute to overall student dissatisfaction. The need for students to invest additional time, effort, and money in navigating these processes detracts from their overall educational experience.

### **2.6 Business Rule of the Existing System**

BR1: The students must full fill minimal criteria of minister of Education to be registered or

possess a competency certification of Level IV from COC Agency.

BR2: Registration date is starting from the announced day to 15 days continued and after the last

days of registration the students will punish 50 birr in addition to normal registration fee to

register.

BR3: The student monthly fee should be based on credit hour of the course, for a credit it is only

90 birr.

BR4: The total credit hour for the courses in a semester must be between 11 and 14.

BR5: The Assessment method for students is 50% Assignment and 50% final Exam.

BR6: The students to take the next course they must complete the pre-request course of the

previous semester.

BR7: The students should reasonable and have evidence for their reason that why they cannot full fill their requirement.

BR8: All students must have their own identification card.Top of Form

# **CHAPTER THREE**

## 3. PROPOSED SYSTEM

The proposed system that we analyze can solve some portion of the existing system. When we see the solution, making the College automate system, it will solve most of the problems in the teaching-learning process. This project has much significance

* Reduce the extravagance of the college resources.
* Reduce the time and task required to perform the operation within the College.
* For students, better satisfaction of the speed provided by the instructor in course material distributing, seeing course result.
* And it improves the moral (motivation) of the users.

### **3.1 Functional Requirement**

The functional requirements are features that needed to satisfy the users, the proposed system will be designed according to the roles of the users.

* System Administrator:

The system will allow the system administrator to perform the following tasks.

* Allow to create student and instructor account
* Allow to update student and instructor account
* Allow to delete student and instructor account
* Allow to view user’s information
* Allow Add course
* Allows to add department
* Allows to assign instructor
* Allows to delete department
* Allows to view module
* Allow to generate user’s password
* Instructor:

The system will allow the instructor to perform the following tasks.

* Allow to post progress quiz
* Allow to upload assignment
* Allow to upload module
* Allow to view assign course
* Allow to view student’s information
* Student:

The system will allow the student to perform the following tasks.

* Allows to update her/his profile.
* Allow to download module
* Allow to submit assignment
* Allow to view grade report
* Allow to view his/her profile
* Allow to take progress quiz

### **3.2 Non-Functional Requirement**

Non-functional requirements describe aspects of the system that are not directly related to the functional behavior of the system, only related to how the system will do the functional requirement. Non-functional requirements include a broad variety of requirements that apply to many different aspects of the system, from usability to performance.

#### **3.2.1 Performance:**

* The system should respond to user requests within 2 seconds to ensure a seamless and efficient user experience.
* It should be able to handle a concurrent user load of at least 1000 users without significant performance degradation.

#### **3.2.2 User interface and human factors:**

* Our proposed system will have an interactive interface and will provide a simple, attractive, flexible interface for every user of our system.
* Users of the system must also have legal practice experience and basic digital literacy.

#### **3.2.3 Reliability:**

* The system should have a 80.9% uptime, allowing for scheduled maintenance.
* It should be capable of recovering from failures within 5 minutes without data loss.

#### **3.2.4 Scalability:**

* The system should be scalable to accommodate a 20% growth in the number of users and courses over the next year.
* It should support an increasing amount of concurrent users as the user base expands.

#### **3.2.5 Security:**

* User data should be encrypted during transmission and storage.
* The system should have measures in place to prevent unauthorized access, including robust user authentication and access controls.

#### **3.2.6 Usability:**

* The user interface should be intuitive and user-friendly, requiring minimal training for users to navigate the system.
* It should be compatible with common web browsers and accessible to users with disabilities.

#### **3.2.7 Compatibility:**

* The system should be compatible with the latest versions of major web browsers (Chrome, Firefox, Safari, Edge).
* It should support various devices, including desktops, laptops, tablets, and smartphones.

#### **3.2.8Availability:**

* The system should be available 24/7, with planned downtime communicated to users in advance.
* In the event of unexpected downtime, a user-friendly maintenance page should be displayed.

#### **3.2.9 Data Backup and Recovery:**

* Regular automated backups of the system data should be performed daily, and backups should be stored securely offsite.
* The system should have a data recovery plan in place to restore data in case of data loss or corruption.

**Documentation:**

Our system will have well-defined documents which help to easily maintain the system. We will also prepare a short and understandable file for users on how to use the system. And the development process will be provided for the user to read to know about the process and what type of model used to develop the system

# **CHAPTER FOUR**

## 4. SYSTEM ANALYSIS

In this chapter we will deal with the proposed system by using use case diagrams, use case descriptions, object model, dynamic model (sequence diagrams and activity diagrams). After identifying the actors and the use cases of our new system, the use cases are developed and textual descriptions are depicted based on the use case. Next the sequence diagram will be depicted based on the use cases which are developed for the newly proposed system. Activities will be represented by activity diagram. Precondition, post condition and flow of event will be covered under this chapter.

### **4.1. System Model**

This section consists of the modeling of the proposed system using object oriented methodologies such as unified modeling language (UML). Here represent the proposed system by using different system models such as use case models, object models, dynamic models, that describe the problem to be solved and as system models represented by graphically they are more understandable than more detailed natural language description of the system requirement.

#### **4.1.1. Use Case Model**

A use-case model is a model of how different types of users interact with the system to solve a problem. A use-case model consists of a number of model elements. The most important model elements are: use cases, actors and the relationships between them. A use case diagram is a summary of who uses the system and what they can do with it. Use case diagram shows the relationships between users (actors) and use cases with in a system.

The following use cases have been identified from the system specification:

Table 4- 1: Use case of the System

|  |  |
| --- | --- |
| Download Files | View Academic sechedule |
| Post quiz | Assign instructor |
| Create and Delete Account | Create Account |
| View user information | View report |
| Approve Grade Report | Assign course |
| View grade and announcement | View Module |
| Delete Account | Finance Report |
| Change Password | Generate User password |
| Add Course | Login,Logout |

**Actors**

In this system the identified actors that will be participating in the system are:

* Student
* Admin
* Instructor



Figure 4- 1:Use Case Diagram

##### **4. 1 .1.1. Use Case Description**

Table 4- 2 : Use case Description for Login

|  |  |  |
| --- | --- | --- |
| Use case number | UC 01 | |
| Use case name | Login | |
| Actor | Instructor, students , System Administrator | |
| Description | Checking the intended user is authorized or not | |
| Precondition | The user must have username and password | |
| Post condition | The users successfully login. | |
| Basic course of action | User action | System response |
| 1.The user opens the system  3. The user enters user name and password then click login button.  6. End use case. | 2. The system displays the login page.  4. The system checks the username and password.  5. The system opens the users’ home page. |
| Alternative course of action | If the username and password is invalid, the system displays an error message, then go back to step 3 of basic course of action. | |

Table 4- 3 : Use case Description of logout

|  |  |  |
| --- | --- | --- |
| Use case number | UC 02 | |
| Use case name | Logout | |
| Actor | Instructor, students , System Administrator | |
| Description | After doing any private activity in the system the user log out from the system. | |
| Precondition | The user should be in private page. | |
| Post condition | The user is in public page. | |
| Basic course of action | User action | System response |
| 1. The user clicks the logout button.   3. End use case. | 1. The system displays the login page. |
| Alternative course of action | If connection is fail, try again. | |

Table 4- 4 : Use case Description of download module

|  |  |  |
| --- | --- | --- |
| Use case number | UC 03 | |
| Use case name | Down load module/learning material | |
| Actor | student | |
| Description | Downloading learning materials | |
| Precondition | The student must be login and the module must be uploaded before | |
| Post condition | The student successfully downloading learning materials | |
| Basic course of action | User action | System response |
| 1. The student clicks module link.  3. The student select and click “download” link.  5. End use case. | 2. The system displays the module option.  4. The system down loads module |
| Alternative course of action | If there is no module uploaded before, the system display “there is no module in the system” message. | |

Table 4- 5 : Use case Description of view Result

|  |  |  |
| --- | --- | --- |
| Use case number | UC 05 | |
| Use case name | View result | |
| Actor | Students. | |
| Description | The user looks result added by the instructor in the system. | |
| Precondition | The user must have user name and password and the result must be added before. | |
| Post condition | The user access and know the result. | |
| Basic course of action | User action | System response |
| 1. The user login to the system.  3. The user click on view result links.  5. the user view exam result  6. End use case. | 2. The system displays the student home page.  4. The system displays available result that is added by the instructor. |
| Alternative course of action | If result is not added, the system displays the message “there is no result available”. | |

Table 4- 6 :Use case Description of Assign Instructor

|  |  |  |
| --- | --- | --- |
| Use case number | UC 08 | |
| Use case name | Assign instructor | |
| Actor | Admin | |
| Description | assigning instructor to class in the system | |
| Precondition | admin must have user name and password and instructor must register already | |
| Post condition | The instructor successfully assign in the system | |
| Basic course of action | User action | System response |
| 1.The admin login to the system  3. The user clicks assign link.  5.The user select the instructor and click assign button  7. End use case. | 2. The system displays the admin home page.  4. The system displays instructor list.  6. The system assign instructor |
| Alternative course of action | If user click assign link without select, the system display “no selection” message then return back to step 5 basic course of action. | |

Table 4- 7 : Use case Description of Add course

|  |  |  |
| --- | --- | --- |
| Use case number | UC 10 | |
| Use case name | Add course | |
| Actor | Admin | |
| Description | Adding course in the system | |
| Precondition | admin must have user name and password | |
| Post condition | The course successfully added in the system by admin | |
| Basic course of action | User action | System response |
| 1.The admin login to the system  3. The user clicks add course link.  5. The user fills the form and click add button.  7. End use case. | 2. The system displays the admin home page.  4. The system displays course registration form  6. The system adds course |
| Alternative course of action | If the user miss some information to fill, error message display and turn back to step 5 basic course of action | |

Table 4- 8 :Use case Description of deactivate user account

|  |  |  |
| --- | --- | --- |
| Use case number | UC 14 | |
| Use case name | Deactivate user account | |
| Actor | System administrator | |
| Description | The system administrator block accounts of the , students and instructors if there is any inconvenience. | |
| Precondition | The administrator must login to the system. | |
| Post condition | Protect user account from unauthorized access | |
| Basic course of action | User action | System response |
| 1.The system admin clicks manage account link  2. The admin clicks on view users account link  4. The system admin enters keywords on search box and clicks on search button  6. The system admin check on block account.  8. End use case. | 3. The system displays search box  5. The system displays the search result.  7. The system displays message. |
| Alternative course of action | If the search result is empty or if the user enters incorrect keyword in the search box the systems lets the user to try again and back to step 4 in basic course of action. | |

Table 4- 9 :Use case Description of activate user account

|  |  |  |
| --- | --- | --- |
| Use case number | UC 15 | |
| Use case name | Activate user account | |
| Actor | Admin | |
| Description | The system administrator unblocks or activate user accounts of the students and instructors | |
| Precondition | The administrator must login to the system and the account of user deactivate before. | |
| Post condition | Successfully activate user account | |
| Basic course of action | User action | System response |
| 1.The system admin clicks manage account link  3. The system admin select deactivate user account and click activate button  5. End use case. | 2.The system displays user account  4. The system activates user account |
| Alternative course of action | If connection is fail, try again. | |

Table 4- 10 :Use case Description of Add student result

|  |  |  |
| --- | --- | --- |
| Use case number | UC 16 | |
| Use case name | Add student result | |
| Actor | Instructor | |
| Description | Add course result of the student in the system | |
| Precondition | The instructor must have user name and password | |
| Post condition | The instructor successfully add course result of the student in the system | |
| Basic course of action | User action | System response |
| 1.The instructor login to the system  3. The instructor clicks add result link.  5. The instructor fills the course information and fills each student result, then click “add” button.  7. End use case. | 2. The system displays the instructor home page.  4. The system displays the result form.  6. The system add result to student |
| Alternative course of action | If user misses some information, system generates error message and back to step 5 to try again. | |

Table 4- 11 :Use case Description of upload learning material

|  |  |  |
| --- | --- | --- |
| Use case number | UC 17 | |
| Use case name | upload learning material | |
| Actor | Instructor | |
| Description | uploading learning materials for the student | |
| Precondition | The instructor must have user name and password. | |
| Post condition | The instructor successfully uploading learning materials. | |
| Basic course of action | User action | System response |
| 1.The instructor login to the system  3. The user clicks module upload link.  5. The user fills the information of learning material and click “upload” button.  7. End use case. | 2. The system displays the instructor home page.  4. The system displays the form.  6. The system up loads modules and other learning materials |
| Alternative course of action | If user miss some information about learning materials, the system displays error messages and back to step 5 to try again | |

Table 4- 12 : Use case Description of add department

|  |  |  |
| --- | --- | --- |
| Use case number | UC 18 | |
| Use case name | Add department | |
| Actor | Admin | |
| Description | Adding department in the system | |
| Precondition | Admin must have user name and password | |
| Post condition | The department successfully added in the system by admin | |
| Basic course of action | User action | System response |
| 1.The admin login to the system  3. The user clicks add department link.  5. The user fills the form and click add button.  7. End use case. | 2. The system displays the admin home page.  4.The system displays department registration form  6. The system adds department |
| Alternative course of action | If the user miss some information to fill, error message display and turn back to step 5 basic course of action | |

### **4.2. Object Model**

Object Model is a collection of objects or classes through which a program can examine and manipulate some specific parts of its world. In other words, the object-oriented interface to some service or system. Such an interface is said to be the object model of the represented service or system.

#### **4.2.1.Class Diagram**

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also constructing executable code of the software application. This class diagram also describes the attributes and operations of a class and also the constraints imposed in the system. The class diagram is widely used in the modeling of object oriented systems because they are The only UML diagrams, which can be mapped directly with object-oriented languages. We are developing object oriented base system, so we are going to use class diagram to model the static view of our system

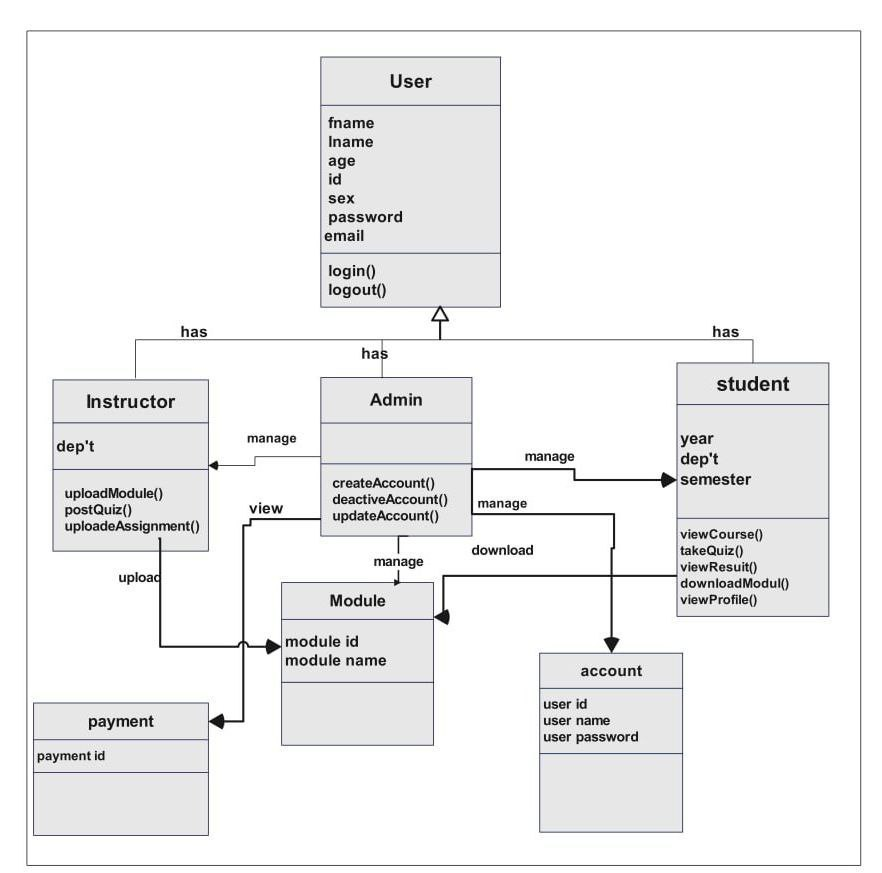


Figure 4- 2 :Class Diagram

#### **4.2.2 Data Dictionary**

Data Dictionary is a collection of names, definitions, and attributes about data elements that are being used or captured in a database, information system, or part of a research project. It describes the meanings and purposes of data elements within the context of a project, and provides guidance on

interpretation, accepted meanings and representation.

**Table 4- 17:Data Dictionary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | **Caption** | **Example** |  | **Data type** | **Constraints** |
| user\_ID | User Identifier | NSR/0753/13 |  | varchar(100) | Primary key |
| firstName | First Name | Endalew |  | varchar(100) | Not null |
| lastName | Last Name | Shumet |  | varchar(100) | Not null |
| gender | Gender | Male |  | varchar(100) | Not null |
| email | Email | [Endalews92@gmail.com](mailto:Endalews92@gmail.com) |  | varchar(100) | Not null |
| address | Address | Wolkite |  | varchar(100) | Not null |
| password | Password | Endex@123 |  | varchar(100) | Not null |

### **4.3. Dynamic Model**

Dynamic models are generally models that contain or depend upon an element of time, especially allowing for interactions between variables over time. A separate idea with the same name is models that are updated over time with new data.

#### **4.3.1. Sequence Diagram**

Sequence diagrams are used to depict graphically how objects interact with each other via

messages in the execution of a use case or operation. They illustrate how the operations are

performed between objects and in what sequence. A virtual course delivery system sequence

diagram is an interaction diagram that shows how processes operate with one another and in

what order. It is a construct of a Message Sequence chart, this sequence diagram shows object

inter actions arranged in time sequence

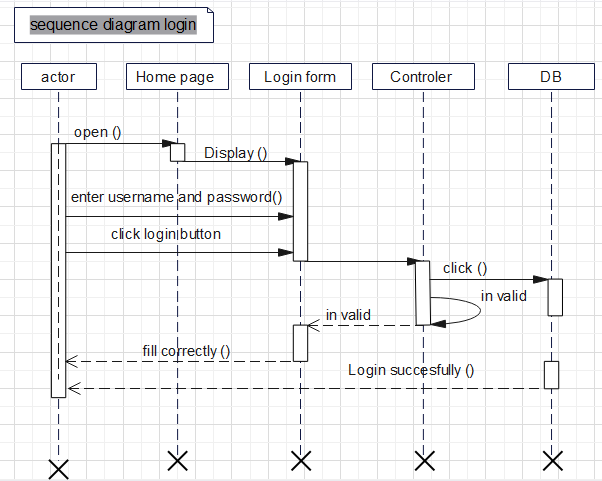
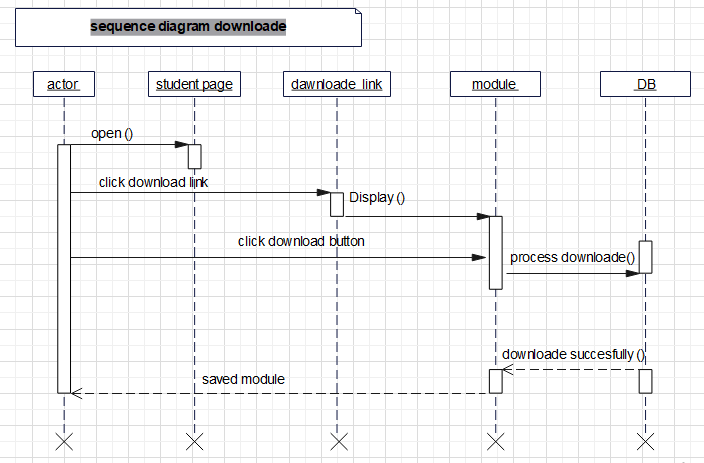


Figure 4- 3 :Sequence diagram login

Figure 4- 4: Sequence diagram download

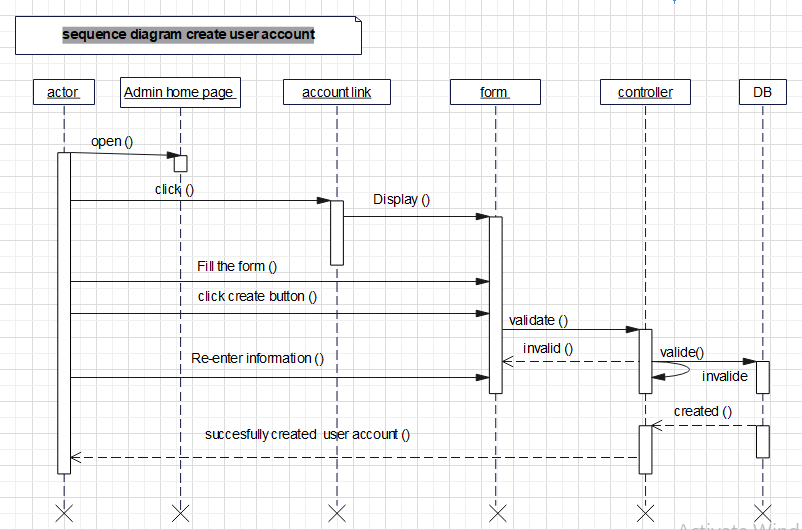


Figure 4- 5 :Sequence diagram create account

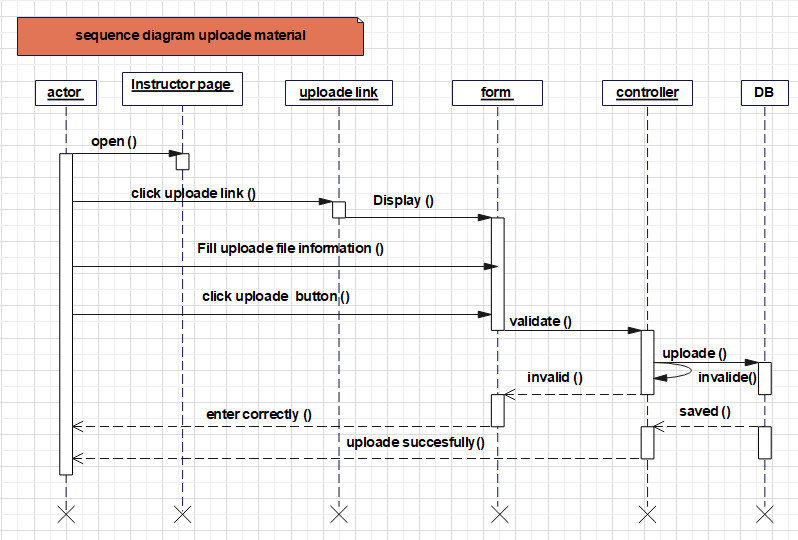


Figure 4- 6 :Sequence diagram upload learning material

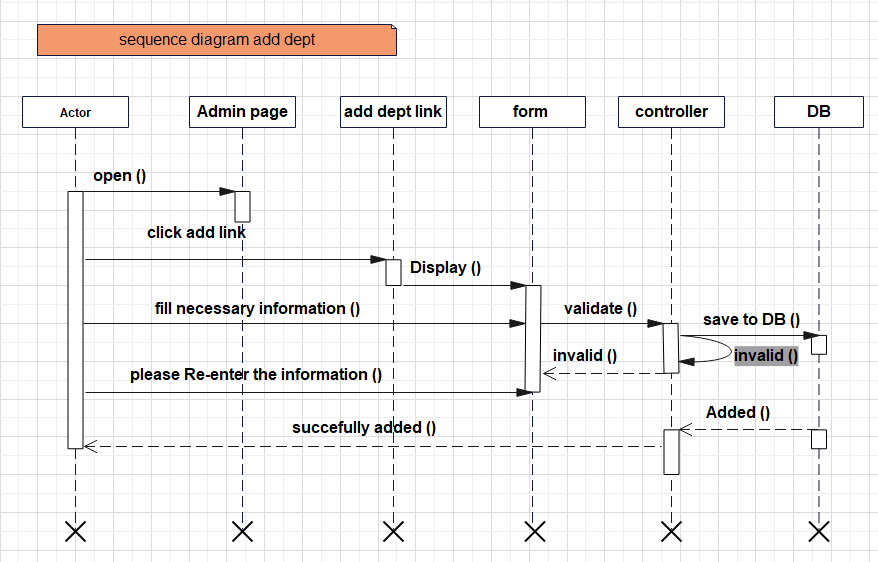
****

Figure 4- 7 :Sequence diagram add dept.

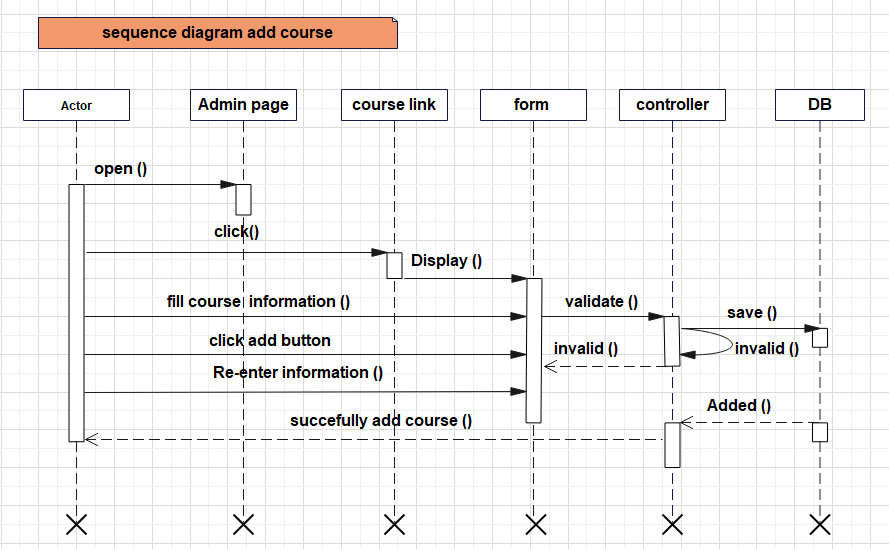
****

Figure 4- 8 :Sequence diagram add course

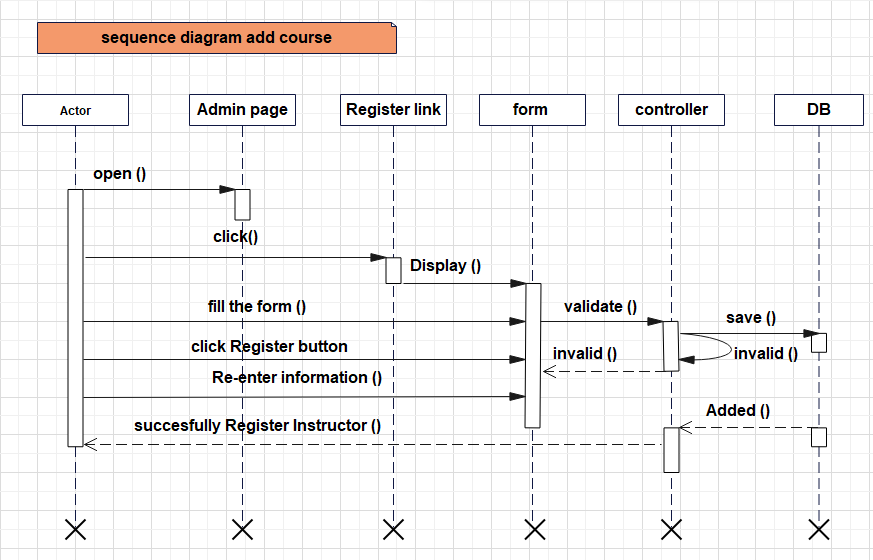
****

Figure 4- 9 :Sequence diagram register Instructor

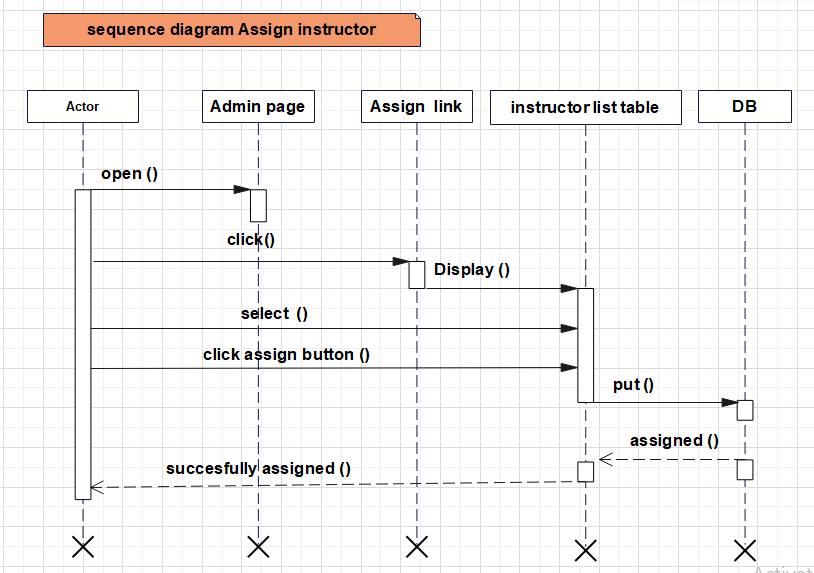
****

Figure 4- 10 : Sequence diagram Assign Instructor

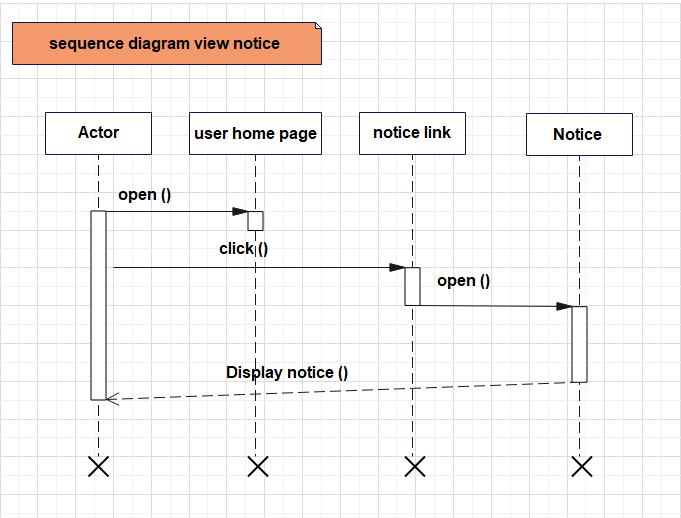
****

Figure 4- 11 :Sequence diagram view notice

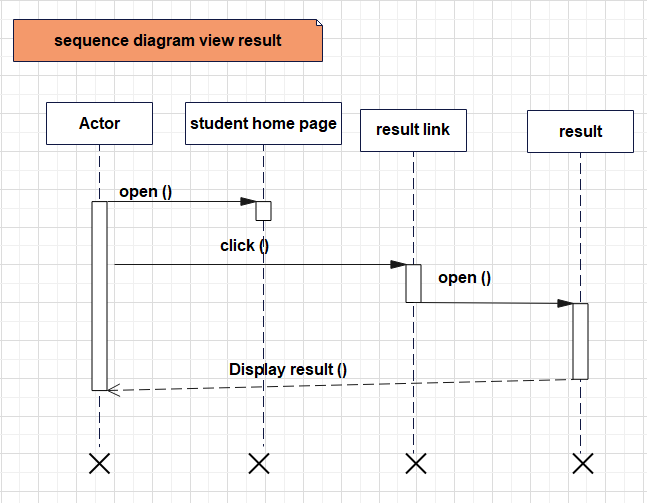
****

Figure 4- 12 : Sequence diagram view result

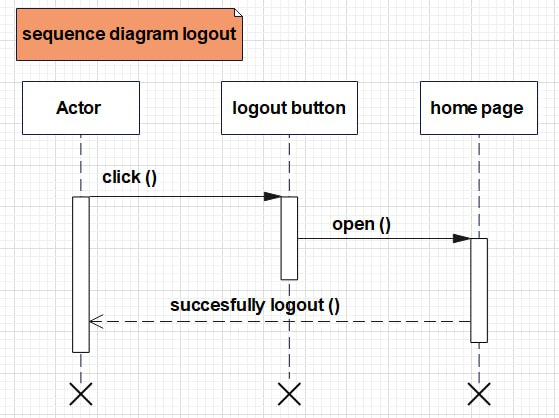


Figure 4- 13 : Sequence diagram logout

#### **4.3.2. Activity Diagrams**

An activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all types of flow control by using different elements such as forks, joins, etc.The basic purposes of activity diagrams are to capture the dynamic behavior of the system.Activity diagrams are used to show message flow from one activity to another.

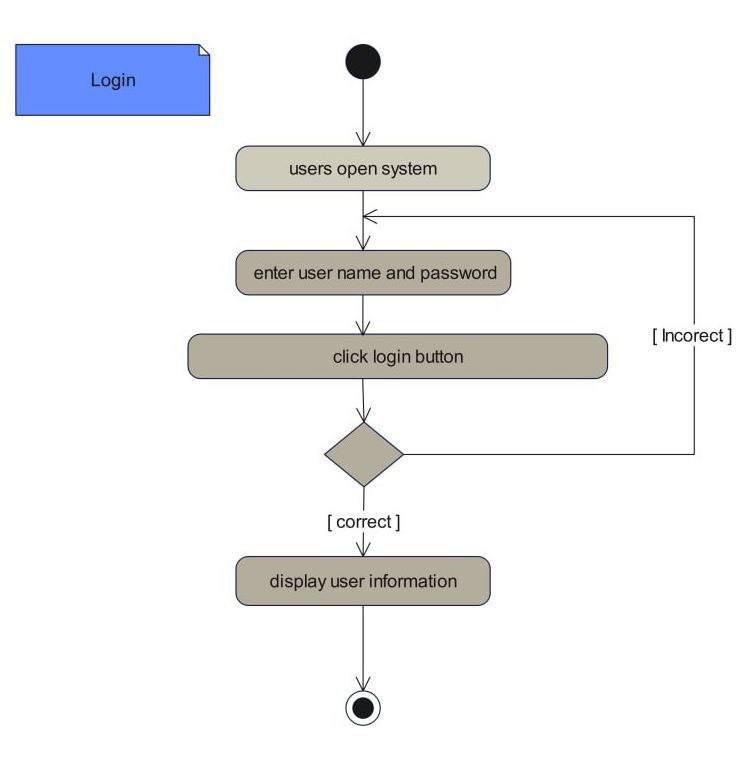


Figure 4- 13: Activity Diagram login

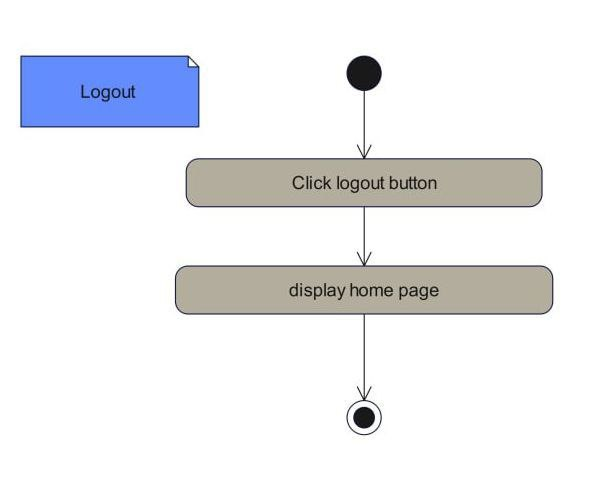


Figure 4- 14 :Activity Diagram logout

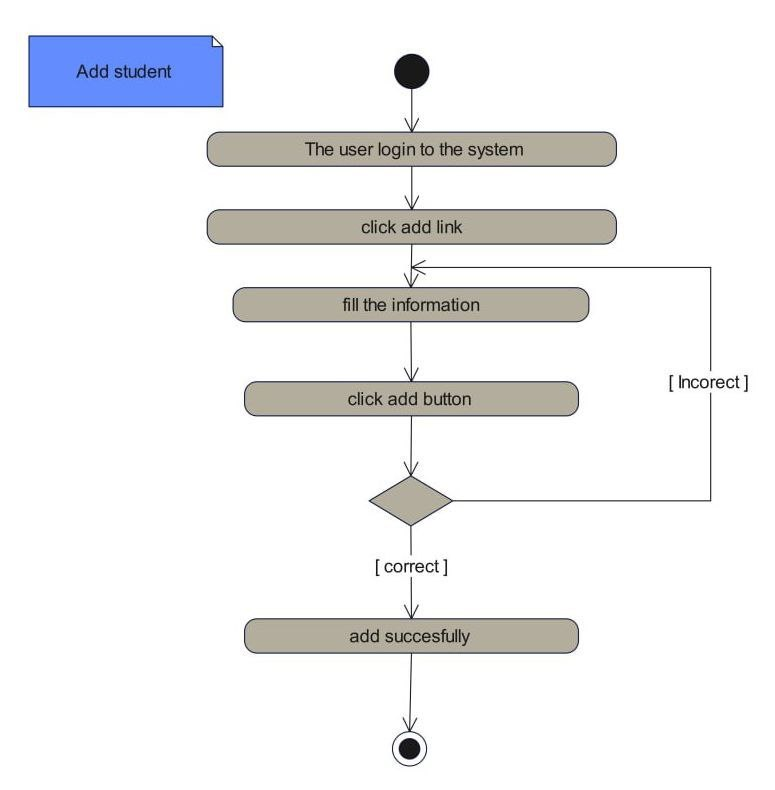


Figure 4- 15 : Activity Diagram add student

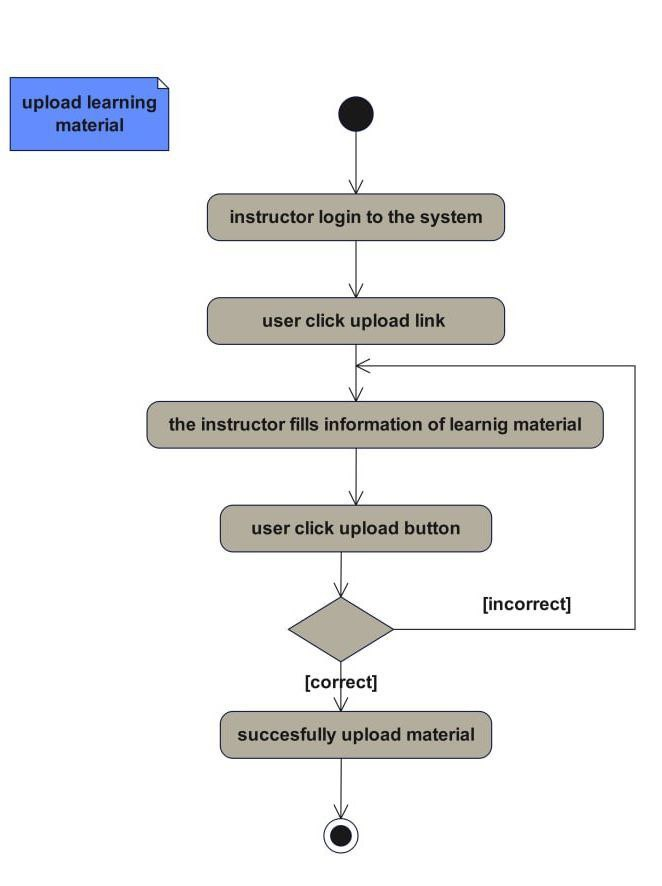


Figure 4- 16 : Activity Diagram upload module

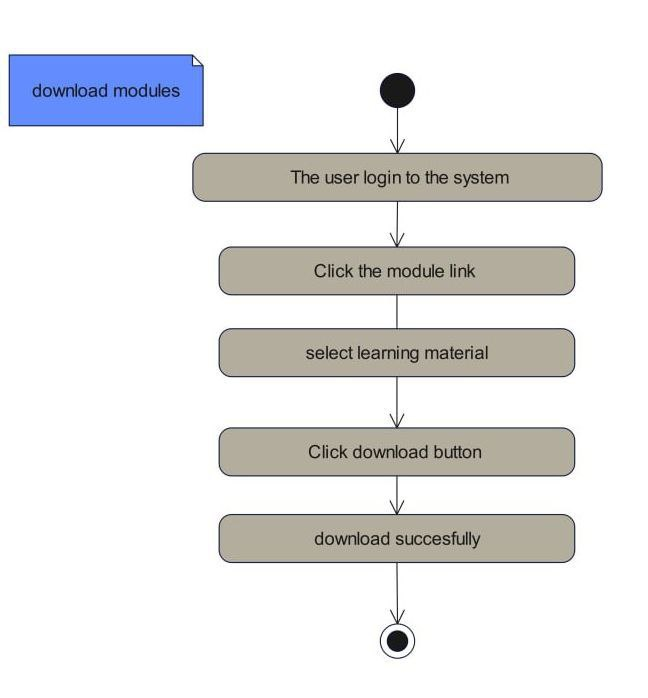


Figure 4- 17: Activity Diagram download module

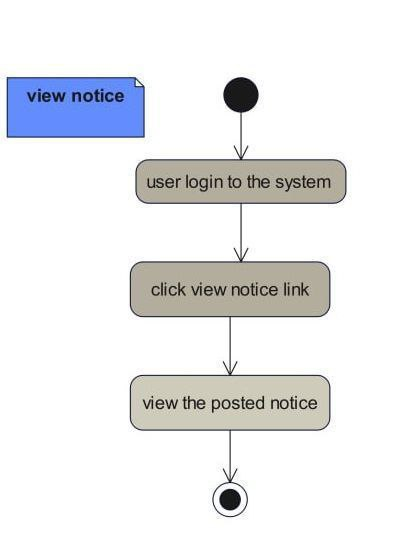


Figure 4- 18 : Activity Diagram view notice

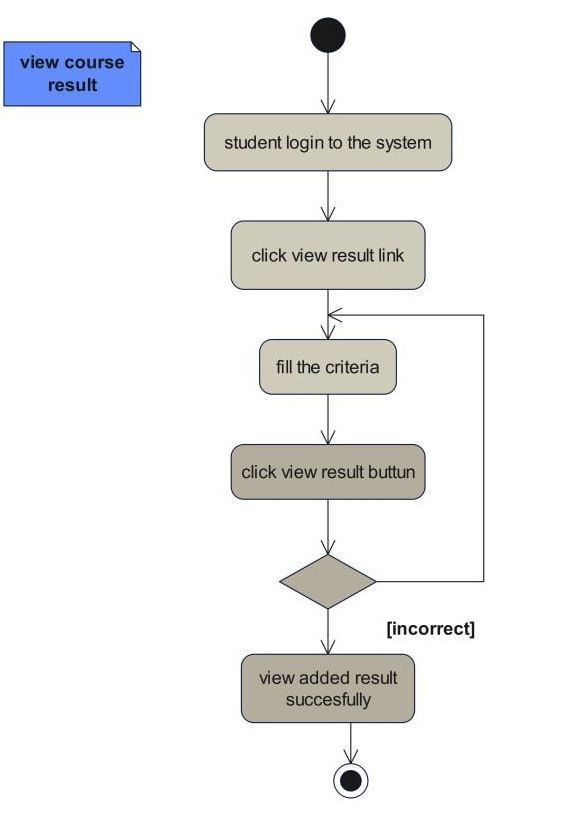


Figure 4- 19 : Activity Diagram view course result

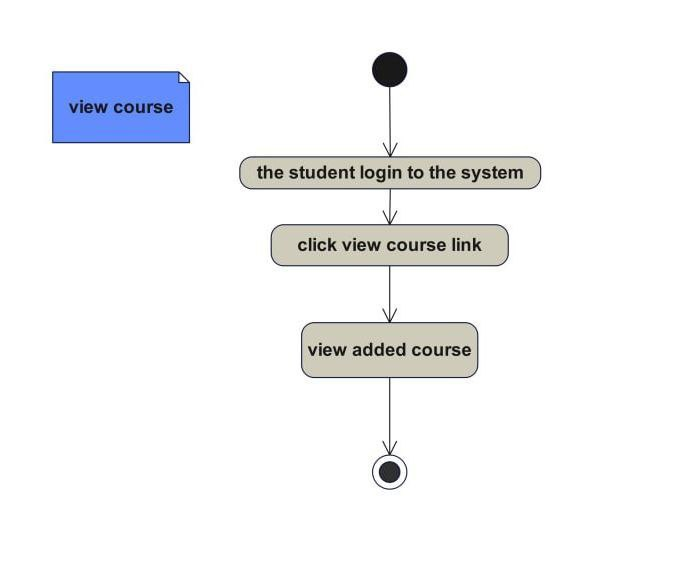


Figure 4- 20:Activity Diagram view course

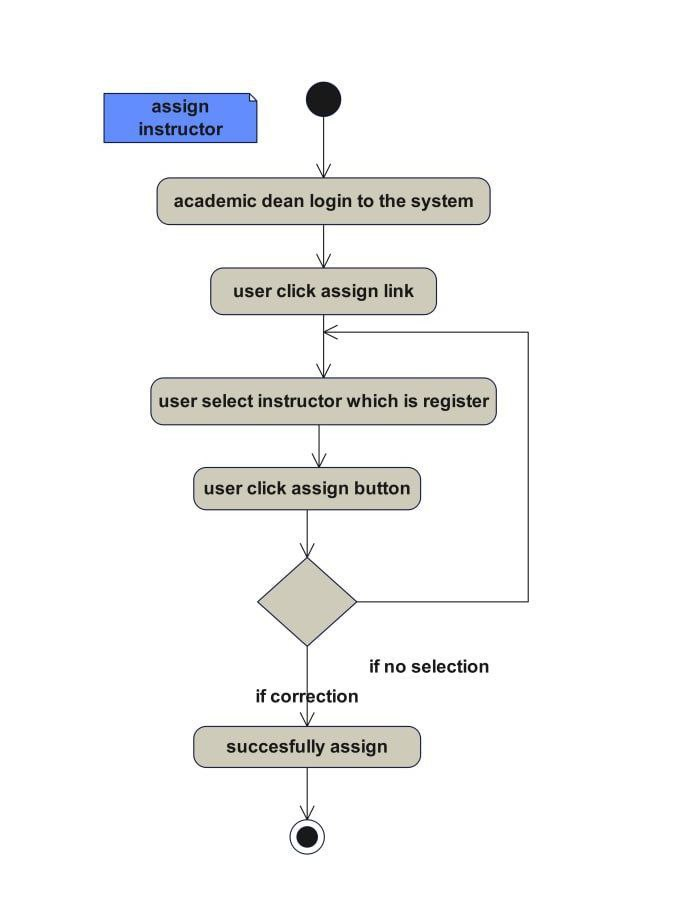


Figure 4- 21 :Activity Diagram Assign instructor

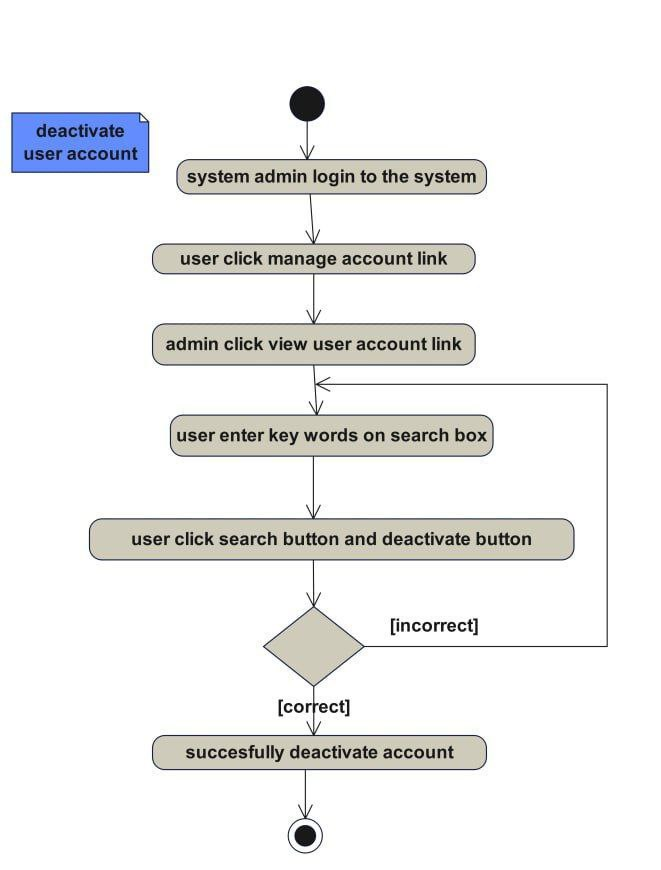


Figure 4- 22 : Activity Diagram deactive account

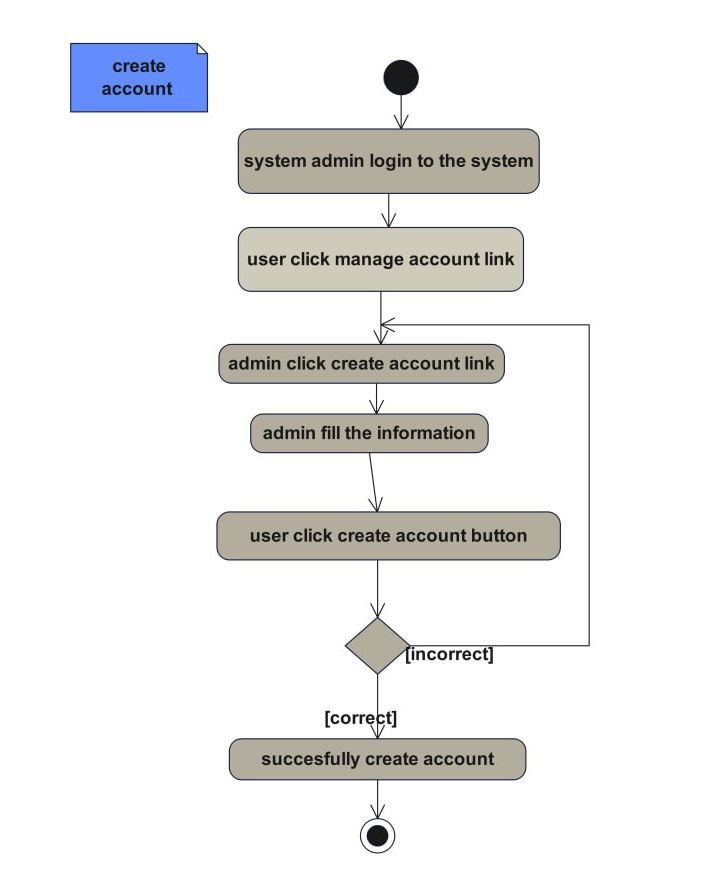


Figure 4- 23 : Activity Diagram create account

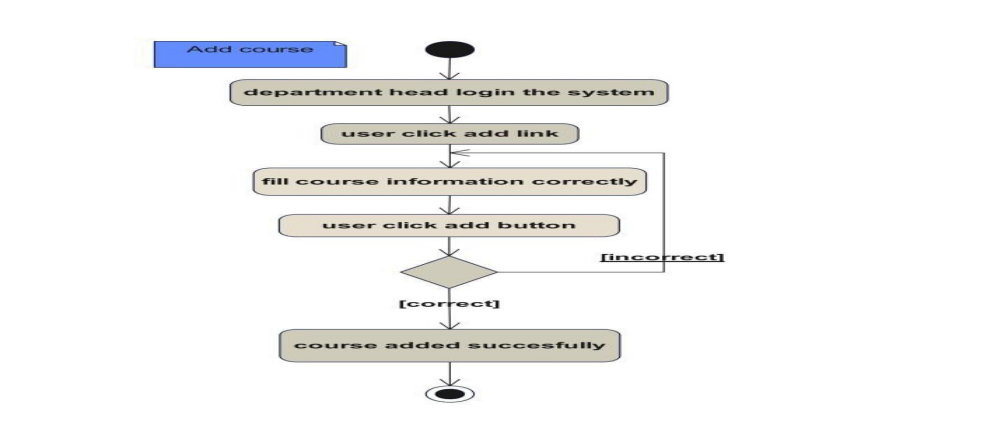


Figure 4- 24: Activity Diagram add course

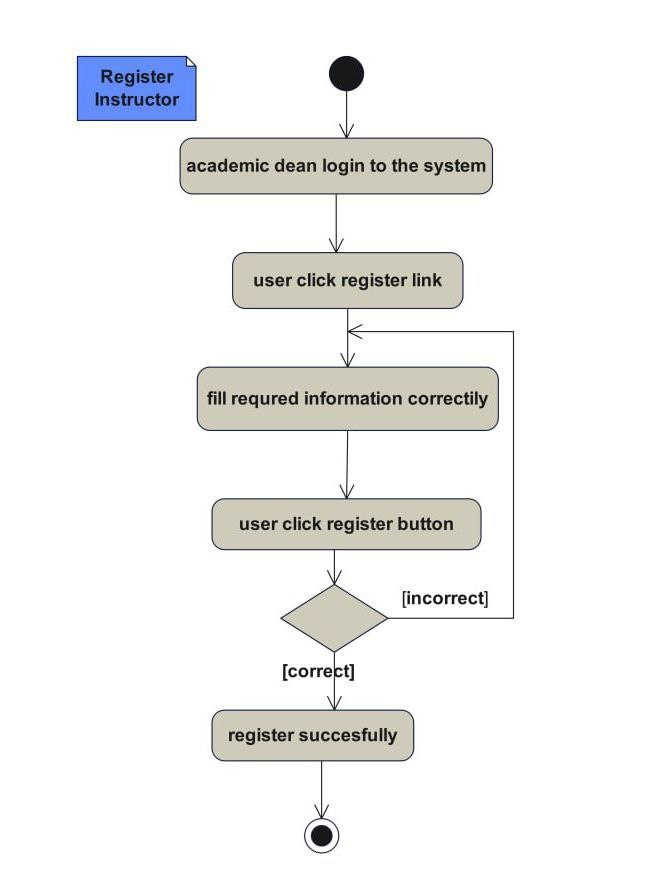


Figure 4- 25 : Activity Diagram Register Instructor



**Figure 4- 26 : Activity Diagram payment**

#### **4.3.3 State Chart Diagram**

state chart diagram defines different states of an object during its lifetime. These states can change based on events that trigger the object. The diagram is particularly useful for modeling reactive systems, which respond to internal or external events.

state chart diagram represent the condition of an object when a specific event occurs. The main purpose of a state chart diagram is to model an object's entire lifespan, from creation to termination.

The Following are the main purposes of using State chart diagrams

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

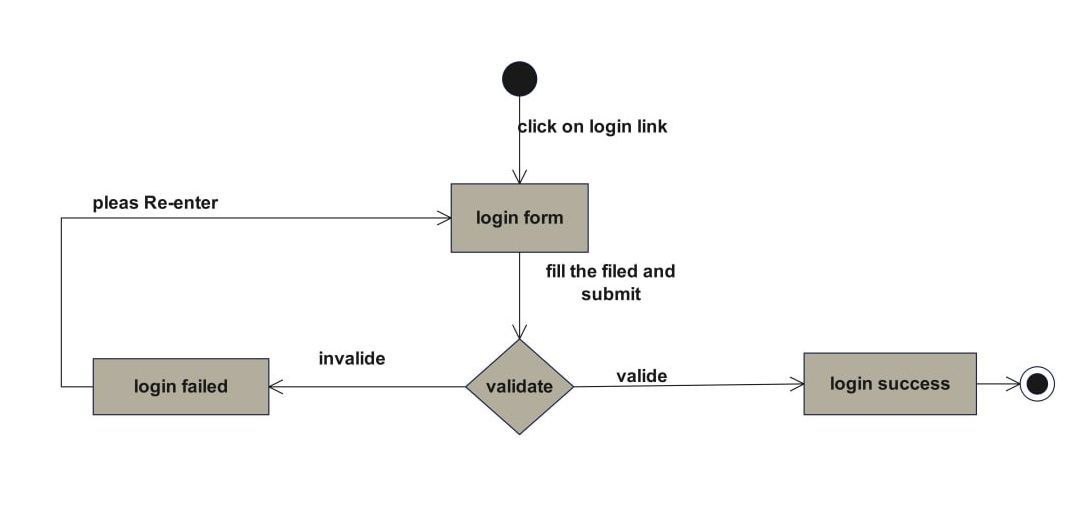


Figure 4- 26 :State chart diagram for login

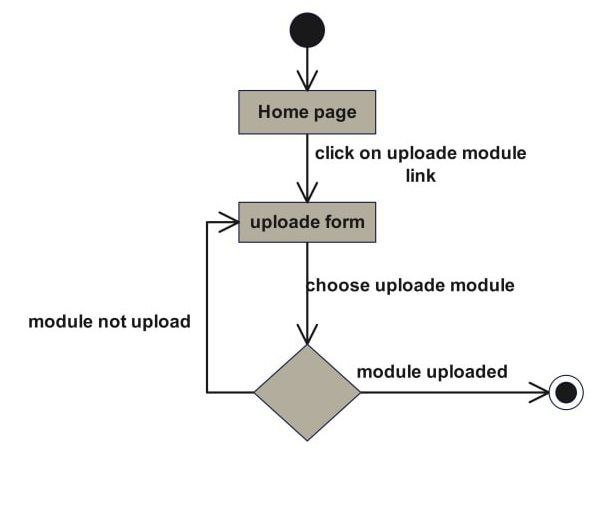


Figure 4- 27: State chart diagram for upload module

# **CHAPTER Five**

## 5. SYSTEM DESIGN

We have identified the functional and non-functional requirements of the system and produced an analysis model in the previous chapter. Now we will discuss in this chapter: The design goal of our system, System architecture, System decomposition, deployment, etc. This project is designed in a manner that solves the problems of Great Vision College by minimizing the workload that appears on the employees, especially regarding sharing files and managing users, and all the problems we identified in the system analysis phase of this project.

In this phase, our group project will try to illustrate about:

* Design goal of the system
* Proposed system architecture
* Subsystem Decomposition and Description
* Persistent Data Management
* Hardware/Software Mapping
* Detailed Class Diagram
* Algorithm design and
* User Interface design

### **5.1 Design Goals**

The design goals are derived from non-functional requirements, which means non-functional requirements are the description of the feature characteristics and attributes of the system, as well as any constraints that may limit the boundary of the proposed solution. The objectives of design are to model the system with high quality. They describe the qualities of the system. These goals consider the following criteria.

### **5.2 Current System Architecture**

The current system is not designed and there is no software architecture for the current software of this project is not existed and the users are using the manual way to get the information.

#### **5.2.1 Performance Consideration**

The system will have good performance as much as possible, this will be attained via easily loadable information and fast response time in searching, updating, and viewing information because we will use good algorithms and minimize lines of code to perform specific tasks (functions).

* Load Time: The system will be loading information within a second.
* Throughput: Depending on the performance of our computers that run the system, the system performs many operations at a given time.
* Response Time: This system gives a response to the user according to the delivery of messages and notifications, reaching within a second. The system will support multiple users at a time, and it works very well with short response times.

#### **5.2.2 Dependability**

This system should achieve the following dependability characteristics in order to resist a crash and be available and reliable:

* Security: This system is secured by preventing unauthorized users from accessing the database system.
* Reliability: This system is reliable by providing the correct information and giving a response correctly and accurately for every query from authorized users.

#### **5.2.3 Maintenance**

The system should be maintainable if there is a failure in a system or when the system needs modification. That means we will develop this system by dividing the system into different modules which are loosely coupled and highly cohesive. If the system needs maintenance, it should meet the following maintenance criteria:

* Extensibility: If it is needed to add new functionality to the system, this must be achieved by only making a separate page and integrating this page with the existing system.
* Modifiability: If the system needs some functionality modification, the modification is done to that function (the function to be modified) or page (the page to be modified) without affecting the overall system organization.

#### **5.2.4 End User**

The user interface of the system should prevent users from issuing commands in the wrong order. That means whenever users of the system want to insert unnecessary commands toward the next function, the user interface prevents them from performing such actions.

### **5.3 Proposed System Architecture**

The proposed system architecture aims to effectively address the multifaceted requirements for seamless operation of the platform. It encompasses crucial components across distinct tiers to cater to user roles, streamline interactions, and manage the intricacies of products, users, and transactions.

User Management stands pivotal, ensuring the presence of distinct user accounts for Buyers, Sellers, and Administrators. This enables tailored access to functionalities pertinent to their respective roles. The Seller Registration process integrates identity verification and product submission during registration, while Admin Privileges empower administrators with the authority to verify submissions, oversee other admin accounts, and manage geographic locations.

Product Management is another essential facet, comprising a dedicated Seller Interface that facilitates product uploading, description, and pricing. Admin Screening ensures initial scrutiny for authenticity, quality, and guideline adherence, followed by making approved products available for purchase within the platform.

Buyer Experience is streamlined to provide an intuitive and seamless journey. It includes functionalities for browsing categorized product listings, selecting desired items, and facilitating secure transactions via the integrated payment gateway. Additionally, the system manages post- transaction processes, ensuring proper payment to the seller and smooth product delivery to the buyer.

The proposed architecture adopts a Three-Tier Model, consisting of the following tiers:

* **Presentation Tier (Client):**
* Housing user interfaces for Buyers, Sellers, and Admins.
* Featuring dedicated spaces for sellers to manage products and enabling buyers to browse and purchase items.
* **Application Tier:**
* Encompassing the core business logic handling user-related processes, product screenings, and transactional procedures.
* Facilitating seller registrations, admin verifications, and automated adjustments in pricing based on product variations.
* **Data Tier:**
* Hosting the database that securely stores crucial information, including user accounts, product details, transaction records, and administrative configurations.
* Ensuring the safekeeping of user credentials, product metadata, transaction histories, and administrative settings.

By adopting this proposed architecture, the e-commerce platform can effectively meet the requirements for seamless operation. It provides a robust foundation for user management, product

management, and a streamlined buyer experience. The adoption of a Three-Tier Model ensures separation of concerns and facilitates scalability, security, and maintainability of the system.

#### **5.3.1. Subsystem Decomposition and Description**

Subsystem decomposition helps to reduce the complexity of the system. The subsystem integrates the classes that this system contains and the operation performed in the class. The following are the subsystems of our system.

* **Manage account subsystem**
* Create account
* Update account
* Activate account
* Deactivate account
* Change password
* **Communication Subsystem**
* Post announcement
* View comment
* Apply to register
* Give Feedback
* Upload module
* **User Subsystem**
* Add user
* View user
* Delete user
* **Material Subsystem**
* Upload module and assignment
* Download module and assignment
* **Work Subsystem**
* Prepare grade report
* Approve result
* Submit result
* Generate report
* View result

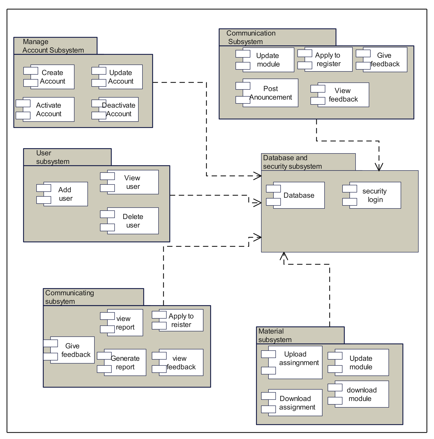


Figure 5- 2 : Subsystem Decomposition Diagram.

#### **5.3.2. Hardware/Software Mapping**

One of the major tasks in system design deals with hardware/software mapping which deals with which components would be part in which hardware and so on.

* User interface and processing management will be deployed on the client machine.
* Web-based distance education system will present the Graphical User Interface (GUI), which is used for display user information.
* Web-based distance education system use existing database, which deployed on the server.
* The data collection module deployed on any computer.

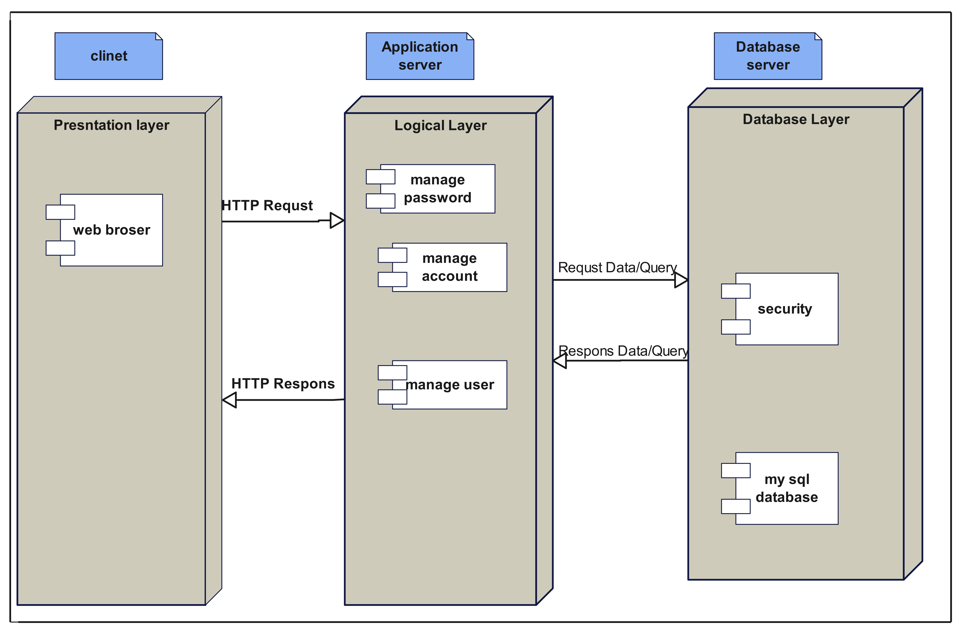


Figure 5- 3: Development diagram Hardware/Software Mapping

#### **5.3.3. Detailed Class Diagram**

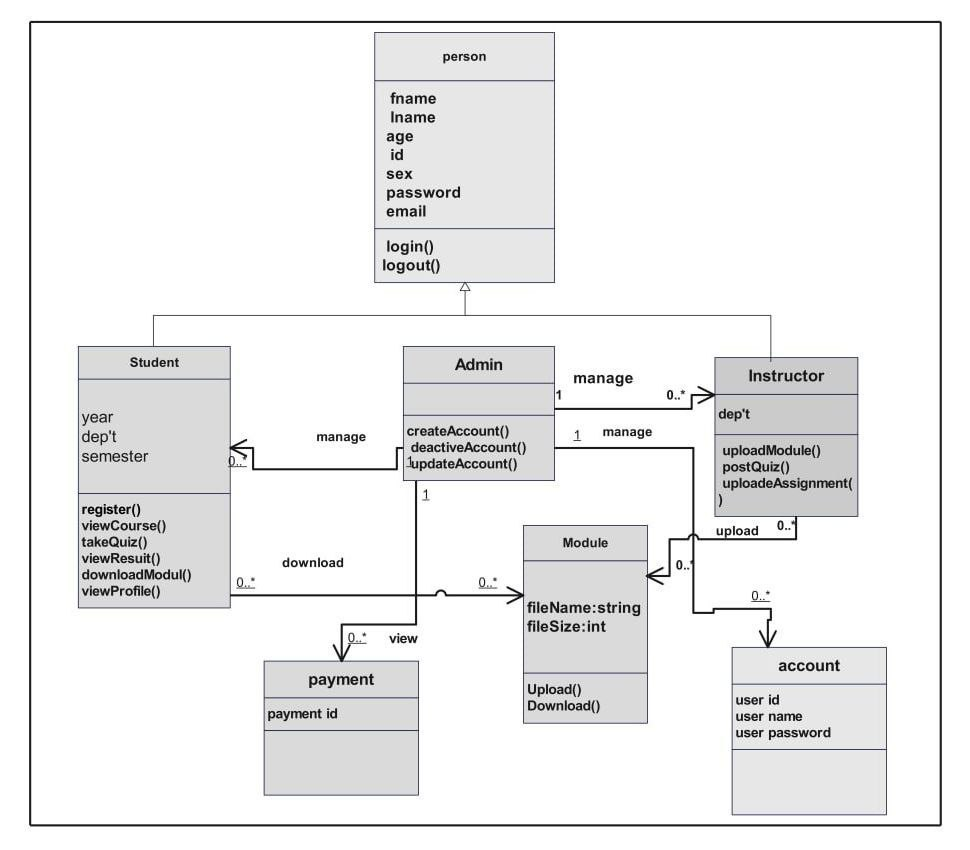


Figure 5- 4: Detailed Class Diagram

#### **5.3.4. Persistent Data Management**

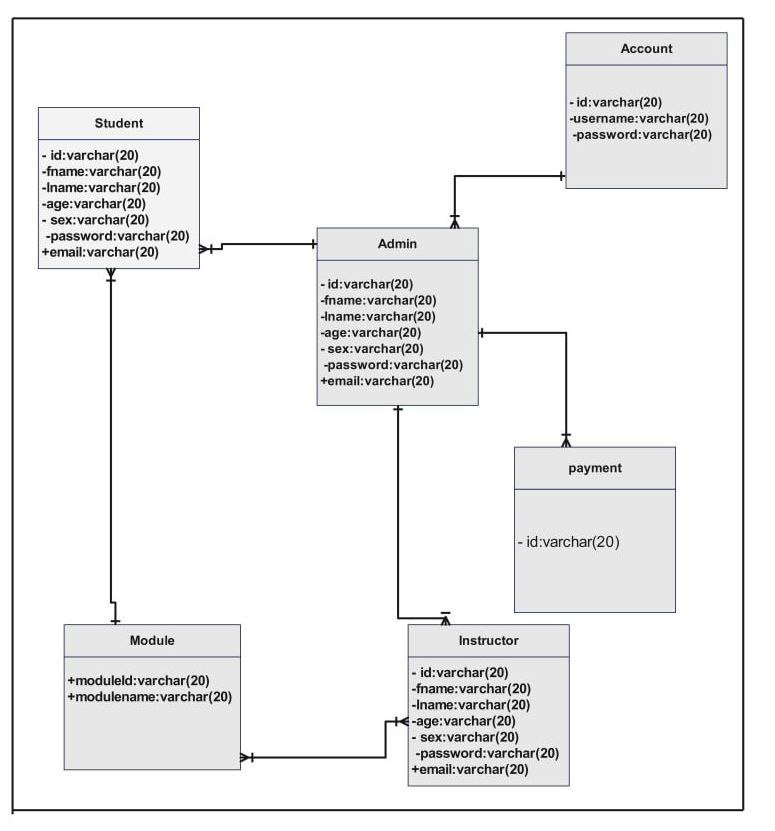


Figure 5- 5 : Persistent Data Management

#### **5.4.5. Access Control and Security**

|  |  |
| --- | --- |
| Actor | Operations |
| Administrator | * Login * Manage Account * Create account * Update Account * Assign instructor * Add dep’t * Delete dep’ * Deactivate Account * Logout |
| Student | * Login * Apply to register * View announcement * View result * Download module * Download assignment * Upload assignment answer * Logout |
| Instructor | * Login * Upload assignment * Download assignment * Change password * Prepare grade report * View student result * Logout |

### **5.5. Packages**

package is a piece of reusable code that can be dropped into any application and be used without any tinkering to add functionality to that code.

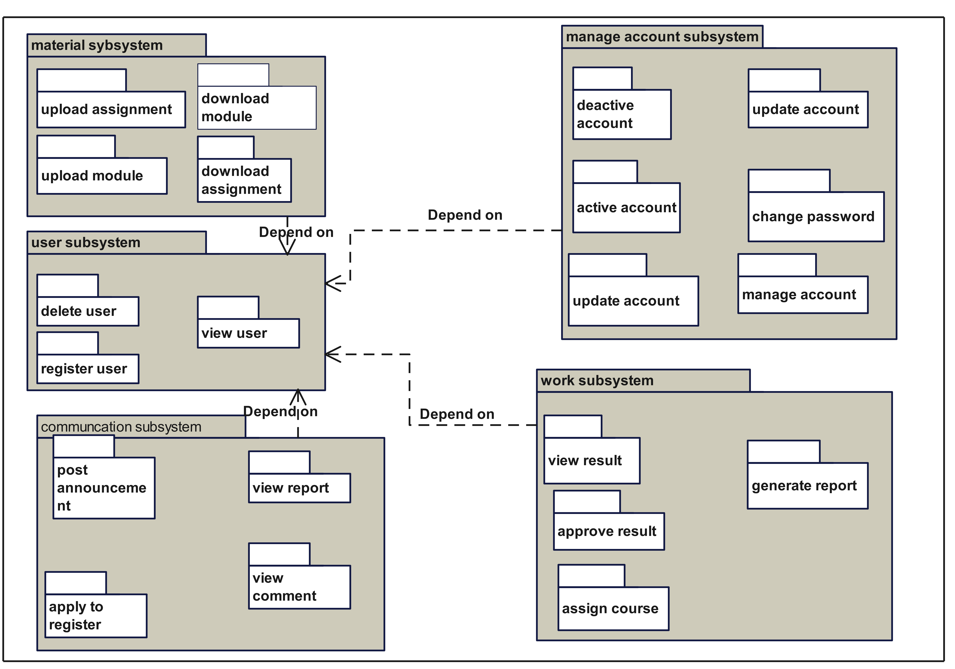


Figure 5- 10 : Packages

### **5.6. Algorithm Design**

* **Login**
* Page displayed on the client machine.
* The user clicks on the login link.
* The system displays the login page or login page.
* The user enters a username and password in the available space.
* If the username and passwords are correct, then
* Login successfully
* Else the user username and password is incorrect,
* The system displays an error message and re displays the login page

### **5.7. User Interface Design**

User interface design is the overall process of designing how a user will be able to interact with a system and this is the design of the new system interface. In this system users will communicate with it through the user interface in this section we show the home page, login and create account page. The home page appears as the site on which the system is deployed is opened. This form contains some links which lead it to the concerned page, and if the user has an account he/she will directly go to concerned page by entering their username and password.

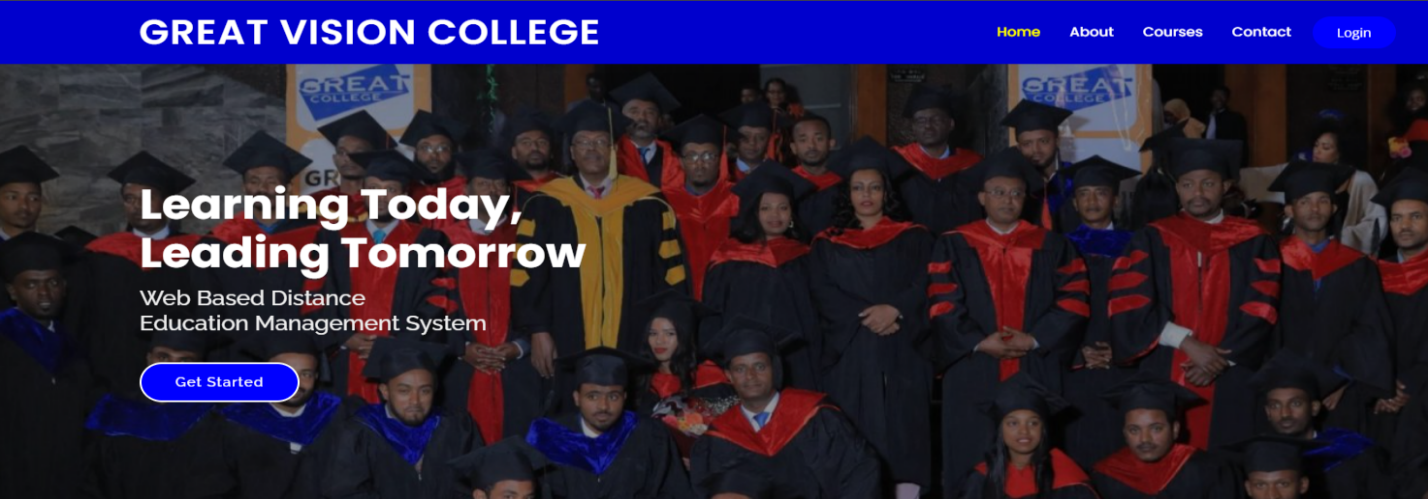


Figure 5- 11 : User Interface for Home Page

**Login page:** This page displayed, when user's clicks on login link. As one user click login there is chose of account type such as Admin, Instructor, Registrar and Student. All have their own password and user name. Those forms will not accessible by other persons except for those who have privilege.

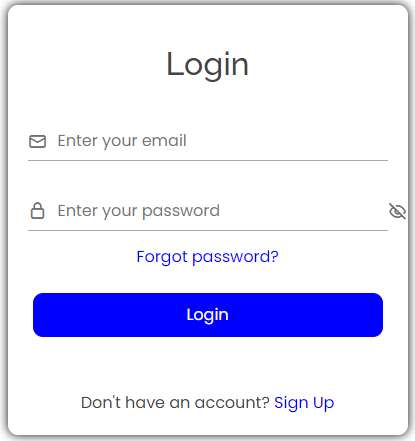


Figure 5- 12: User Interface for Login Page

# **CHAPTER SIX**

## IMPLEMENTATION AND TESTING

In this chapter, we explore the implementation of database, detailed class diagram, application server, and application security a reality. These are important parts of the project development phase where we turn project ideas into actual results. During implementation, our main job is coding. We work hard to bring the ideas from the project document into a real project. As we work on implementation and even after the project is up and running, we take the plans for how everything should look and work and turn them into computer code. We test this code thoroughly to fix any mistakes. After that, we set up the system and make sure users are ready to start using it. They switch completely to the new system for their tasks. This whole process involves managers keeping an eye on things like coding, testing, and installation.

### **Implementation of database**

* We have chosen MySQL database for effective management of databases because it is free-to-use open-source database a stable, reliable and powerful solution with advanced features including.
* It's secure: MySQL includes find dam security layers that protect sensitive data from intruders.
* It’s easy to use with only a few simple SQL statements we can build and practice with MySQL.
* Its fast: MySQL will offer all of the features required by most database development with its required speed.
* It's scalable: MySQL can handle almost any amount of data.
* MySQL runs on many operating systems, including Windows and Linux.
* We have coded the implementation of database model in HTML language in bootstrap framework as follow :

CREATE TABLE `notification\_read` (

  `notification\_read\_id` int(11) NOT NULL,

  `student\_id` int(11) NOT NULL,

  `student\_read` varchar(50) NOT NULL,

  `notification\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `notification\_read\_teacher` (

  `notification\_read\_teacher\_id` int(11) NOT NULL,

  `teacher\_id` int(11) NOT NULL,

  `student\_read` varchar(100) NOT NULL,

  `notification\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `question\_type` (

  `question\_type\_id` int(11) NOT NULL,

  `question\_type` varchar(150) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE `quiz` (

  `quiz\_id` int(11) NOT NULL,

  `quiz\_title` varchar(50) NOT NULL,

  `quiz\_description` varchar(100) NOT NULL,

  `date\_added` varchar(100) NOT NULL,

  `teacher\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `quiz\_question` (

  `quiz\_question\_id` int(11) NOT NULL,

  `quiz\_id` int(11) NOT NULL,

  `question\_text` varchar(100) NOT NULL,

  `question\_type\_id` int(11) NOT NULL,

  `points` int(11) NOT NULL,

  `date\_added` varchar(100) NOT NULL,

  `answer` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `result` (

  `id` int(11) NOT NULL,

  `st\_id` int(11) NOT NULL,

  `marks` int(5) NOT NULL,

  `sub` varchar(50) NOT NULL,

  `semester` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `school\_year` (

  `school\_year\_id` int(11) NOT NULL,

  `school\_year` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `student` (

  `student\_id` int(11) NOT NULL,

  `firstname` varchar(100) NOT NULL,

  `lastname` varchar(100) NOT NULL,

  `class\_id` int(11) NOT NULL,

  `username` varchar(100) NOT NULL,

  `email` varchar(100) NOT NULL,

  `password` varchar(100) NOT NULL,

  `location` varchar(100) NOT NULL,

  `code` varchar(100) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE `student\_assignment` (

  `student\_assignment\_id` int(11) NOT NULL,

  `assignment\_id` int(11) NOT NULL,

  `floc` varchar(100) NOT NULL,

  `assignment\_fdatein` varchar(50) NOT NULL,

  `fdesc` varchar(100) NOT NULL,

  `fname` varchar(50) NOT NULL,

  `student\_id` int(11) NOT NULL,

  `grade` varchar(5) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `student\_class\_quiz` (

  `student\_class\_quiz\_id` int(11) NOT NULL,

  `class\_quiz\_id` int(11) NOT NULL,

  `student\_id` int(11) NOT NULL,

  `student\_quiz\_time` varchar(100) NOT NULL,

  `grade` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `subject` (

  `subject\_id` int(11) NOT NULL,

  `subject\_code` varchar(100) NOT NULL,

  `subject\_title` varchar(100) NOT NULL,

  `category` varchar(100) NOT NULL,

  `description` longtext NOT NULL,

  `unit` int(11) NOT NULL,

  `Pre\_req` varchar(100) NOT NULL,

  `semester` varchar(100) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE `teacher` (

  `teacher\_id` int(11) NOT NULL,

  `username` varchar(100) NOT NULL,

  `password` varchar(200) NOT NULL,

  `firstname` varchar(100) NOT NULL,

  `lastname` varchar(100) NOT NULL,

   `email` varchar(100) NOT NULL,

  `department\_id` int(11) NOT NULL,

  `location` varchar(200) NOT NULL,

  `code` varchar(500) NOT NULL,

  `teacher\_status` varchar(20) NOT NULL,

  `teacher\_stat` varchar(100) NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE `teacher\_class` (

  `teacher\_class\_id` int(11) NOT NULL,

  `teacher\_id` int(11) NOT NULL,

  `class\_id` int(11) NOT NULL,

  `subject\_id` int(11) NOT NULL,

  `thumbnails` varchar(100) NOT NULL,

  `school\_year` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Our project web-based distance education management system is properly implemented with identified date types.

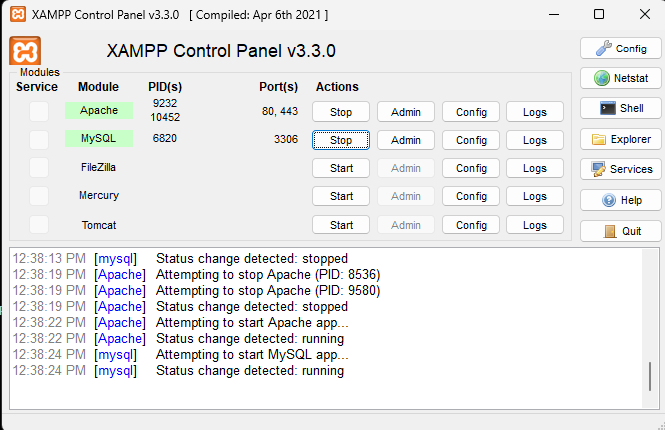
### **Implementation of the Class Diagram**

During class implementation, we engage in the following activities: -

* Defining attributes with suitable data types and access visibility.
* Defining all methods with appropriate return types, parameters, corresponding data types, and access visibility.

### **Configuration of the Application Server**

We choose XAMPP because it's easy to use and doesn't take up much space. It helps us make a local web server for testing and putting our websites online without much hassle. All we have to do is set up a few things like Apache (the server part), a database, and PHP (the programming language). XAMPP works on different kinds of computers.



### **Configuration of Application Security**

In our WEB-BASED DISTANCE EDUCATION MANAGEMENT SYSTEM FOR GREAT VISION COLLEGE, we use scripting languages like JavaScript for input validation. Upon logging in, users, including Admin, students, instructors, each with their own access privileges, can access their respective tasks. We ensure that all forms accept correct input to the best of our ability. Session control authenticates users, and the system operates 24/7 from anywhere with internet access.

Additionally, we implement message security for services invoked at the interface between applications and connected queue managers. To strengthen security, we carefully validate inputs, including login credentials, incorrect entries, and unmatched data. Clear roles are defined, with Administrator managing user information.

### Implementation of user interface

Our systems can be utilizing a user model to represent the different features of end users and their roles within the system. Our approach to user interface design focuses on user-adjusted design, aiming to create an attractive interface that is compatible, well-matched, and user-friendly. We prioritize consistency and dependability, ensuring that the interface remains stable and does not cause confusion for users. Clear and steady navigation further enhances the user experience, making it easily understandable.

### Testing

Testing is crucial to ensure the entire system works seamlessly. It involves assessing

forms, code, and modules to detect any errors. Our testing process carefully evaluates all

functionalities of the system, uncovering and addressing any issues in forms, functions,

and modules. A successful test is one that has a high chance of identifying previously

unnoticed errors. We strive to systematically uncover various types of errors

efficiently, minimizing both time and effort expended during testing.

#### **6.6.1 Test case**

#### **6.6.2 Testing Tools and Environments**

The Test Environment is essential for effective test execution, encompassing software, hardware, and network setups. It must accurately replicate the production environment to detect any potential issues. In a standard web-based application setup, we rely on Apache server, MySQL database and Windows operating system. Testing tools are pivotal for successful testing phases and product outcomes. During our unit testing, we use VS Code, Google Chrome, and XAMPP.

#### **Unit Testing**

Unit testing for authentication on the login page assumes that users can successfully log in to their appropriate pages. Test data includes various combinations of empty, valid, and invalid usernames and passwords. The testing process involves executing steps such as providing the test data and observing the expected and actual results.

* When testing with empty username and password fields, the expected result is a prompt to enter a valid email address and password. In the actual test, the system correctly prompts the user to fill out these fields.
* For cases involving invalid usernames or passwords, the expected outcome is a notification to enter valid credentials. In the actual test, the system accurately identifies the incorrect username or password.
* For cases involving invalid usernames or passwords, the expected outcome is a notification to enter valid credentials. In the actual test, the system accurately identifies the incorrect username or password.

#### **System Testing**

System testing evaluates how well the various subsystems of the complete Web Based Distance Education Management System function cohesively to achieve the desired educational outcomes. This phase ensures that the system operates seamlessly and efficiently when used by students, instructors, and administrators. It focuses on process descriptions, flows, and integration points, rather than checking individual design parts. This phase aims to verify the system as a whole, acting as a giant component. By analyzing how subsystems collaborate, system testing uncovers faults that may only surface when testing the entire system or significant portions of it. While it covers areas like performance, security, validation, and load/stress.

Sample Tests: -

* Ensure that students, instructors, and administrators can use the system efficiently to achieve their goals.
* Verify that the system operates seamlessly and efficiently for all users in the context of examinations and grading.
* Validate the overall functionality to ensure it meets the user's requirements.
* Verify that an administrator can manage user accounts, oversee course enrollment, generate reports, and manage the grading system.

#### **Integration Testing**

From the time that we integrate all the modules the testing of the functionalities of each role type comes on to the stage. We have the specific permissions related to each user type (authorization) and authentication mechanism. Our integration testing procedure is given below.

* Firstly, we will create three users who have the 3 role types namely Students, Instructor’s and Admin.
* Then will log in with the user names and passwords of each user type. this tests whether the authentication mechanism works correctly
* Then we will also try some wrong user name and wrong password. We expect an error message by trying this case.

After testing authentication mechanism, we are going to test the capabilities of each role type and many combinations of the role types and they are loosely coupled with each other.

# **CHAPTER SEVEN**

## 7. CONCLUSION AND RECOMMENDATION

### **7.1.** Conclusion

The development of a web-based distance education management system for Great Vision College signifies a paradigm shift in academic administration. This innovative solution harnesses the power of web technologies to streamline various aspects of distance learning, from course delivery to student management.

Through a single online platform accessible from anywhere, students gain the convenience of accessing course materials, submitting assignments, and participating in discussions without being physically present. This flexibility breaks down barriers to learning, allowing students to engage with their studies at their own pace and convenience.

The development of a web-based distance education management system equips Great Vision College with the means to adapt to the changing landscape of higher education. By embracing this user-friendly technology, the college enhances the accessibility, efficiency, and quality of its remote learning programs, ultimately improving the educational experience for everyone involved.

Throughout the development process, Object-oriented system analysis and design (OOSAD) and

Unified Modeling Language (UML) were employed to ensure a robust and scalable system architecture. The implementation utilized PHP for backend development, HTML and Bootstrap for frontend design, and MySQL server for data management, resulting in a user-friendly interface and seamless functionality.

### **7.2. Recommendation**

In accordance with the project's scope, our team has developed an automated system to meet the specified requirements. Because of the time constraint we cannot do beyond to our scopes, but in the future the team believes that this system can be fully operational by having enough time and full information Throughout the project development, our team encountered numerous challenges, which were overcome through collaborative efforts and guidance from advisor. As a result, we have successfully reached our final objectives. Looking ahead, we recommend further

So, we recommend next developers to include the following tasks: -

Enhanced Security Features: -

Implement robust security measures to protect user data, including encryption of sensitive information, secure authentication mechanisms, and regular security audits.

Incorporate role-based access control to restrict access to certain features and information based on user roles and permissions.

Enhanced User Interface (UI/UX):

Improve the overall design and usability of the platform to provide a more intuitive and visually appealing user experience.

Implement responsive design to ensure seamless access across various devices such as desktops, tablets, and smartphones.

Develop a notification system to keep users informed about important updates, announcements, upcoming deadlines, and changes to course schedules.

Implement evaluation tools to assess student performance, track progress, and identify areas for improvement.

# **APPENDEX: QUESTIONAIRE**

1.What are the primary challenges faced in managing distance education programs?

2.What support mechanisms are in place for students facing challenges?

3.How is collaboration facilitated among students and instructors in a virtual environment?

4.What communication tools are commonly used for effective information exchange?

1. What are the key features and components of the existing system?
2. How is the student registration process managed in the distance education system?
3. How are course modules and materials distributed to students in a distance education setting?
4. What technologies or platforms are utilized for the seamless distribution of educational content?
5. What are the primary challenges faced by the college in managing distance education programs?
6. How is the privacy of student data maintained in the system?

# **Reference**

1. Agile modeling the Object primer 2nd edition Scott W.Ambler’s book.

[2]. Object Oriented System Analysis and Design learning materials

[3]Refer from dean of great vision college

[4]https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram

[5]https://en.wikipedia.org/wiki/Activity\_diagram,Sequence\_diagram

[6] B. H. W. D. T. Alan Dennis, System Analysis &Design an Object-Oriented Approach

with UML, WILEY, 2015

[7] (n.d.). Retrieved from Visual-paradign: https://[www.visual-paradigm.com/guide/uml-unified-modeling-](http://www.visual-paradigm.com/guide/uml-unified-modeling-) language/what-is-use-case-diagram

[8] Joseph S. Valacich, J. F. (n.d.). Modern Systems Analysis and Design 8th Edition. In J. F. Joseph S. Valacich, Modern Systems Analysis and Design 8th Edition.

[9] Object Oriented System Analysis and Design learning materials. (n.d.).

[10]studysmarter(n.d.).Retrievedfrom studysmarter: https://[www.studysmarter.co.uk/explanations/law/uk-](http://www.studysmarter.co.uk/explanations/law/uk-) legal-system/judges/

[11] W.Ambler, S. (n.d.). Agile modeling the Object primer 2nd edition.

[12] wikipedia. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Class\_diagram