

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION TECHNOLOGY

PROJECT TITLE: VIRTUAL CLASS ROOM AND CHATTING SYSTEM FOR WOLKITE UNIVERSITY BUTAJIRA CAMPUS

NAME	***ID No.***
1. SEFANIT MENGISTU	***CIR/222/10***
2. KEDIR OLANA	***CIR/197/10***
3. MUBAREK NAZIF	***CIR/213/10***

Advisor: Mr. Amanuel Tamirat

Wolkite University, Wolkite, Ethiopia

Submission Date: 25/08/2021

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION TECHNOLOGY

VIRTUAL CLASS ROOM AND CHATTING SYSTEM FOR WOLKITE UNIVERSITY BUTAJIRA CAMPUS

SUBMITTED TO DEPARTMENT OF INFORMATION TECHNOLOGY IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHLER OF SCIENCE IN INFORMATION TECHNOLOGY

BY

1. Sefanit Mengistu
2. Kedir Olana
3. Mubarek Nazif

ADVISOR: Mr. Amanuel Tamirat

Wolkite University, Wolkite, Ethiopia

May, 25/08/2021

DECLARATION

This is to declare that this project work, which is done under the supervision of Mr. Amanuel Tamirat and having the title:

Virtual Class Room And Chatting System For Wolkite University Butajira Campus System Is The Contribution of:

1. Sefanit Mengistu
2. Kedir Olana
3. Mubarek Nazif

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.

Date: _____

Group Members:

Full Name	Signature
_____	_____
_____	_____
_____	_____

APPROVAL SHEET

This is certified that the project entitled on virtual classroom system and chatting system for WKU developed and submitted by

Name and Signature of Group members

No.	Name	ID No.	Signature	Date
1.	Sefanit Mengistu	CIR/222/10	…….....	…….....
2.	Kedir Olana	CIR/197/10	…….....	…….....
3.	Mubarek Nazif	CIR/213/10	…….....	…….....

A Project Proposal Submitted to the Department of Information Technology of Wolkite University in Partial Fulfillment of Requirements for the Degree of Bachelor of Science in Information Technology

Advisors' Name	Signature	Date
Mr. Amanuel Tamirat	_____	_____

APPROVED BY

Examining Board	Signature	Date
1. _____	_____	_____
2. _____	_____	_____
3. _____	_____	_____

ACKNOWLEDGMENT

First of all, we would like to praise our God who helps us to accomplish this project documentation successfully. Our Next deepest gratitude will go to Mr. Amanuel Tamirat who is our advisor. He gave us a chance to acquire more detailed knowledge about the project and he helps us in any confusion that was happening during the documentation development. This makes us more knowledgeable about the project named Virtual Class Room and Chatting System. Thank you! Teacher. We are proud to be your students. Finally, we would like to thank all the people who were with us in everything they have without hesitation.

Contents

ACKNOWLEDGMENT I
LIST OF TABLES V
LIST OF FIGURES VI
LIST OF ABBREVIATIONS AND ACRONYMS VIII
ABSTRACT IX
CHAPTER 1 1
1. INTRODUCTION 1
1.1. Background 1
1.2. Statement of the Problem 2
1.3. Objective 2
1.3.1. General Objective 2
1.3.2. Specific Objectives 2
1.4. Methodology 3
1.4.1. Data collection methodology 3
1.4.2. System Development Methodology 3
1.5. Tools and Material 4
1.5.1. Software tools 4
1.5.2. Hard ware tools 4
1.6. Scope 4
1.7. Significance of the work 5
1.8. Beneficiary of the project 5
1.9. Feasibility Assessment 6
1.9.1. Operational Feasibility 6
1.9.2. Economic Feasibility 6
1.9.3. Technical Feasibility 7
CHAPTER 2 8
2. DESCRIPTION OF THE EXISTING SYSTEM 8
2.1. Introduction of Existing system 8
2.2. Users of the Existing system 8
2.3. Major function of the existing system 8
2.4. Drawbacks of the Existing system 9

2.5. Business Rule 9
CHAPTER 3 11
3. PROPOSED SYSTEM 11
3.1. Functional requirement 11
3.2. Nonfunctional requirement 13
3.2.1. User interface 13
3.2.2. Hardware consideration 13
3.2.3. Security Issues 14
3.2.4. Performance consideration 14
3.2.5. Error Handling and validation 14
3.2.6. Backup and recovery 14
3.2.7. Physical Environment 14
CHAPTER 4 16
4. SYSTEM ANALYSIS 16
4.1. System model 16
4.1.1. Use case model 16
4.2. Object Model 34
4.2.1. Class Diagram 34
4.2.2. Data Dictionary 35
4.3. Dynamic Model 38
4.3.1. Sequence Diagram 38
4.3.2. Activity Diagram 49
4.3.3. State chart diagram 55
CHAPTER 5 56
5. SYSTEM DESIGN 56
5.1. Design Goal 56
5.2. Proposed system Architecture 57
5.2.1. Sub system decomposition and Description 58
5.2.2. Hardware /Software Mapping 59
5.2.3. Detailed Class Diagram 61
5.2.4. Persistent Data Management 62
5.2.5. Access Control and Security 63

5.3. Packages 64
5.4. Algorithm Design 65
5.5. User Interface Design 66
CHAPTER 6 67
6. IMPLEMENTATION AND TESTING 67
6.1 Implementation of the Database 67
6.3 Configuration of the Application Server 71
6.4 Configuration of Application Security 71
6.5 Implementation of User Interface 71
6.6 Testing 76
6.6.1 Testing Tools and Environment 76
6.6.2 Unit Testing 77
6.6.3 System Testing 77
6.6.4 Integration Testing 78
6.6.5 Acceptance Testing 78
CHAPTER 7 79
7. CONCLUSION AND RECOMMENDATION 79
7.1 Conclusion 79
7.2 Recommendation 79
REFERENCES 81

LIST OF TABLES

Table 1: Actor Description..... 18
Table 2: Log in use case description..... 18
Table 3: Add course use case description..... 19
Table 4: Delete course use case description..... 21
Table 5: Modify user account use case description..... 22
Table 6: Add instructor use case description..... 23
Table 7: Delete instructor use case description..... 25
Table 8: Update instructor use case description..... 26
Table 9: Follow up course use case description..... 28
Table 10: Upload assignment use case description..... 29
Table 11: Add course material use case description..... 30
Table 12: Delete course material use case description..... 31
Table 13: Chatting use case description..... 33
Table 14: Persistent database design..... 35
Table 15: Access Control and Security..... 63

LIST OF FIGURES

Figure 1: Iterative Model 3
Figure 2: Use case diagram 17
Figure 3: Class Diagram: Class Diagram 34
Figure 4: Login sequence diagram 38
Figure 5: Add Course Sequence Diagram 39
Figure 6: Delete Course Sequence Diagram 40
Figure 7: Update user account sequence diagram 41
Figure 8: Add instructor record sequence diagram 42
Figure 9: Delete Instructor Sequence Diagram 43
Figure 10: Update Instructor Record 44
Figure 11: Follow up course sequence diagram 45
Figure 12: Add course material sequence diagram 46
Figure 13: Delete Course material 47
Figure 14: User Chatting Sequence diagram 48
Figure 15: Log in activity diagram 49
Figure 16: Modify User account (Changing Password) activity diagram 49
Figure 17: Add Course Record Activity Diagram 50
Figure 18: Delete course record activity diagram 50
Figure 19: Update course record activity diagram 51
Figure 20: Add Instructor Record activity diagram 51
Figure 21: Delete instructor record activity diagram 52
Figure 22: Update instructor Record activity diagram 52
Figure 23: Search activity diagram 53
Figure 24: Add course material activity diagram 53
Figure 25: Delete course material activity diagram 54
Figure 26: Chatting activity diagram 54
Figure 27: Course state chart diagram 55
Figure 28: Proposed system architecture 58
Figure 29: Sub system decomposition 59
Figure 30: Hardware /Software Mapping 60

Figure 31: Detailed Class Diagram 61
Figure 32: Persistent Modeling/ Database Design 62
Figure 33: Packages 64
Figure 34: home page user interface diagram 66
Figure 35: login page user interface diagram 66
Figure 36 Login sample code 72
Figure 37: Upload material 75
Figure 38: Download uploaded materials 75
Figure 39: Download Uploaded Assignments 76

LIST OF ABBREVIATIONS AND ACRONYMS

VCRCS: Virtual Class Room and Chatting System (The proposed system project title)

PHP: Hypertext Preprocessor (programming language used to develop web pages)

HTML: Hyper Text Mark Up language (programming language used to develop web pages)

CSS: Cascading Style Sheet (programming language used to develop web pages)

MySQL: Open source relational database management system

UC: Use case (Unified Modeling Language)

UI: User Interface

UML: Unified Modeling Language

VoIP: Voice over Internet Protocol

ADMIN: Administrator

BR: Business Rule

PK: Primary Key

ABSTRACT

Virtual Class Room and Chatting System for Wolkite University Butajira Campus a web based application which aimed to change the manual system of the learning and teaching process to automated system. It is one of the major learning methods in the era of information that allows students and lecturers to go online and continue their normal school activities. The traditional method of teaching and learning has some shortfalls. The proposed system is equipped with features that minimizes problems in traditional learning method. The virtual classroom is more conducive and the students can readily get their study materials easily. There is a module where the system administrator can create other lecturers that will use the system. There is a module where students can login to read announcements, there is assessment module where lecturers may post and students can view. This system will encourage learning of the individual topic contained therein and enhance the student's practical knowledge and broaden their theoretical knowledge while creating a more interacting learning process. The system is implemented using PHP for the front end and MySQL for the database.

CHAPTER 1

1. INTRODUCTION

A virtual classroom is an online learning environment. The environment can be web-based and accessed through a portal or software-based and require a downloadable executable file. Many schools and businesses have rolled out virtual classrooms to provide synchronous distance education. Virtual classroom software applications often employ multiple synchronous technologies, such as web conferencing, video conferencing, live streaming, and web-based VoIP to provide remote students with the ability to collaborate in real time. To enhance the educational process, applications may also provide students with asynchronous communication tools, such as message boards and chat capabilities. A virtual classroom is a learning environment created in the virtual space. Such type of learning is very less expensive in comparison to other type of learning in higher education.

Developing a virtual classroom system to promote a greater count of students to splurge into the field of education. It integrates the benefits of a physical classroom with the convenience of a 'no-physical-bar' virtual learning environment, minus the commuting hazards and expenses. It will usher in the immense flexibility and sophistication in the existing learning platform structures, with the perfect blend of synchronous and asynchronous interaction. It provides a means of collaborative learning for the students. Our objective is not to merely duplicate the characteristics and effectiveness of the face to face class. Rather, we can use the powers of the computer to actually do better than what normally occur in the face to face class [1].

1.1. Background

Wolkite University (WKU) is one of the third generation higher institutions that have been founded in 2012. It is established for the purpose of providing and promoting higher education learning, research, and outreach programs in the country to ensure the realization of the national vision of reaching the level of middle income countries by 2020. The University is located in the Southern Nation Nationalities Regional State, in Guraghe zone, 158 km southwest of the capital city, Addis Ababa, on the way to Jima. In November 2009 the late prime minister, his Excellency Mr. Meles Zenawi, laid the foundation stone of the University in a plain landscape which is quite ideal for academic pursuit. It is situated at Gubreye sub-city, 14 km away from Wolkite town, of the

Gubrye-Butajira road. The major link road to the University is a direct route to Wolkite-Jimma, Wolkite-Hossana and Wolkite-Butajira.

1.2. Statement of the Problem

Nowadays the era of information technology is changing the social aspect of the society. From the most important aspects, the teaching and learning process is the major one. The Traditional which is more obvious way of teaching learning process is replaced by easy, flexible, more interactive and advanced technology like the virtual class rooms.

The problem that exists in the current system are:

- The current system which can maximize the spreading of pandemic virus like COVID-19
- Students or instructors cannot attend the class if they are distance or gap between course instructor and student.
- Need for expensive travel (or even accommodation). Also maximize your cost related to driver, gas and fuel.
- Students lack many resources which is essential to their course like video and audio.

1.3. Objective

1.3.1. General Objective

The objective of this paper is to develop a virtual classroom and chatting system to enhance learning on WKU.

1.3.2. Specific Objectives

The specific objectives of the system are:

- To develop a web application through which teachers can appear in front of webcam in their desk, and students can view the teacher and listen to her/his voice through a web page at real time.
- To facilitate the learners or viewers with multiple options like raising hands, lettering in whiteboard.
- To provide the facility to extract the loaded resources at any instant.

1.4. Methodology

1.4.1. Data collection methodology

In order to gather the relevant data, we use both primary and secondary way of data collection.

Primary Data Collection:

Practical Observation- enables us to understand and list out the manual system problems via observing the limitations.

Secondary Data Collection:

Referencing- For having sufficient information for our system we will use different documents as a reference for developing this system.

1.4.2. System Development Methodology

For developing our system, we will use Iterative model. The following figure shows the steps involved in iterative model.

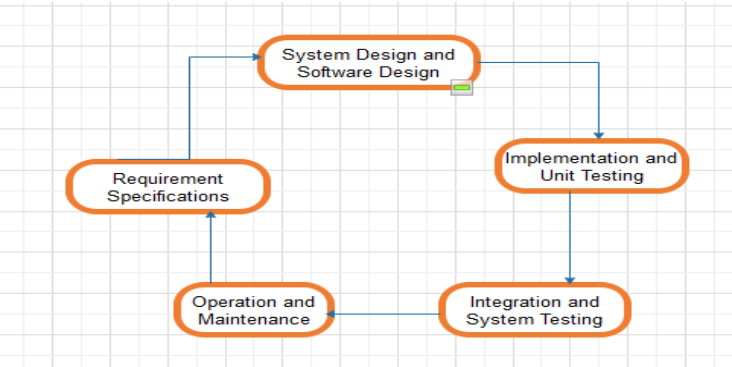


Figure 1: Iterative Model

We preferred Iterative model approach for the following advantages:

- It is more cost effective to change the scope of the requirements in Iterative model.
- Parallel development can be planned.
- Testing and debugging during smaller iteration is easy.
- Risks are identified and resolved during iteration; and each iteration is an easily managed.

1.5. Tools and Material

1.5.1. Software tools

WAMP: the combination of three software's for windows APACHA, MYSQL, and PHP.

- MYSQL for data base store
- PHP version5.3.5 for programming
- APACHA version2.2.17for as a server

Sublime for code editor

CSS HTML and JavaScript for style, for form display and client side scripting respectively.

E_Draw, Paint to draw the UML Diagrams and for designs

1.5.2. Hard ware tools

Computers with the following storage uses for developing purpose

- 4GB RAM for computer storage when computer start up.
- 465GB hard disk for data storage permanently
- 2.4GHz processor with Intel and Pentium internally

Flash and CD for data recovery and backup

1.6. Scope

The virtual class room system is an online web-based teaching learning environment to improve access to advanced educational experiences by allowing users to participate in teaching learning communities using personal computers, and to improve the quality and effectiveness of education by using the computer to support a collaborative learning process only for Wolkite University Butajira Campus students. This virtual system provides not only online teaching also chatting function to the students. The system mainly focuses on the following points and it will provide:

- Since the project is for Wolkite University Butajira Campus Student there will be fixed user id and password. The user id cannot be changed but the password given to user changes in the first login and can be changed any time.
- The registered users can follow up for the course that the curriculum given to follow.

- Many available resource for different course.
- Reminding message via email for all user.
- Sends weekly assignments for the user via user account or user email and then post exam or assignment result.
- Chat between users.
- Active or deactivate user account.
- Admin can Post news.

1.7. Significance of the work

The proposed virtual class room system is expected to bring the remarkable change in the teaching learning environment over the existing or traditional system in the following major significant

- The system is cost effective because, there is no need for expensive travel (or even accommodation). As well as saving travel costs, virtual classrooms will also reduce your cost related to gas and fuel.
- The system will be functional independent of time and place because you can see different uploaded videos and documents.
- Teaching learning resources availability (the system will provide sufficient materials for the user)
- Inspire and motivate users at different level to learn and gain knowledge teacher can give interesting insights that inspire new discussions.
- Useful for distance learning, which can also minimizes the spreading of pandemic virus like COVID-19

1.8. Beneficiary of the project

Virtual classroom allows student and instructors to communicate synchronously using features such as audio, video, text chat, interactive whiteboard and application sharing

Instructors

- Incorporate immediate feedback for students
- Host live chats
- Share videos

- Offer interactive lessons
- Quick delivery of lessons(in fast and easy way)

Students

- Online learning accommodates everyone's needs
- Lectures can be taken any number of times without limitation
- Allow to access updated content
- Reduced costs (virtual learning may help students to get those course which were if the join some university they may huge charge on they pockets)
- Effectiveness (students were encouraged, trained for improving their practical skills)

1.9. Feasibility Assessment

A feasibility study decides whether or not the proposed system is worthwhile in different dimensions. It measures how much the proposed system is beneficial or practical at development of the system. The feasibility factors of our project are:

- Economic feasibility
- Operational feasibility
- Technical feasibility
- Schedule feasibility

1.9.1. Operational Feasibility

Measure how much the proposed system solves the existing system problems. This project is surely operationally feasible because the proposed system (the project) is a good solution maker of the problem and create a good environment towards the user of our application by providing easy, user interactive and everywhere access for those who cannot attend a normal class.

1.9.2. Economic Feasibility

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study outlines the economic justification for the new system.

A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include

- Increased customer satisfaction
- Improvement in quality
- Better decision making timeliness of information
- Expediting activities
- Improved accuracy of operations
- Better documentation and record keeping
- Faster retrieval of information

1.9.3. Technical Feasibility

Technical feasibility is the measure of the practicality of a specific technical solution and the availability of technical resources. In technical feasibility we should notify that our proposed system can implement with current technology and also the system user has enough experience using that technology. Technical feasibility addresses three main things:

- Is the technology practical?
- Do we currently pass the necessary technology?
- The ability to do on the technologies.

So we can say that our system is technically feasible because of three main reasons. The first one is our project is compatible with the era that we are living in, that is information era. The second reason is we can implement our system using current technology that is by using notepad++ for PHP development and also for organizing our data we have MySQL data base and for system design we have EdrawMaxv5.1 and Visio 2013. The last reason is users cannot face additional burdens to be familiar with the system.

CHAPTER 2

2. DESCRIPTION OF THE EXISTING SYSTEM

2.1. Introduction of Existing system

The present system is a manual system. Manual system involves paper work in the form of maintaining various files and manuals. The teacher who communicates in-person with a group of students in a brick and mortar facility. This method of learning typically involves students gathering in physical classrooms during a certain timeframe with the purpose of learning about specific topics or to gain specific hands-on job training and experience.

2.2. Users of the Existing system

As we study we could find that there are two users that are benefited from our projects.

Teacher

- Planes the lecture
- Delivers the lecture in the classroom
- Develops contents of didactic resource

Student

- Attend a lecture
- Use the didactic resource
- Do assignment

2.3. Major function of the existing system

In a typical approach, students sat quietly at their places and listened to one student after another recite his or her lesson, until each had been called upon. The teacher's primary activity was assigning and listening to these recitations; students studied and memorized the assignments at home. A test or oral examination might be given at the end of a unit, and the process, which was called "assignment- study-recitation-test", was repeated. In addition to its overemphasis on verbal answers, reliance on rote memorization (memorization with no effort at understanding the meaning), and disconnected, unrelated assignments, it was also an extremely inefficient use of

students' and teachers' time. This traditional approach also insisted that all students be taught the same materials at the same point; students that did not learn quickly enough failed, rather than being allowed to succeed at their natural speeds.

2.4. Drawbacks of the Existing system

In the existing system students need to learn course in the class room only. The classroom is far time is a factor to take into account. You have to decide whether the trip is worth it or not.

Traditional education is more limited when it comes to educational formats. Sometimes pictures, videos or forums aren't accessible 24/7. This feature might limit the learning and hinder a good study plan.

2.5. Business Rule

A business rule is effectively an operating principle or polices that we try to specify for both the existing system and the new system must satisfy. The business rule is a principle or a policy in which the proposed system operates accordingly. Therefore, our new system has the following business rules

Name: Valid student

Identifier: BR#01

Description: Anyone who uses the system should be in a particular department.

Name: Attending class attentively

Identifier: BR#02

Description: Students should attend class and took evaluation properly

Name: Submitting assignment

Identifier: BR#03

Description: Students have to submit any assignment before the dead line

Name: Preparing course material

Identifier: BR#04

Description: Instructors have to prepare a course material to the students

Name: - Make sure your device is charged and ready to use.

Identifier: - BR#05

Description: - Don’t forget to plug your device up each night so you are ready to go for the next day.

Name: **Do not drink or eat during class**

Identifier: - BR#06

Description: - In general, you shouldn’t engage in activities that don’t contribute to the lesson. Eating and drinking is an obvious example here.

Name: - Manage course

Identifier: - BR#07

Description: -Department head add, delete and update course.

Name: - Create a quiet and comfortable study space

Identifier: - BR#08

Description: - Find a location that is free from any distraction by other family members, pets, background noises, or cell phones.

CHAPTER 3

3. PROPOSED SYSTEM

The proposed virtual class room system provides the number of advantages over the existing system. The development of this new system contains the following activities, which try to automate the entire process.

Some of the major features of the proposed system over the existing system are:

- The system will have rich user interface as a result it will be user friendly.
- The system makes the overall teaching and learning process much easier and flexible.
- It can be accessed over the Internet.
- The user information can be stored on a centralized database since data is not stored in the client machine this gives us good security.
- Authentication is provided for this application only registered members can access. As the virtual classroom is one that aims to give the student an experience equal to or better than the sort they would find in a traditional classroom, there are obviously many advantages of the virtual classroom to the student, as well as the teacher (or instructor) and the associated educational institution [2].

3.1. Functional requirement

Functional requirements are the functions of the proposed system. It describes the interactions between the system and its environment independent of its implementation. The major functionalities of the system are the following:

FR#01: Login into the system

Description:

The system must allow all authorized users to access the system more efficiently.

FR#02: Manage user's account:

Description:

A. The system must allow for administrator to creating user accounts.
B. The system must allow for administrator to deleting user accounts.
C. The system must allow for administrator to Activate and deactivating user accounts.
D. The system must allow for administrator to Post news or changed policies in the campus.

FR#03: Uploading modules and assignments:

Description:

A. The system must allow for Instructors to upload module and assignment questions.
B. The system must allow for Instructors to post the assignment submission date.

FR#04: Download modules and assignments:

Description:

The system must allow for Instructors to download assignment answers submitted by students.

FR#05: Download modules and assignments:

Description:

The system must allow for Student to download module and assignment questions.

FR#06: Submitting assignment

Description:

The system must allow for Students to submit any assignment.

FR#07: Preparing course material

Description:

A. The system must allow for Instructors to View assigned course.
B. The system must allow for Students' to follow up Courses.

FR#08: Chatting

Description:

A. Students can chat with students in order to exchange information.
B. Students can chat with teachers in order to exchange information.

FR#09: Manage course

Description:

A. The system must allow for head of department to add course.
B. The system must allow for head of department to delete course
C. The system must allow for head of department to update course.

FR#10: Manage course material

Description:

A. The system must allow for instructor to add course material.
B. The system must allow for instructor to delete course material.

3.2. Nonfunctional requirement

Nonfunctional requirements are requirements that do not directly affect the performance of the system. These requirements relate to the technical aspects of the system that has to be addressed, such as the security issue. Because nonfunctional requirements are requirements that are constraints on the system.

Some of the nonfunctional requirements are:

3.2.1. User interface

Users can easily input and retrieve their profile and need information. This works as an interface between the user and the system by properly guiding the user how to use it and perform operations.

3.2.2. Hardware consideration

The proposed system considers computer hardware without any error and run on any personal computer.

3.2.3. Security Issues

The system has to be well protected unauthorized access.

- Encrypted Password.
- All users should be authenticated before granting access to the Virtual Classroom. All users should have access according to their respective authorization privileges.

3.2.4. Performance consideration

The system should have to perform in a proper way without any problem. Performance requirements define acceptable response times for system functionality.

- The load time for user interface screens shall take no longer than five seconds.
- The log in information shall be verified within five seconds.
- Queries shall return results within five seconds.
- The system should be able to accommodate at least 50 users simultaneously in the video conferencing mode. This requirement will ensure that at least one class can proceed during any time slot. This requirement can be strengthened when more resources are added to the system.

3.2.5. Error Handling and validation

- The proposed system will minimize crash during its runtime, since more than one user could use the system simultaneously.
- Good validations for user inputs will be done.
- Avoid incorrect storage of records.

3.2.6. Backup and recovery

The system will be developed for easy backup and recovery. The system has to be easily understandable for the users and have to be easy for maintenance.

3.2.7. Physical Environment

Virtual class room gave rise to a new form of immersive learning environment, virtual immersive learning environments that, unlike the seemingly effortless origins of their physical classroom

forbears, are intentionally engineered to mimic physical reality. Virtual immersive learning environments are so dominant today that they are widely identified by a simple acronym - ILE. ILEs are used for training purposes because virtual immersive environments offer a safe, efficient and inclusive way for individuals to learn and apply new skills that can then be applied in the real world [3].

CHAPTER 4

4. SYSTEM ANALYSIS

4.1. System model

System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system.

4.1.1. Use case model

A use case model is the representation of the system intended functions and its environment. The functionalities are specified by the “use case” and “the actor “specified to the environment. Since the identification of these two entities goes hand in hand, we have identified them together.

The actors are:

- Instructor
- Student
- Administrator
- Department head

The most important use cases for the proposed system are:

1. Login
2. Manage Account
3. Manage Course
    - Add Course
    - Delete Course
    - Update Course
4. Manage Instructor
    - Add Instructor
    - Delete Instructor
    - Update Instructor
5. View assigned course
6. Manage Course Material

- Add Course Material
- Delete Course Material

7. Evaluation
   - Manage Assignment
8. Modify User Information
9. Chatting
10. Follow up Courses
11. Take Assignment Evaluation

4.1.1.1. Use case diagram

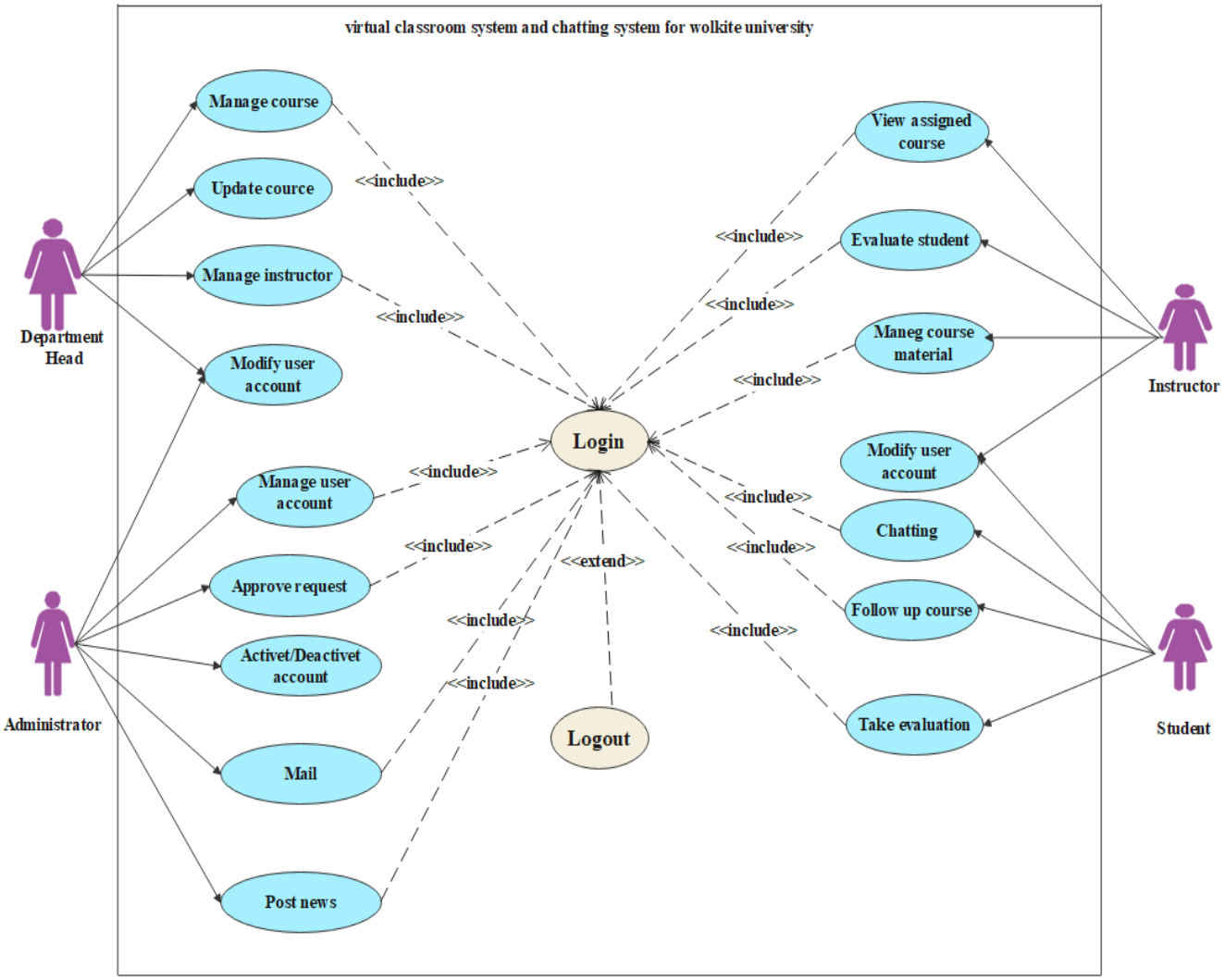


Figure 2: Use case diagram

Actor Description Table

Table 1: Actor Description

S No.	Actor Name	Description
1.	**Department Head**	Manage course and instructor, assign course to specific instructor.
2.	**Administrator**	Manage the overall user account; supremely control the detail functions of the system
3.	**Instructor**	Manage the teaching learning functionality of the system including managing courses, course materials, and user evaluation.
4.	**Students**	Users are those who can access their own user profile. They can follow up course take evaluation and print handouts.

4.1.1.2. System use case Description

Use case description explains in detail the general flow of use case diagrams. Each table contains the use case name, use case ID, the actor that initiates and interacts with the use case, and flow of event that show the interaction between the actor and the use case which enable the user to easily understand the functions of the proposed system.

Login Use Case Description

Table 2: Log in use case description

Use case name	**Login**
Use case identifier	UC01
Actor(s)	Administrator, Department head, Instructors and Users
Description	It allows all the user to use and interact with the system
Pre-condition	The user must have a valid user name and password.

Basic Course of Action	**1.** The user sends the request to the server using web browser. **2.** The System displays the Login page. **3.** The user enters username and password and press Login button. **4.** The system validates the user account. **5.** The user is valid then the system displays home page. **6.** The user access the system. **7.** Use case end.
Alternative Course of Action	**Alternate Course A**: the user is not found in the system A.4. system displays an error message **"Incorrect User Name and Password".** A.5. The use case ends.
Post Condition	The user access the system.

Manage Course Use Case Description

Table 3: Add course use case description

Use Case name	**Add Course**
Use Case identifier	UC02
Actor(s)	Department head
Description	It allows the department head to add available courses to the system
Pre-condition	The department head must have a valid account and must have list of available courses in the system database.

Basic Course of Action	**1.** The department head logs in to account. **2.** The System displays the administrator page. **3.** The department head clicks on the **Manage Course Button**. **4.** The system displays the course manage page. 5. Then the department head clicks on **ADD Course** button. 6. The system displays the page to add new course. 7. The department head add the available course to the system then click **Save** button. 8. Use case ends.
Alternative Course of Action	**Alternate Course A**: Department head have invalid account A.2. If the Department head does not have a valid account. A.3. The system displays error message. A.4. Use case end. **Alternate Course B**: No available course B.7 If there is no available course to add. B.8. The system displays error message. B.9. Use case end.

Delete course use case description

Table 4: Delete course use case description

Use Case name	**Delete Course**
Use Case identifier	UC03
Actor(s)	Department head
Description	It allows the department head to delete an existing course from the system
Pre-condition	The department head must have an administrator account and must have list of available courses.
Basic Course of Action	**1.** The department head logs in to account. **2.** The System displays the department head page. **3.** The department head clicks on the **Manage Course Button**. **4.** The system displays the course manage page. 5. Then the department head search for the course to be deleted using course name and course code. 6. The system search and find the course to be deleted. 7. The department head then clicks **Delete** button then click **Save** button. 8. Use case ends.
Alternative Course of Action	**Alternate Course A**: Department head have invalid account A.2. If the department head does not have a valid account. A.3. The system displays error message. A.4. Use case end.

	Alternate Course B: Course to be delete don't found B.7 If the course to be deleted don't found. B.8. The system displays **'Search Not Found!!'** error message. B.9. Use case end.
Post-condition	The selected course will be removed from the system.

Modify User Profile Use Case Description

Table 5: Modify user account use case description

Use Case name	**Modify User Profile**
Use Case identifier	UC04
Actor(s)	Instructor, Administrator, Department head and Student
Description	Allows the Student, instructor, Administrator or department head to modify the user account information.
Pre-condition	The student, instructor, Administrator or department head must have a valid user account.
Basic Course of Action	**1.** The student, instructor, Administrator or department head logs in to account. **2.** The System displays the user's page. **3.** The student, instructor, Administrator or department head clicks on the **Edit Profile** link. **4.** The system displays the page to edit the profile. 5. Then the user searches for the information to be modified and edit the required information then click **Save** button.

	6. Use case ends.
Alternative Course of Action	**Alternate Course A**: User have invalid account A.2. If the user does not have a valid account. A.3. The system displays error message. A.4. Use case end. **Alternate Course B**: Incorrect information entered B.5. If the information that is being modified is not valid. B.6. The system displays **'Invalid Information Entered!!'** error message. B.7. Use case end.

Manage Instructor Use Case Description

Table 6: Add instructor use case description

Use Case name	**ADD Instructor**
Use Case identifier	UC05
Actor(s)	Administrator
Description	It allows the Administrator to add new instructor and detail profile to the system.
Pre-condition	The Administrator must have valid account.
Basic Course of Action	**1.** The Administrator logs in to account. **2.** The System displays the Administrator head page. **3.** The Administrator clicks on the **Manage Instructor Button**.

	4. The system displays the instructor manager page. 5. Then the Administrator clicks on **ADD Instructor** button. 6. The system displays the form to add new instructor. 7. The Administrator fill all the required information then click **Save** Button. 8. Use case ends.
Alternative Course of Action	**Alternate Course A**: Administrator have invalid account A.2. If the Administrator does not have a valid account. A.3. The system displays error message. A.4. Use case end. **Alternate Course B**: Incorrect information entered B.7. If the information that is being modified is not valid. B.8. The system displays **'Invalid Information Entered!!'** error message. B.9. Use case end.

Delete instructor use case description

Table 7: Delete instructor use case description

Use Case name	**Delete Instructor**
Use Case identifier	UC06
Actor(s)	Administrator
Description	It allows the Administrator to delete an existing instructor from the system
Pre-condition	The Administrator must have valid account.
Basic Course of Action	**1.** The Administrator logs in to account. **2.** The System displays the Administrator page. **3.** The Administrator clicks on the **Manage Instructor Button**. **4.** The system displays the instructor manager page. 5. Then the Administrator search for the instructor to be deleted using name and instructor **ID**. 6. The system search and find the Instructor to be deleted. 7. The Administrator then clicks **Delete** button then click **Save** button. 8. Use case ends.
Alternative Course of Action	**Alternate Course A**: Administrator have invalid account A.2. If the Administrator does not have a valid account. A.3. The system displays error message. A.4. Use case end. **Alternate Course B**: Instructor to be delete don't found

	B.5. If the instructor to be deleted don't found. B.6. The system displays **'Search Not Found!!'** error message. B.7. Use case end.

Update instructor use case description

Table 8: Update instructor use case description

Use Case name	**Update Instructor**
Use Case identifier	UC07
Actor(s)	Administrator
Description	It allows the Administrator to update an existing instructor in the system
Pre-condition	The Administrator must have a valid account.
Basic Course of Action	**1.** The Administrator logs in to account. **2.** The System displays the Administrator page. **3.** The Administrator clicks on the **Manage Instructor Button**. **4.** The system displays the instructor manager page. 5. Then the Administrator search for the instructor to be updated using name and instructor **ID**. 6. The system search and fine the instructor to be updated. 7. The Administrator then clicks **Update** button. 7. Then updating the required information for the instructor take place by the system. 8. Use case ends.

Alternative Course of Action	**Alternate Course A**: Administrator have invalid account A.2. If the Administrator does not have a valid account. A.3. The system displays error message. A.4. Use case end. **Alternate Course B**: Instructor to be Update don't found B.5. If the instructor to be updated don't found. B.6. The system displays **'Search Not Found!!'** error message. B.7. Use case end. **Alternate Course C**: Incorrect information entered C.7. If the information that is being modified is not valid. C.8. The system displays **'Invalid Information Entered!!'** error message. C.9. Use case end.

Follow Up Course Use Case Description

Table 9: Follow up course use case description

Use Case name	**Follow Up Course**
Use Case identifier	UC08
Actor(s)	Students
Description	It allow the student to attend semester course.
Pre-condition	The student must have a valid user account.
Basic Course of Action	**1.** The student logs in to the system. **2.** The System displays the user account page. **3.** The Student clicks on the **Semester Courses** Button on the homepage inside user account. **4.** The system provides the entire available courses list with their detail according to the curriculum. **5.** The student click on **Follow Up** button to start attending the semester course. **6.** The system displays a password box for the student to confirm that the student is who claims to be. 7. Use case ends.
Alternative Course of Action	**Alternate Course A**: Student have invalid account A.2. If the student does not have a valid account. A.3. The system displays error message. A.4. Use case end.

	Alternate Course B: Confirm password B.7. If the student enters an invalid password. B.8. The system displays error message. B.9. Use case end.
Post-condition	The user will successfully register for the course and can learn it.

Manage Assignment Use Case Description

Table 10: Upload assignment use case description

Use Case name	**Upload Assignment**
Use Case identifier	UC09
Actor(s)	Instructor
Description	Allow the instructor to manage the overall assignment evaluation of students.
Pre-condition	The instructor must have a valid user account.
Basic Course of Action	**1.** The instructor logs in to account. **2.** The System displays the instructors account page. **3.** The instructor clicks on the **Assignment Evaluation** Button on the homepage inside user account. **4.** The system displays the **Assignment** page. **5.** The Instructor upload assignment for his students. 6. Use case ends.

Alternative Course of Action	**Alternate Course A**: Instructor have invalid account A.2. If the instructor does not have a valid account. A.3. The system displays error message. A.4. Use case end.
Post-condition	User will be evaluated using the assignment successfully.

Manage Course Material Use Case Description

Table 11: Add course material use case description

Use Case name	**Add Course Material**
Use Case identifier	UC10
Actor(s)	Instructor
Description	It allows the instructor to add available course materials to the system for a specific course.
Pre-condition	The instructor must have a valid account and must have list of available course materials.
Basic Course of Action	**1.** The instructor logs in to account. **2.** The System displays the instructor page. **3.** The instructor clicks on the **Manage Course Material** Button. **4.** The system displays the course material manager page. 5. Then the instructor clicks on **ADD Course Material** button. 6. The system displays the page to add new course material.

	7. The instructor adds the available course material information to the system. 9. The instructor click on **Upload** button. 10. The System display System open file dialog box. 11. The instructor select and then click **Open** button. 12. The instructor click **Post** button. 13. The system upload the course material. 14. Use case ends.
Alternative Course of Action	**Alternate Course A**: Instructor have invalid account A.2. If the instructor does not have a valid account. A.3. The system displays error message. A.4. Use case end.

Delete course material use case description

Table 12: Delete course material use case description

Use Case name	**Delete Course Material**
Use Case identifier	UC11
Actor(s)	Instructor
Description	It allows the instructor to delete an existing courses material from the system.
Pre-condition	The instructor must have a valid account.

Basic Course of Action	**1.** The instructor logs in to account. **2.** The System displays the administrator page. **3.** The instructor clicks on the **Manage Course Material Button**. **4.** The system displays the course material manager page. 5. Then the instructor search for the course material to be deleted using course material name and course material code. 6. The system search and find the course material to be deleted. 7. The instructor then clicks **Delete** button then click **Save** button. 8. Use case ends.
Alternative Course of Action	**Alternate Course A**: Instructor have invalid account A.2. If the Instructor does not have a valid account. A.3. The system displays error message. A.4. Use case end. **Alternate Course B**: Course material to be delete don't found B.7. If the course material to be deleted don't found. B.8. The system displays **'Search Not Found!!'** error message. B.9. Use case end.
Post-condition	The selected course material will be removed from the system.

Chatting Use Case Description

Table 13: Chatting use case description

Use Case name	**Chatting**
Use Case identifier	UC12
Actor(s)	Student
Description	It allow the student to communicate with other member
Pre-condition	The student must have a valid user account.
Basic Course of Action	**1.** the student logs into account 2. the system display the student page 3. The student click **Chat Area** button 4. The system display the chat page 5. The system provide all the message that has been sent before and display message area to send message to other user 7. Students chat with each other by typing message and use Send button. 8. Use Case end
Alternative Course of Action	**Alternate Course A**: Student have invalid account A.2. If the student does not have a valid account. A.3. The system displays error message. A.4. Use case end.

4.2. Object Model

4.2.1. Class Diagram

A **class** diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

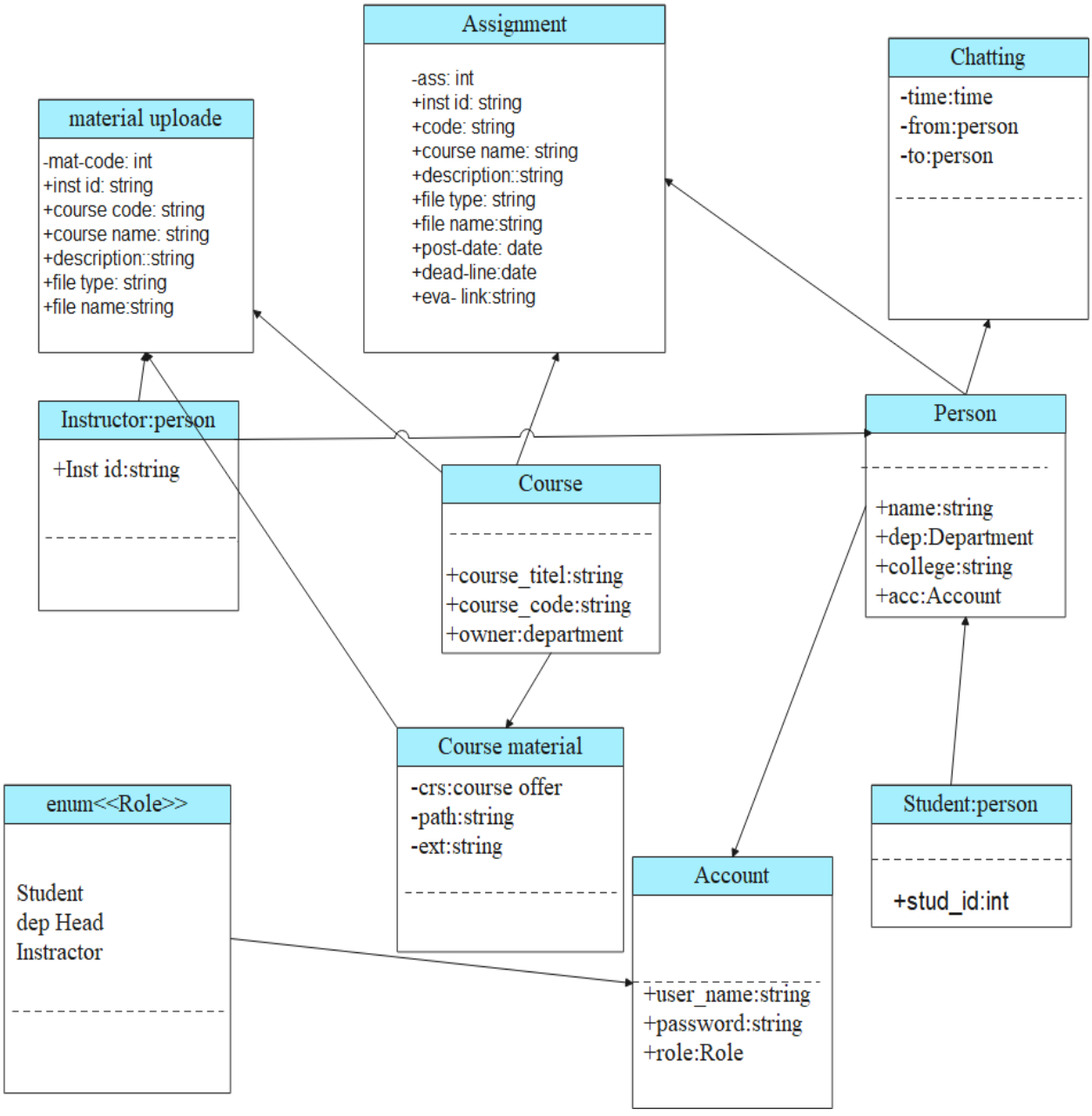


Figure 3: Class Diagram: Class Diagram

4.2.2. Data Dictionary

Data Dictionary is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

Database Name: VCRCS

Table:

- Account
- Student
- Instructor
- Course Record
- Course_Material_Record
- Evaluation Record
- Message

Table 14: Persistent database design

No	Table name	Attributes	Data type	Primary key	Foreign key
1.	Account	**User Id**	**Varchar (20)**		
		Password	Varchar (20)		
		First Name	Char (20)		
		Middle Name	Char (20)	User Id	
		Last Name	Char (20)		
		Sex	Char (20)		
		Role	Enum		

2.	Student	**Student Id** **Course Code** Password First Name Middle Name Last Name Sex Age Phone Number City Region Nationality Year	**Varchar (20)** Varchar (20) Char (20) Char (20) Char (20) Varchar (20) Int Int Char (20) Char (20) Char (20) Char (20)	Student_Id	Course_Code
3.	Instructor	**Instructor Id** Password First Name Middle Name Last Name Sex Experience Phone Number	**Varchar (20)** Varchar (20) Char (20) Char (20) Char (20) Varchar (20) Varchar (30) Int	Instructor_Id	Course_Code

		Course Code Region Nationality	**Char (20)** Char (20) Char (20)		
4.	Message	**Registrar Id** **Student_Id** Content E-mail Date **Message_Id**	**Varchar (20)** Varchar (20) Varchar (100) Varchar (30) Varchar (30) Varchar (20)	Message_Id	Registrar_Id Student_Id
5.	Course_Record	**Course Code** Course Name Credit Hour **Instructor Id** Year Semester	**Varchar (20)** Char (20) Varchar (20) **Varchar (20)** Char (20) Char (20)	Course_Code	Instructor_Id
6.	Course_Material_Record	Course Material Name **Course Material Code** Course Name **Course Code** Course Material Description **Instructor Id**	Char (20) **Varchar (20)** Char (20) **Varchar (20)** Varchar (50) **Varchar(20)**	Course Material _Code	Course_Code Instructor_Id

4.3. Dynamic Model

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. Virtual class room **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 interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the system.

Login sequence diagram

Login sequence diagram shows a sequence of activities that happen when user login in to the system.

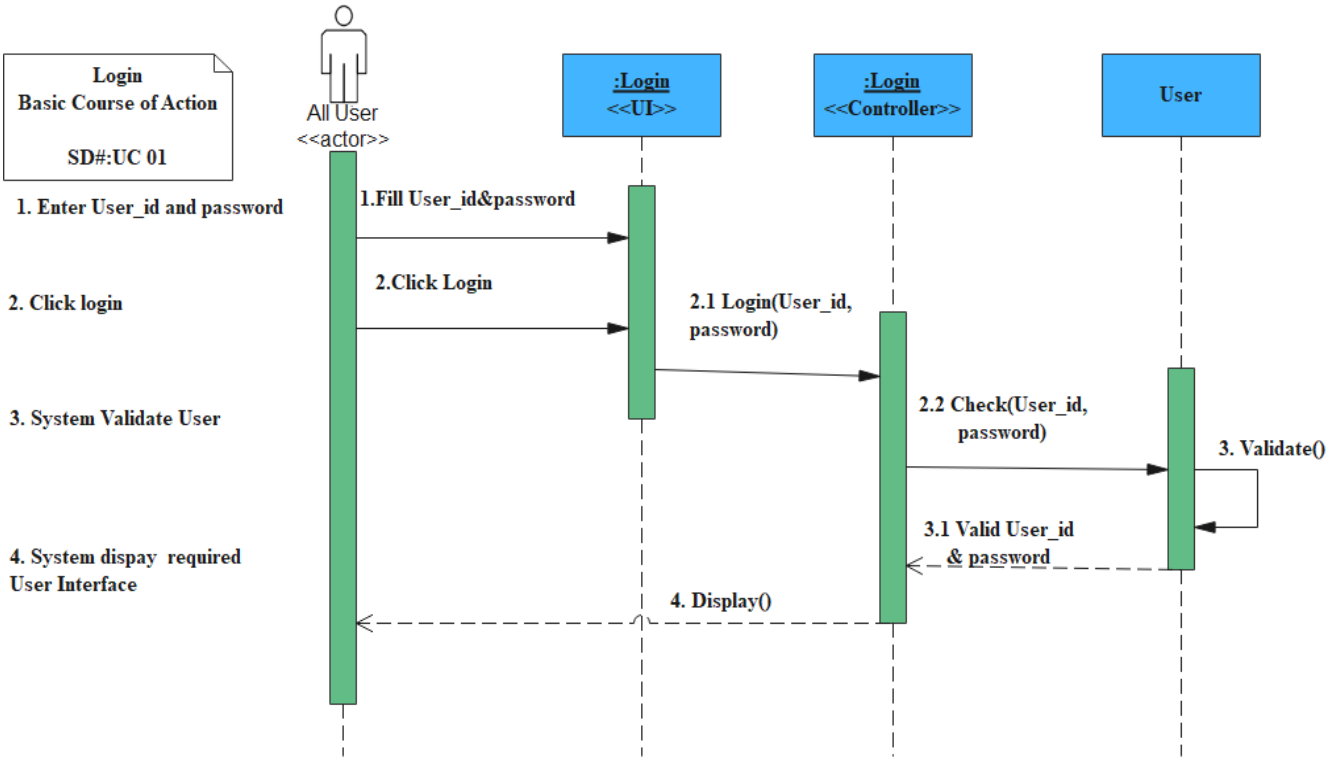


Figure 4: Login sequence diagram

Add Course Sequence Diagram

This sequence diagram shows the sequence of activities that happen when the Department head is adding a new course to the system.

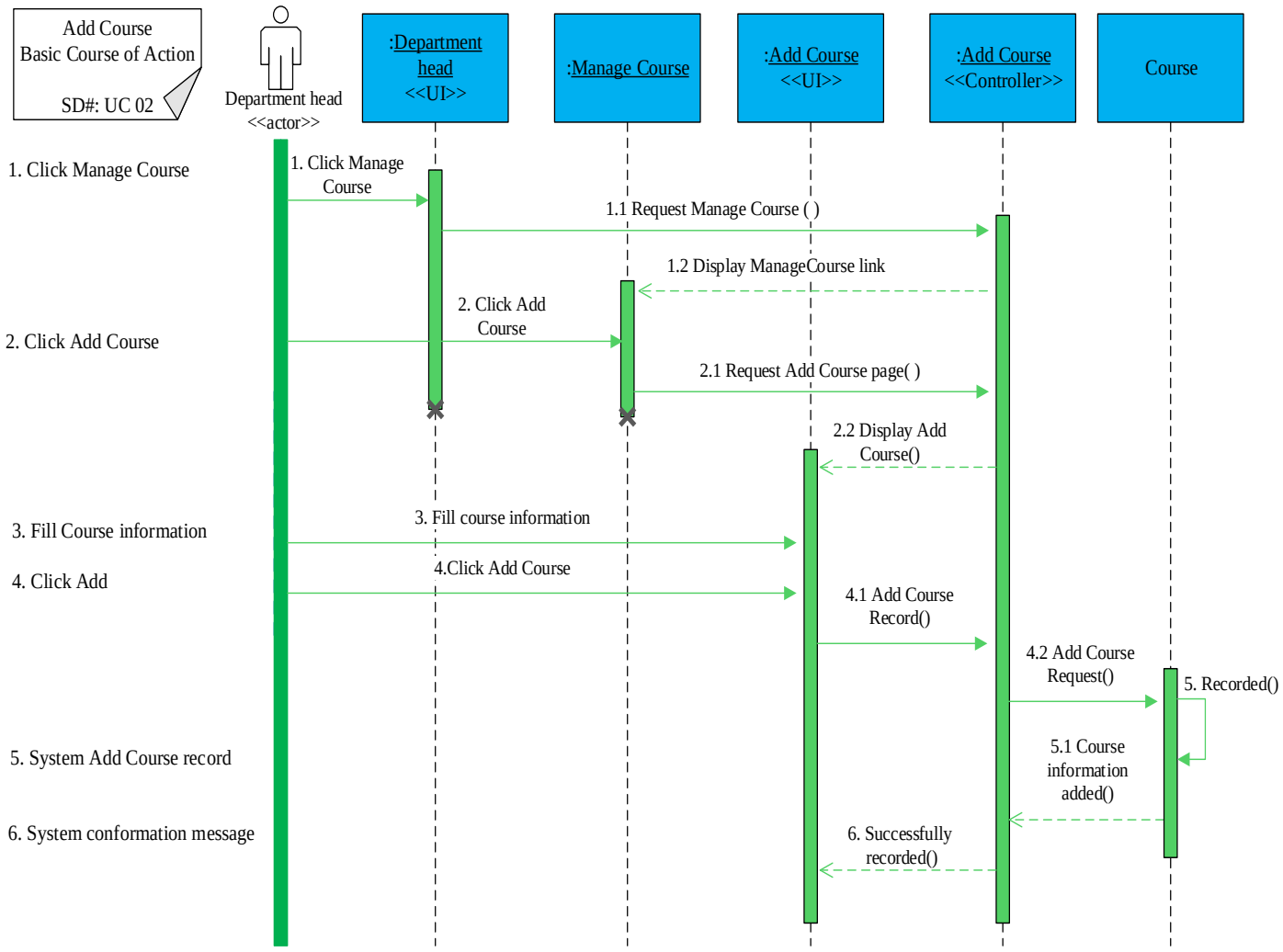


Figure 5: Add Course Sequence Diagram

Delete Course Sequence Diagram

This sequence diagram is to show the sequential activities to delete the existing course form the system.

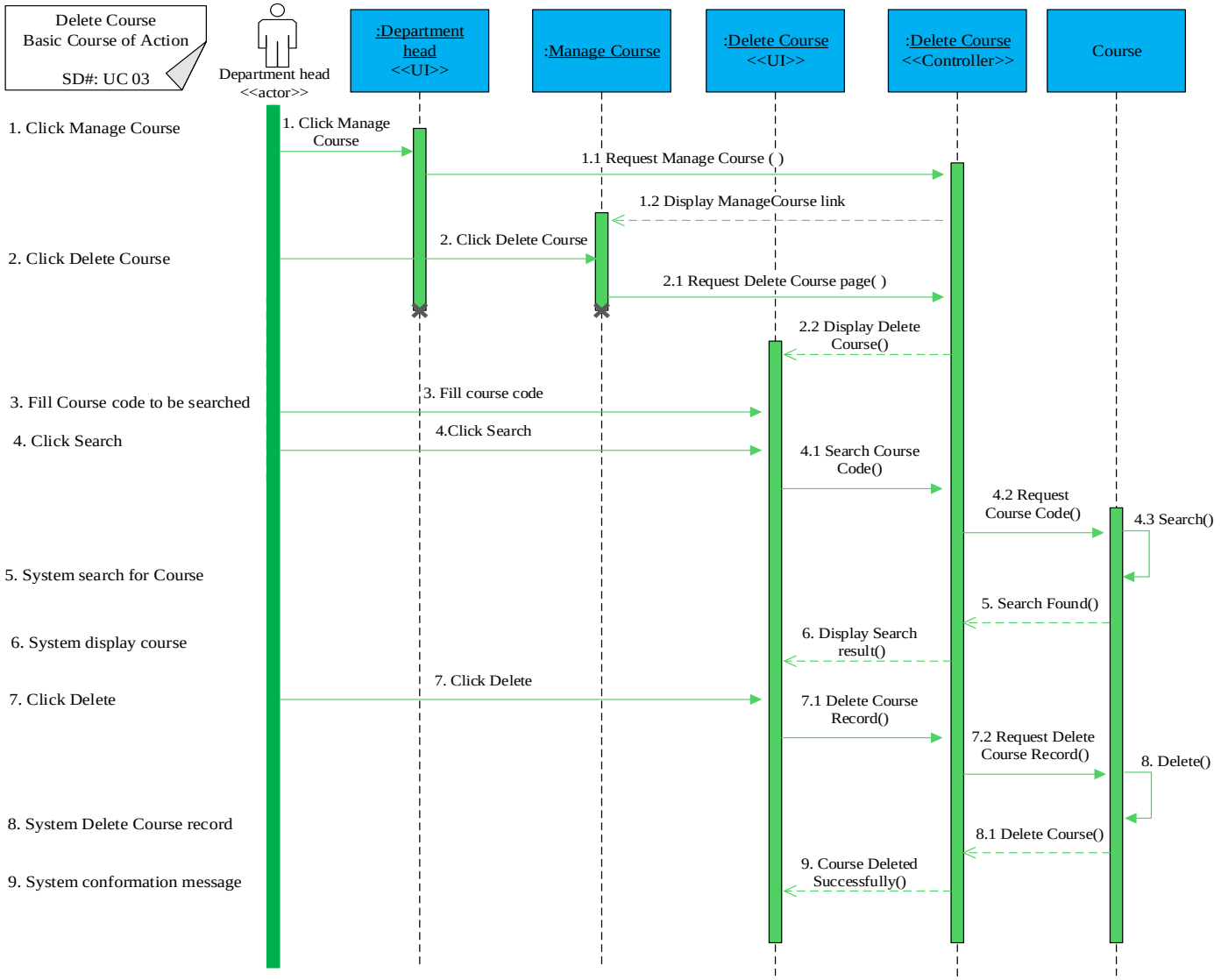


Figure 6: Delete Course Sequence Diagram

Update user Profile sequence diagram

This sequence diagram shows the sequential flow of updating user profile.

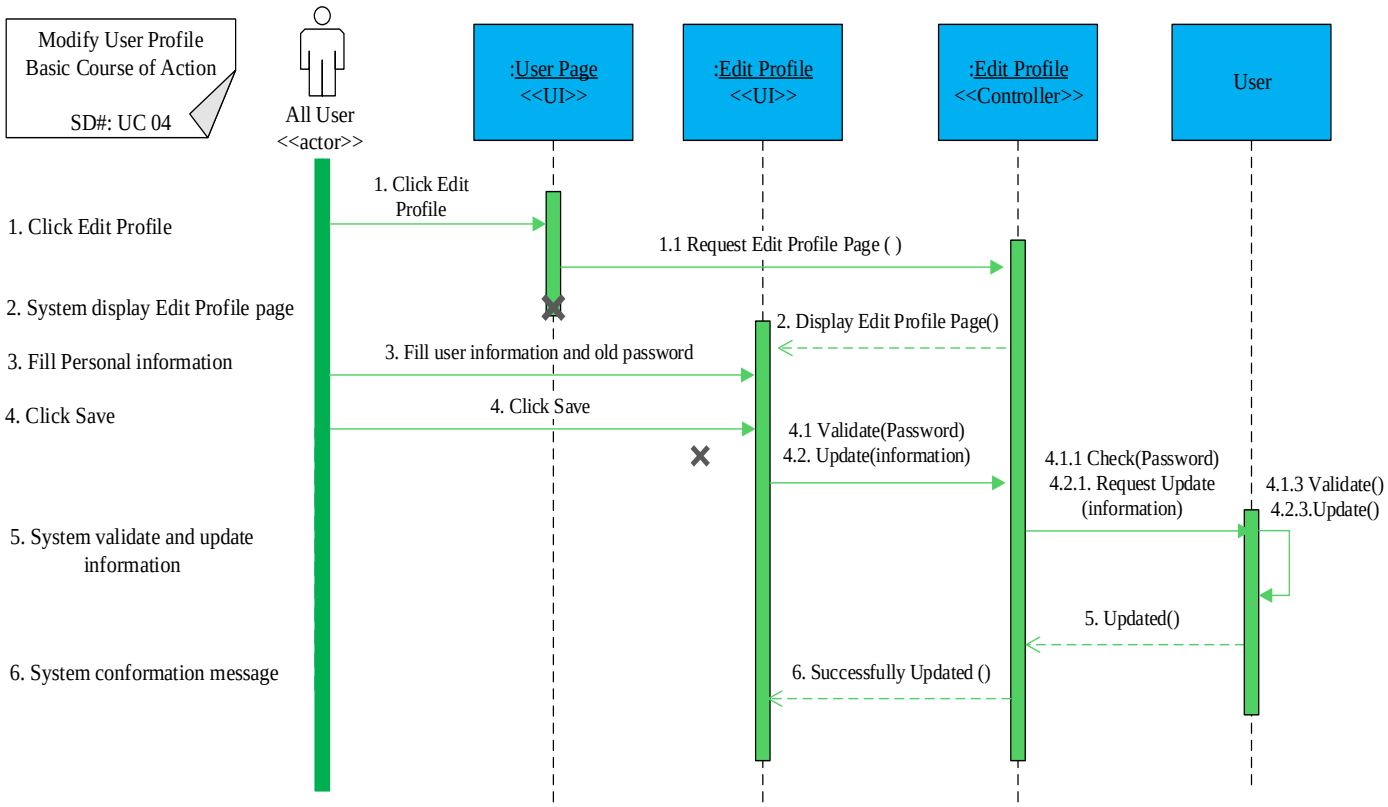


Figure 7: Update user account sequence diagram

Add instructor record sequence diagram

This sequence diagram shows the sequential activities to add new instructor to the system.

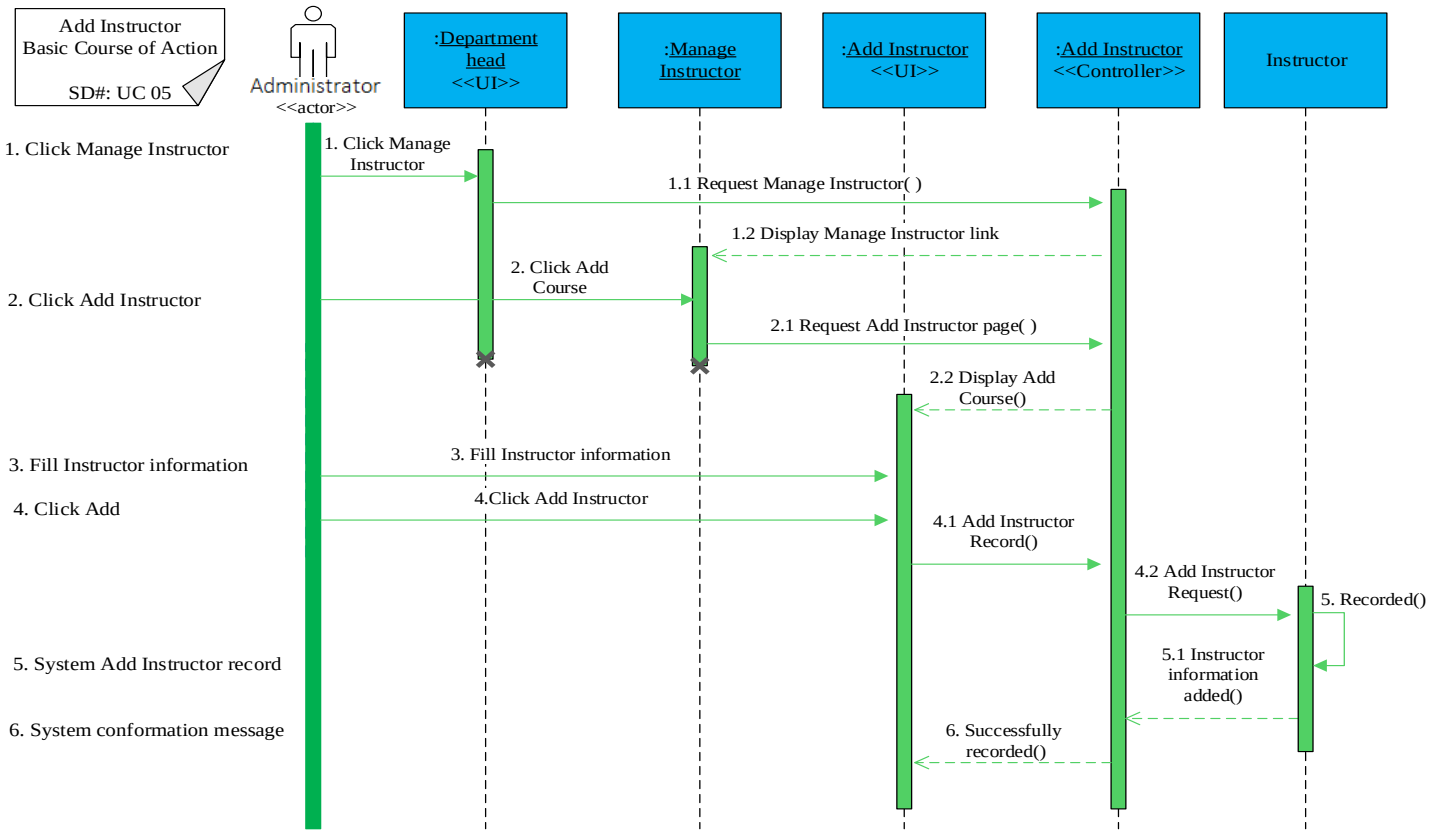


Figure 8: Add instructor record sequence diagram

Delete Instructor Sequence Diagram

This diagram shows the sequential flow of activities to delete the existing instructor's record form the system.

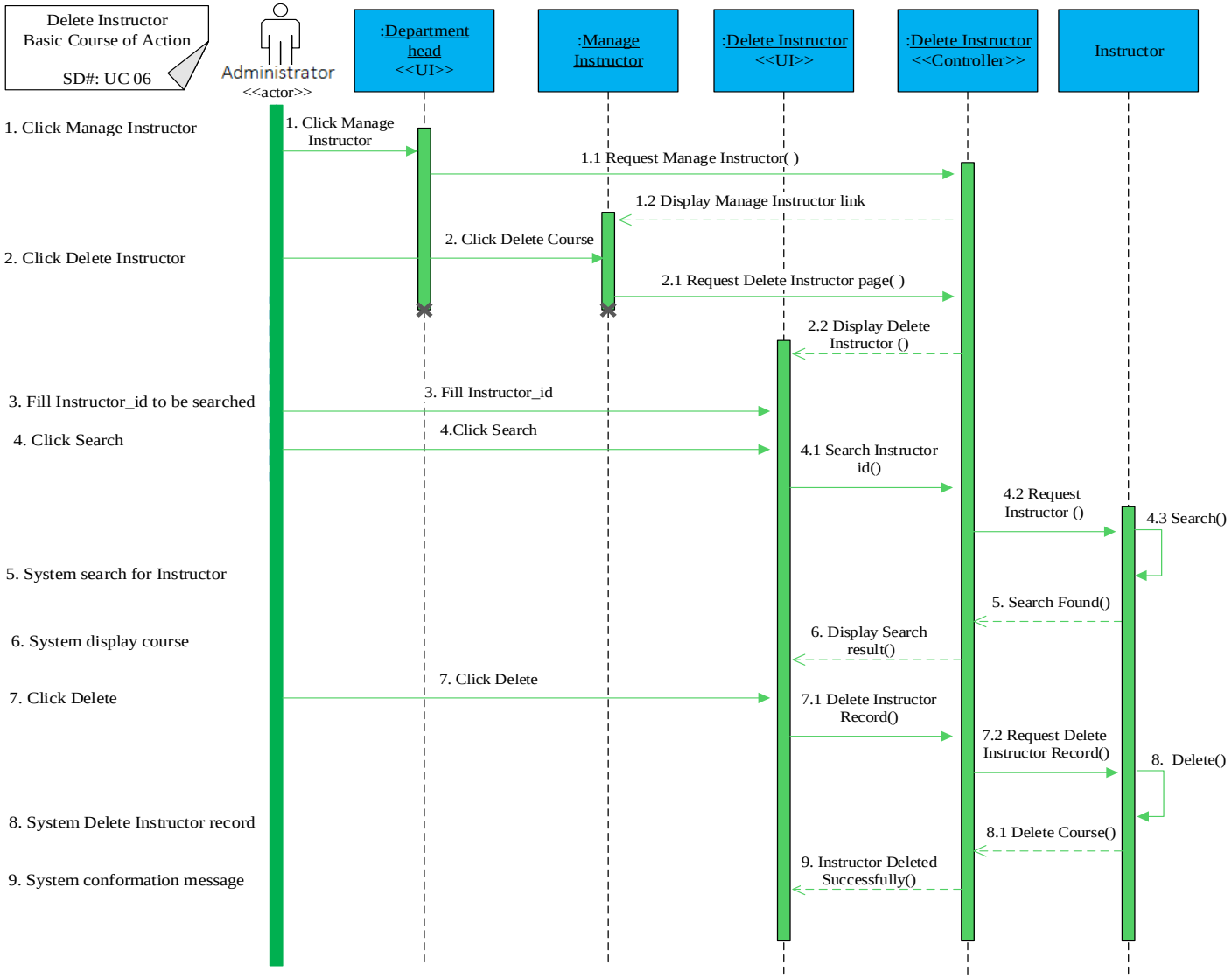


Figure 9: Delete Instructor Sequence Diagram

Update Instructor Record

This diagram shows how to update the instructor's record sequentially.

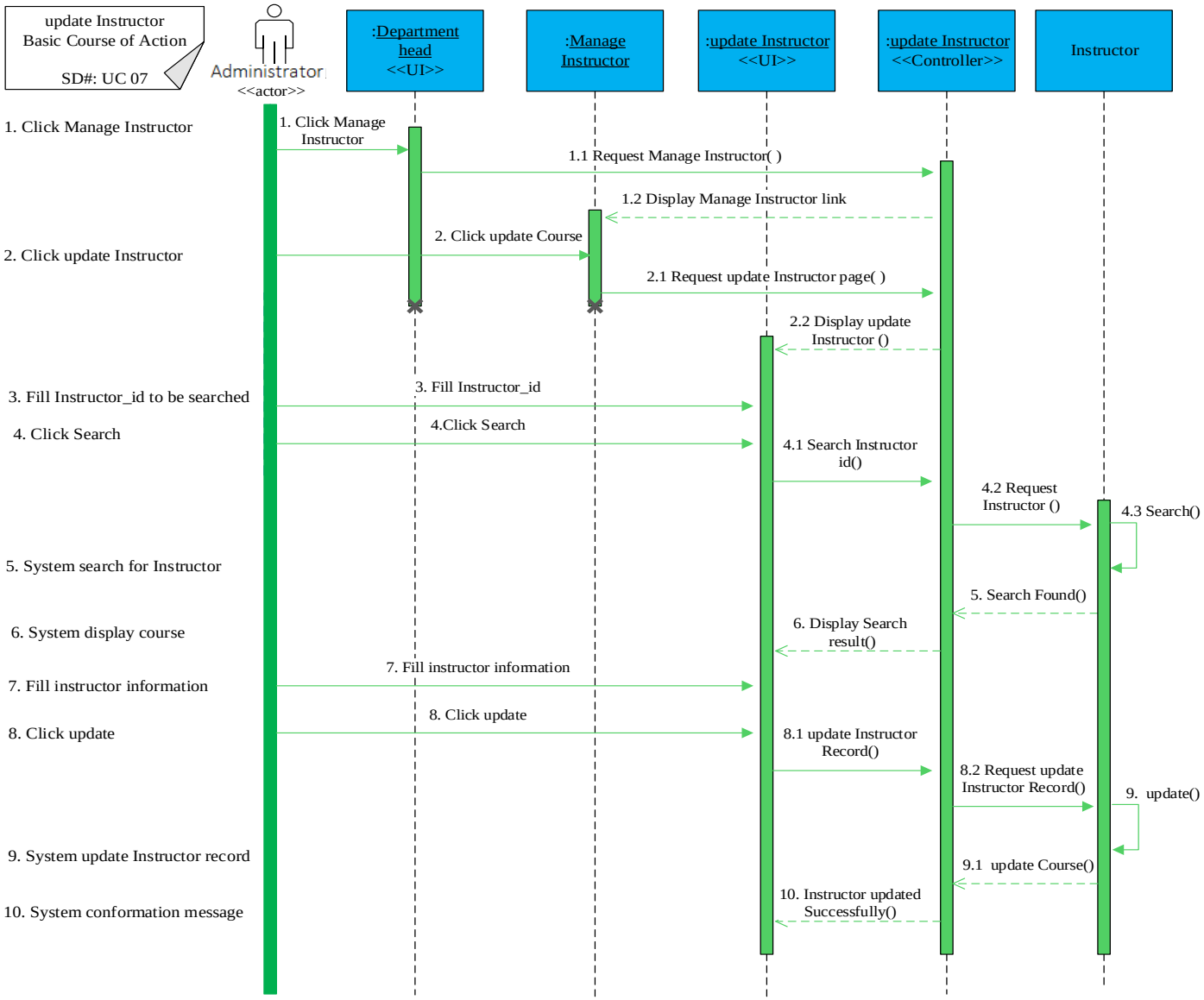


Figure 10: Update Instructor Record

Follow up course sequence diagram

This diagram shows the sequential activities to follow up or activate the semester course or lecture.

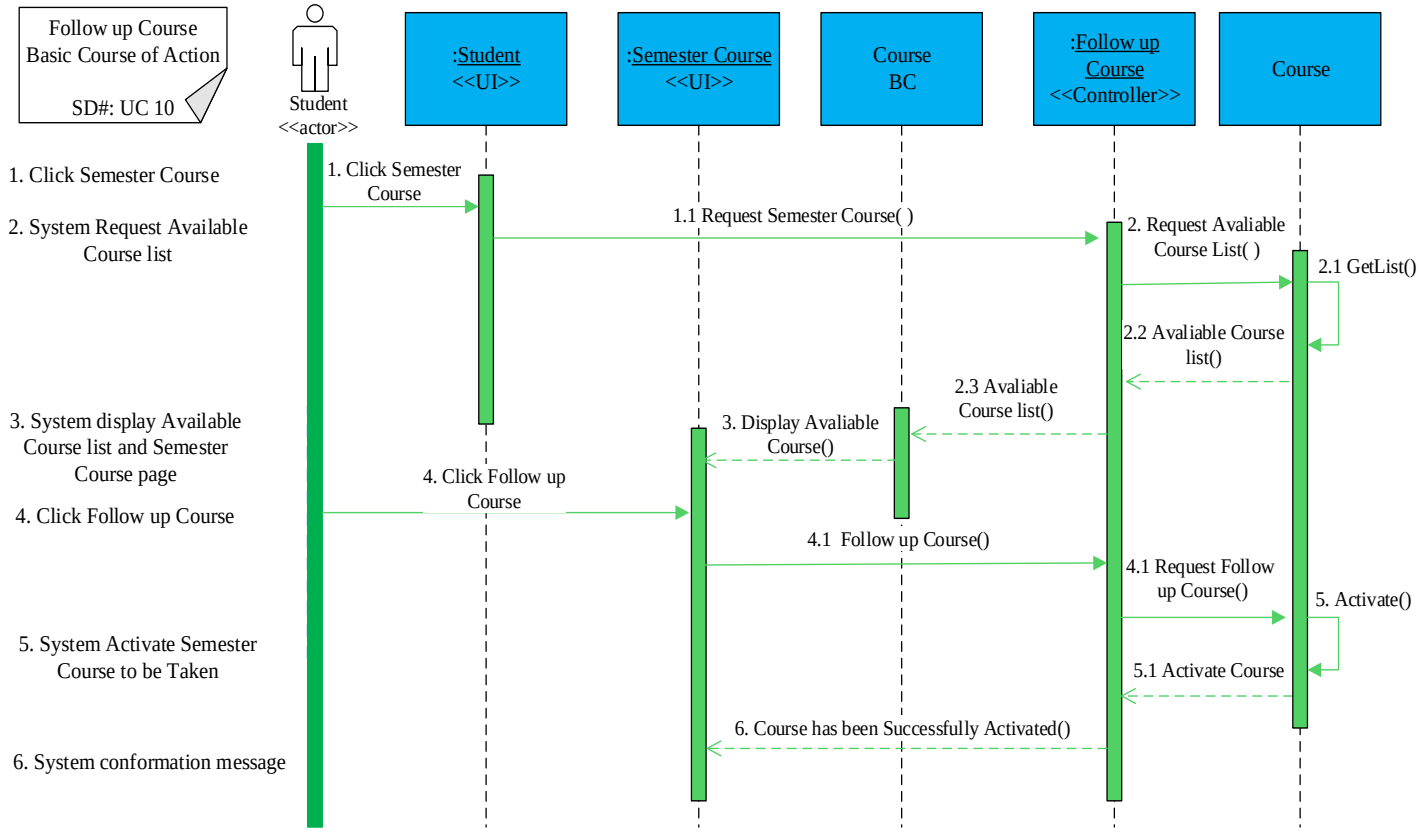


Figure 11: Follow up course sequence diagram

Add course material sequence diagram

This diagram helps to show the sequential flow of activities to add course material to the existing course in the system.

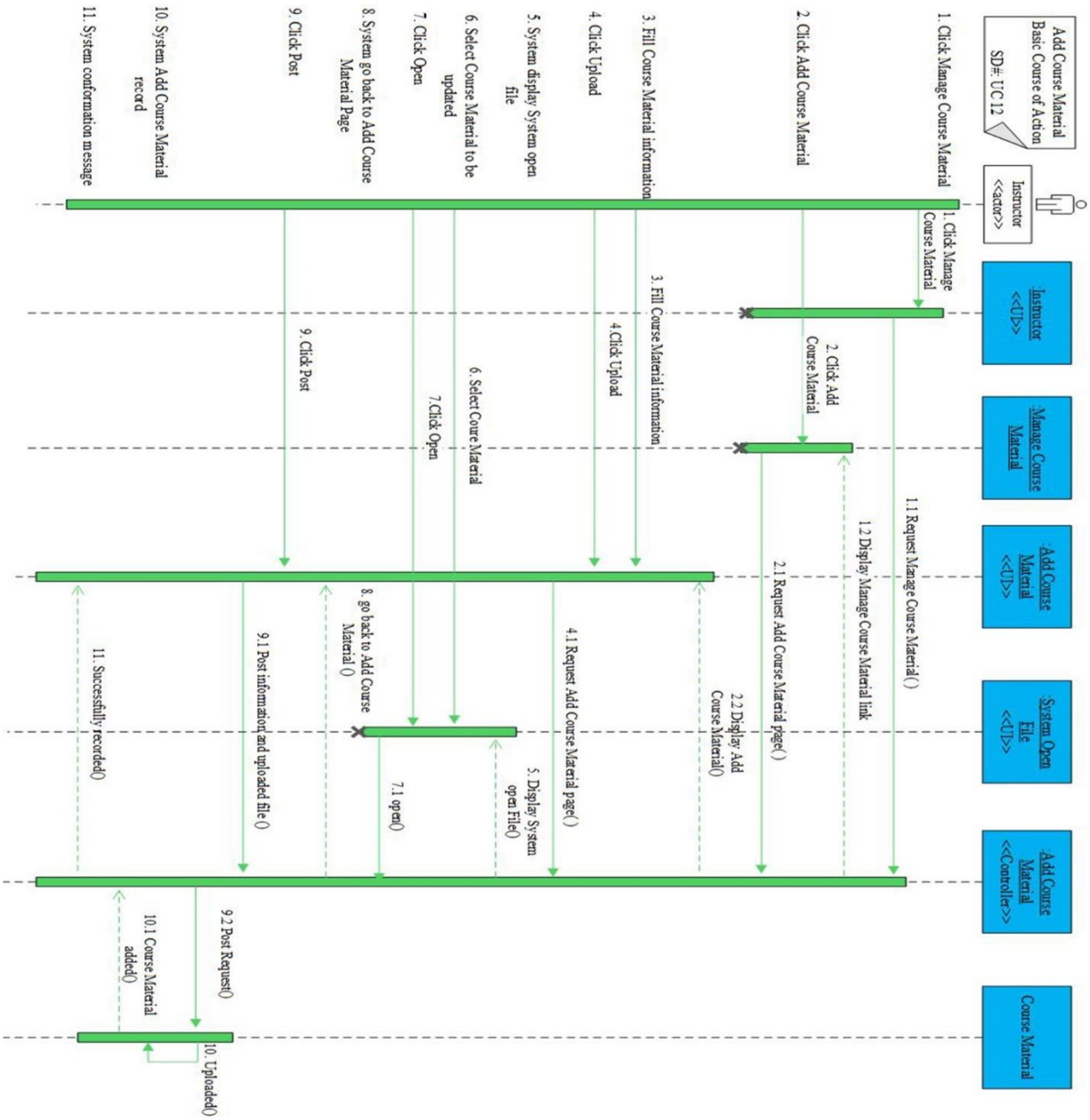


Figure 12: Add course material sequence diagram

Delete Course material

This sequence diagram shows the sequential activities to delete a course material from the system.

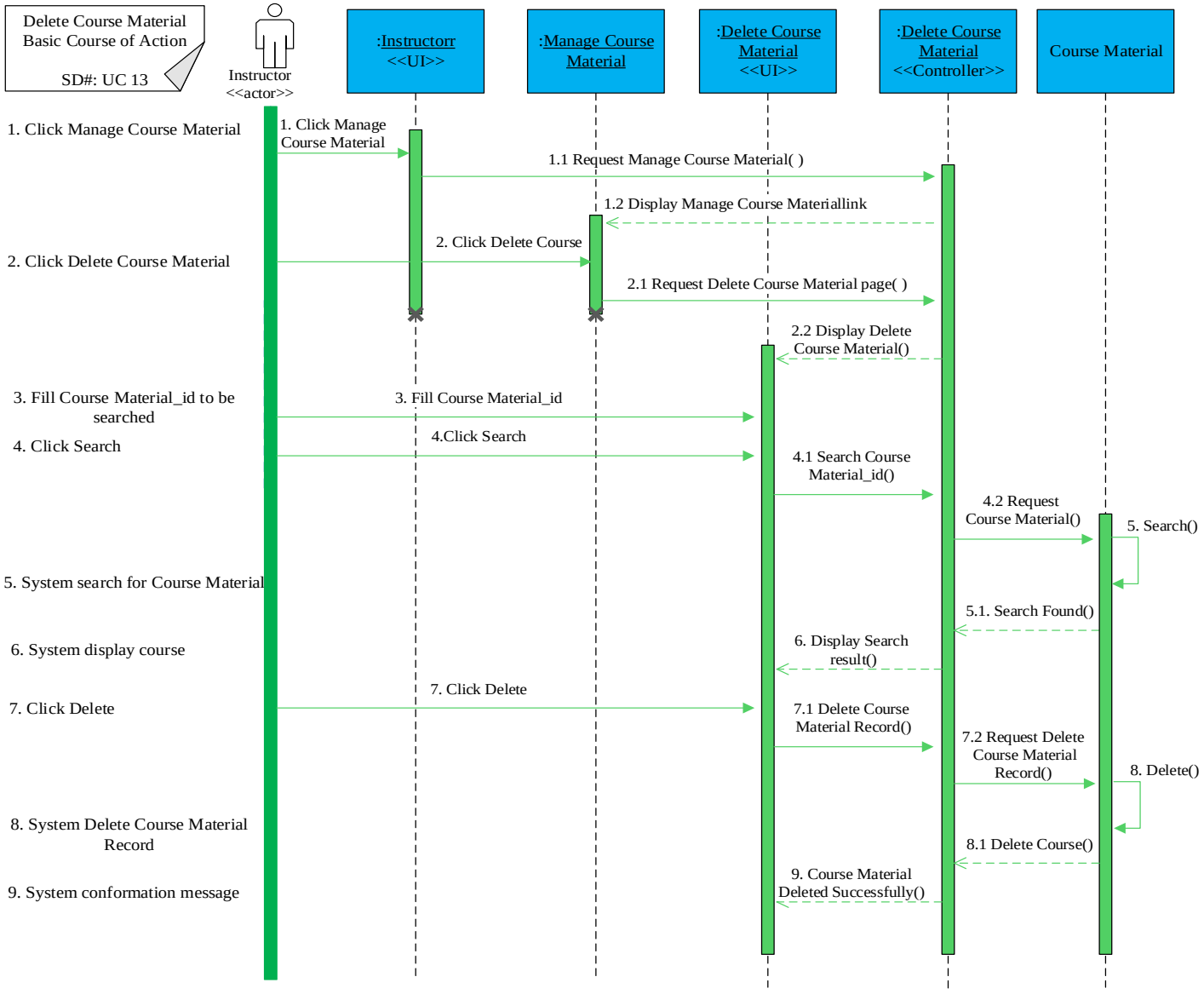


Figure 13: Delete Course material

Chatting Sequence diagram

This diagram shows sequential steps to withdraw from the campus by different reason if the registrar finds it essential.

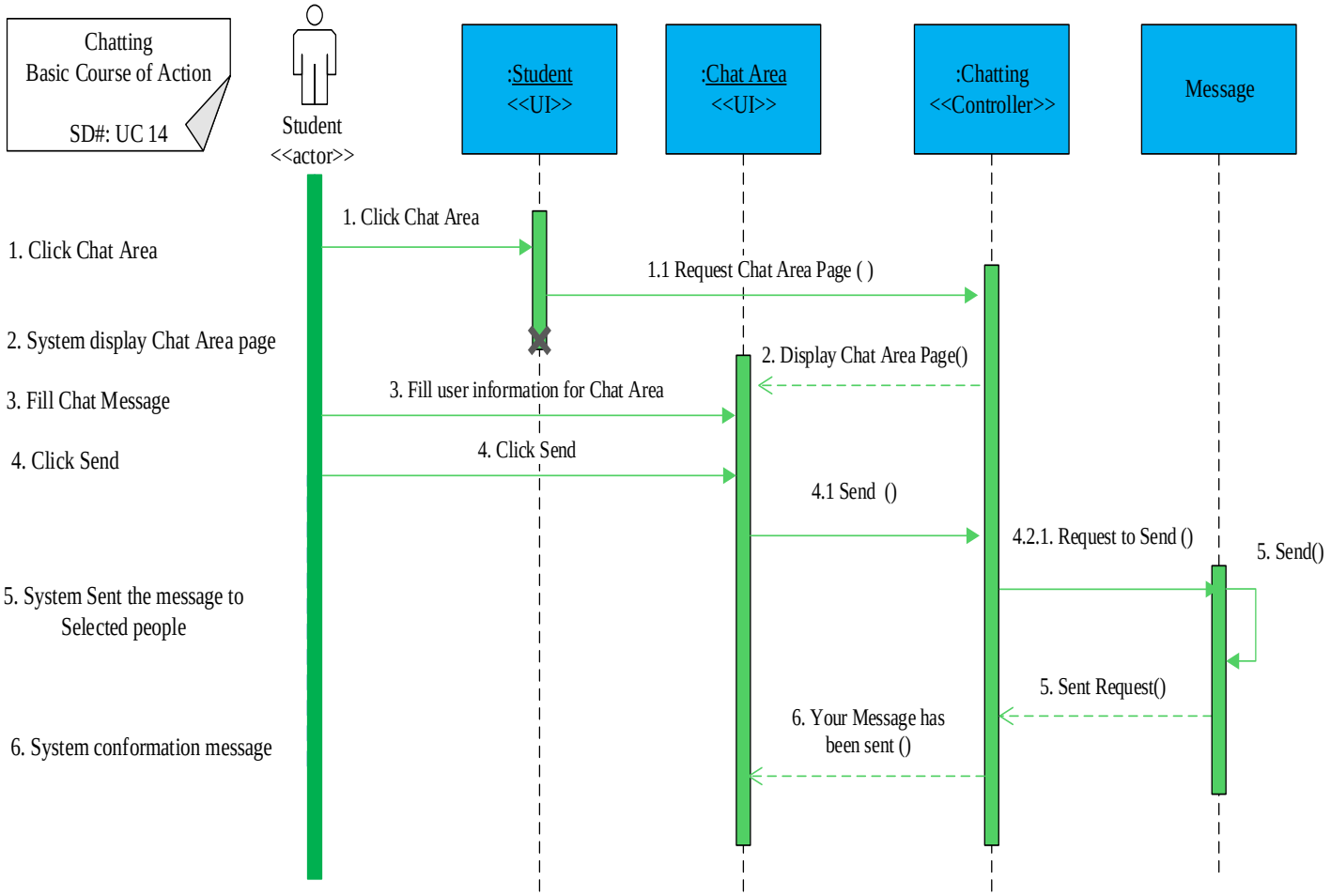


Figure 14: User Chatting Sequence diagram

4.3.2. Activity Diagram

An Activity diagram focuses on the flow of activities involved in a single process. The Activity diagram shows how these single-process activities depend on one another.

Log in activity diagram

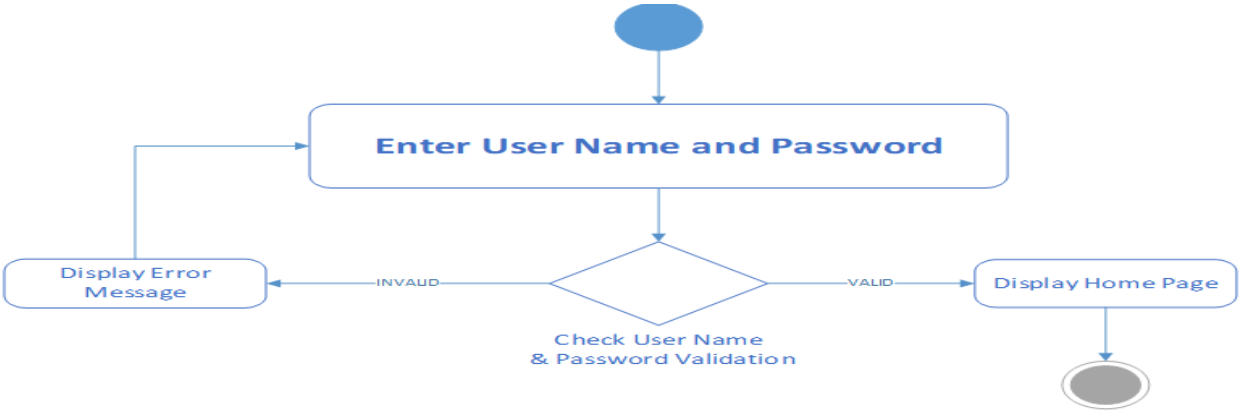


Figure 15: Log in activity diagram

Modify User account (Changing Password) activity diagram

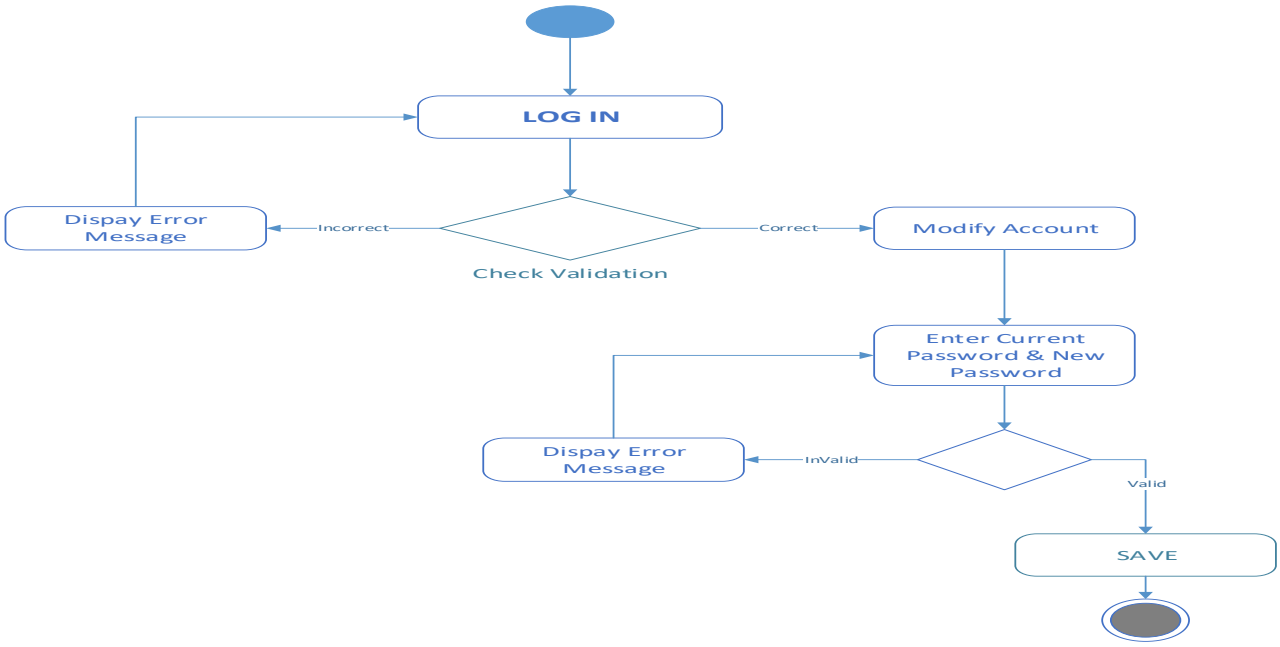


Figure 16: Modify User account (Changing Password) activity diagram

Add Course Record Activity Diagram

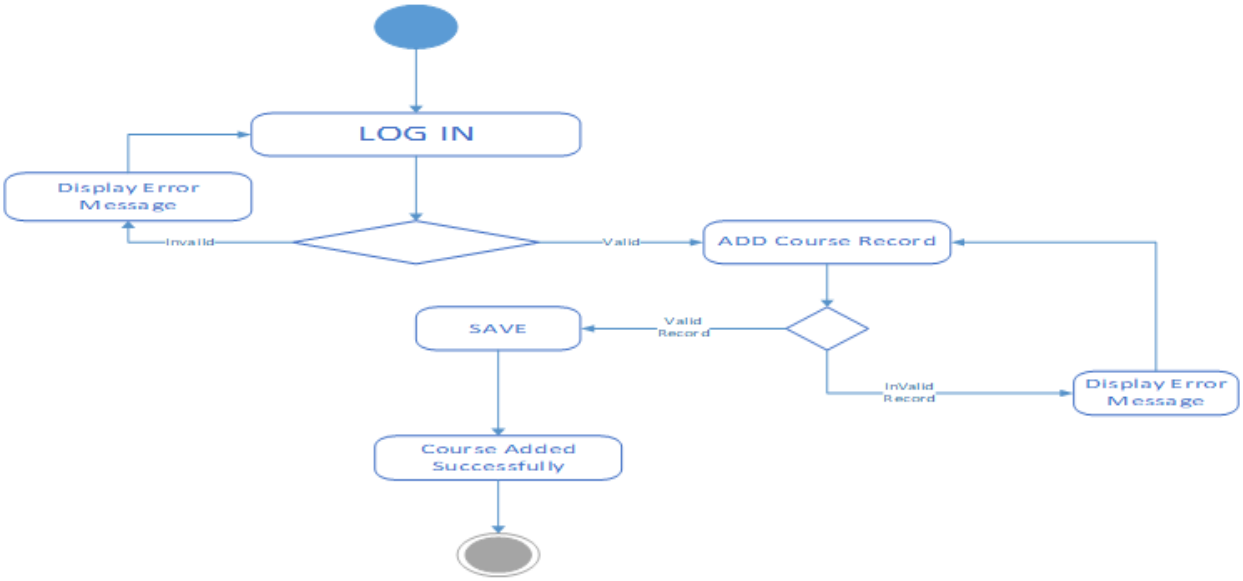


Figure 17: Add Course Record Activity Diagram

Delete course record activity diagram

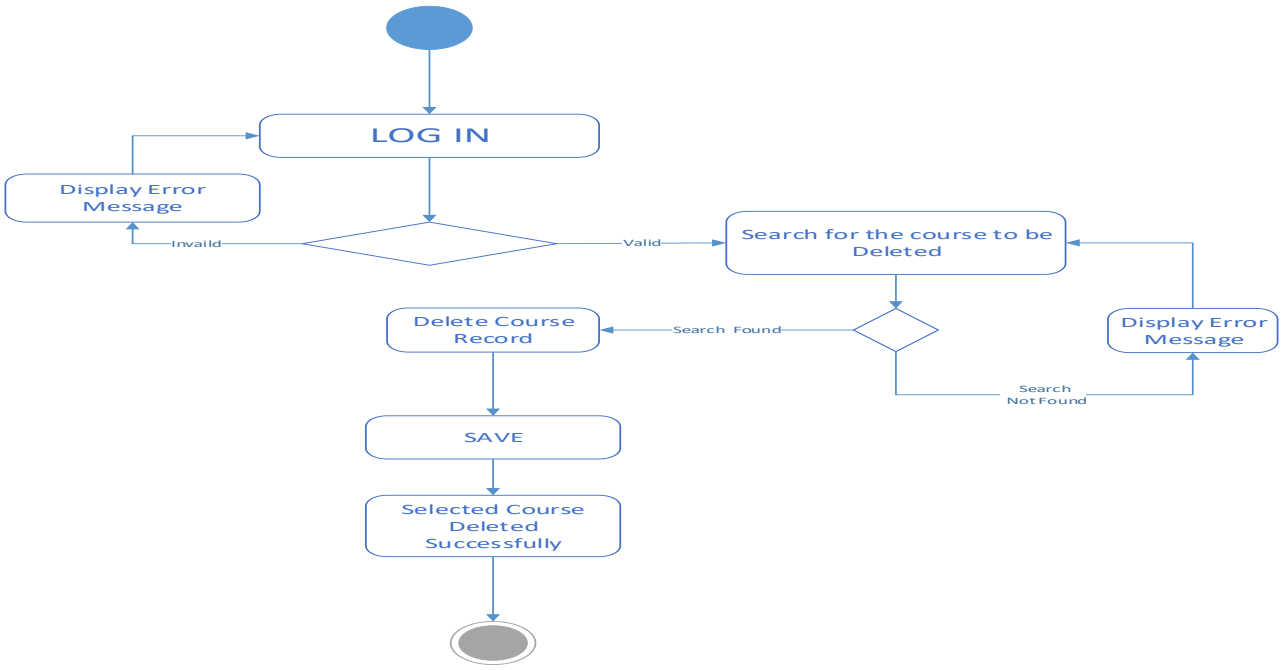


Figure 18: Delete course record activity diagram

Update course record activity diagram

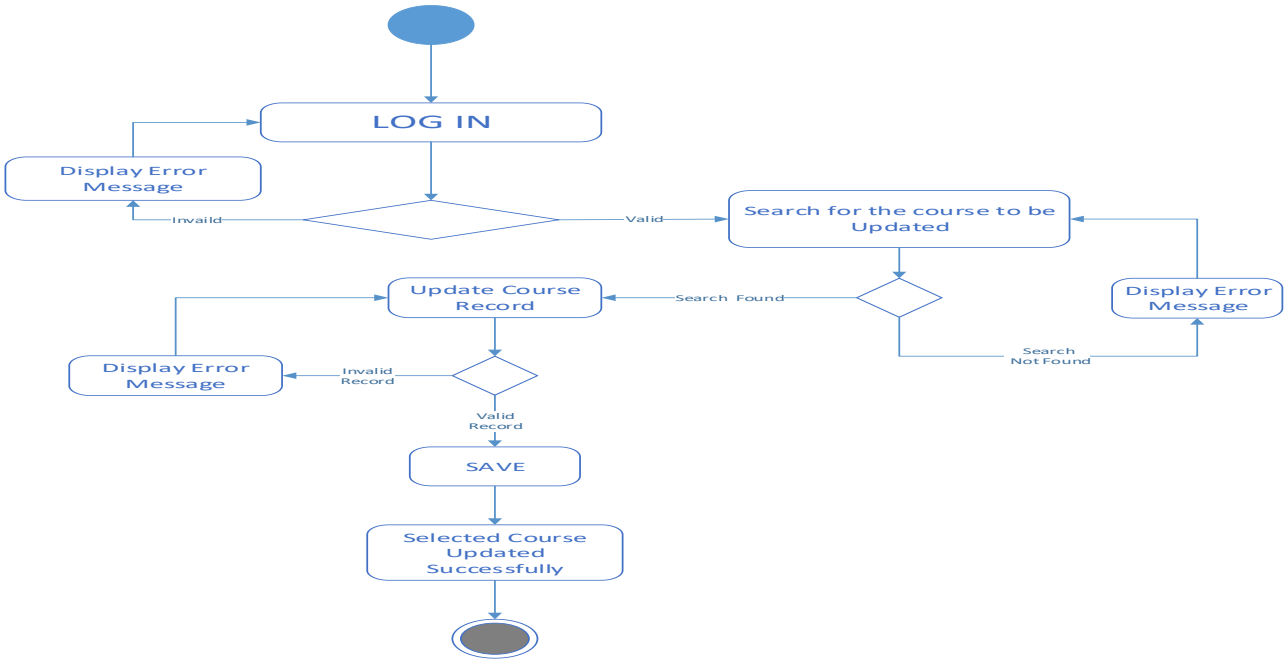


Figure 19: Update course record activity diagram

Add Instructor Record activity diagram

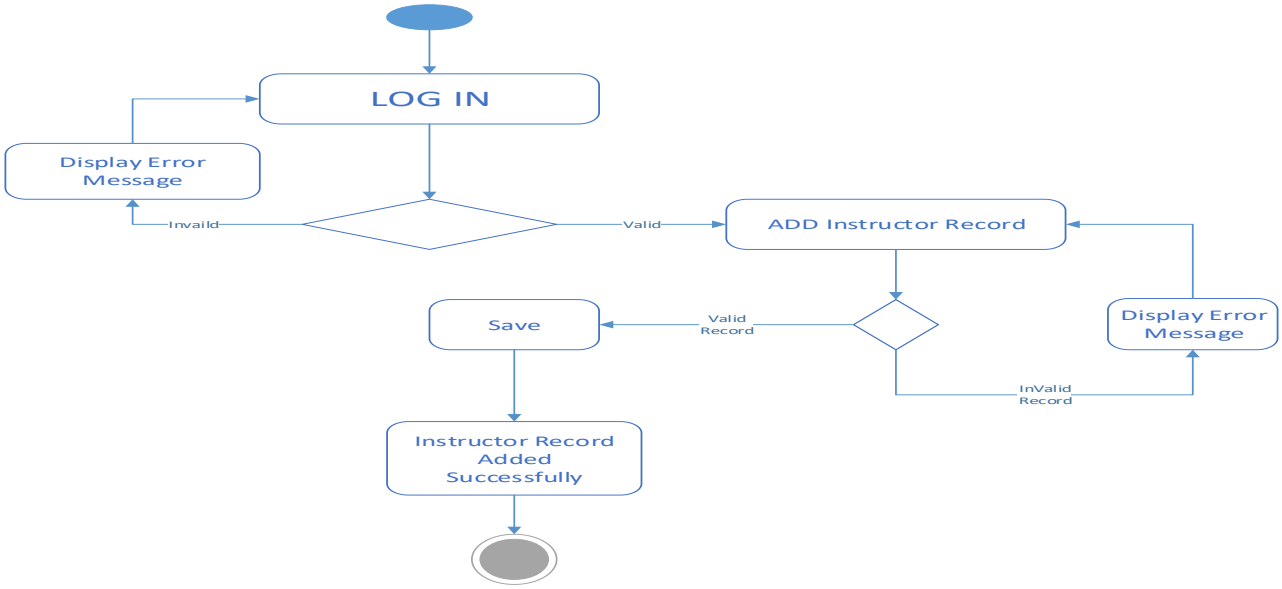


Figure 20: Add Instructor Record activity diagram

Delete instructor record activity diagram

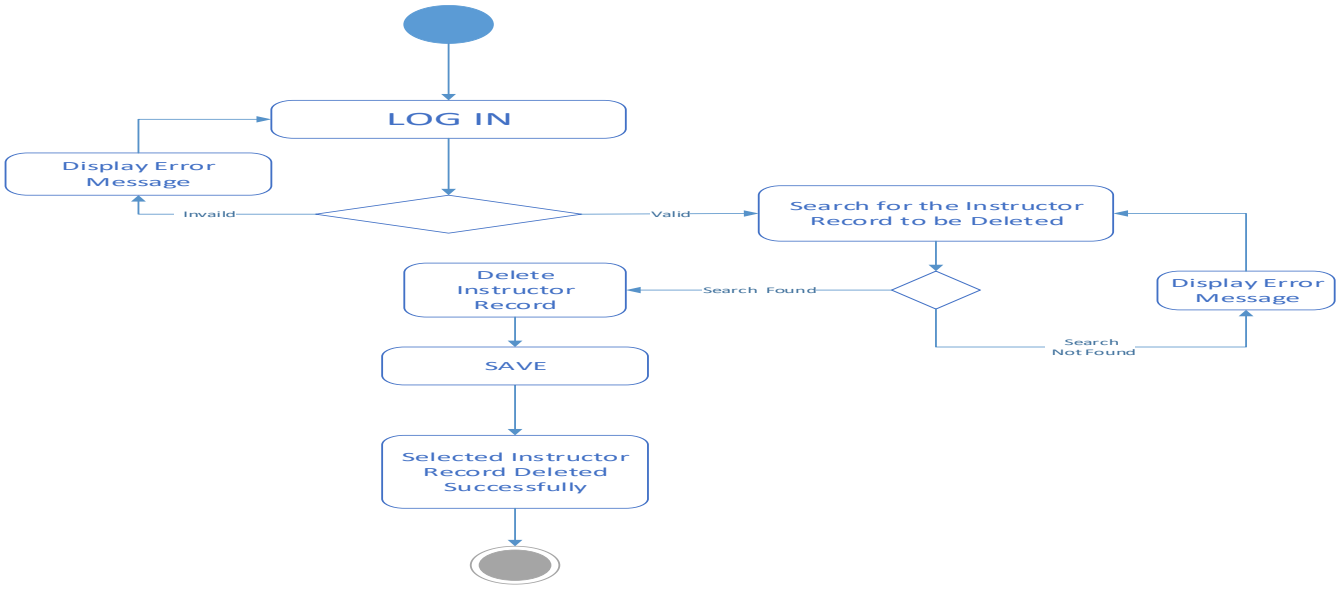


Figure 21: Delete instructor record activity diagram

Update instructor Record activity diagram

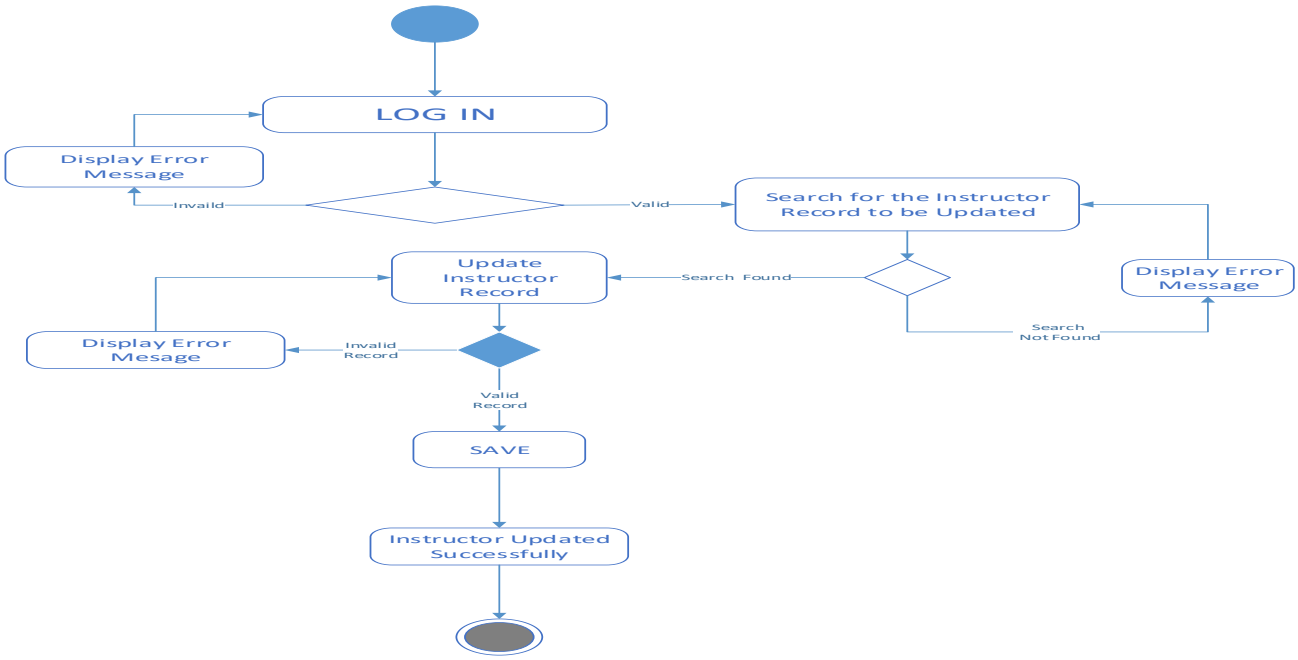


Figure 22: Update instructor Record activity diagram

Search activity diagram

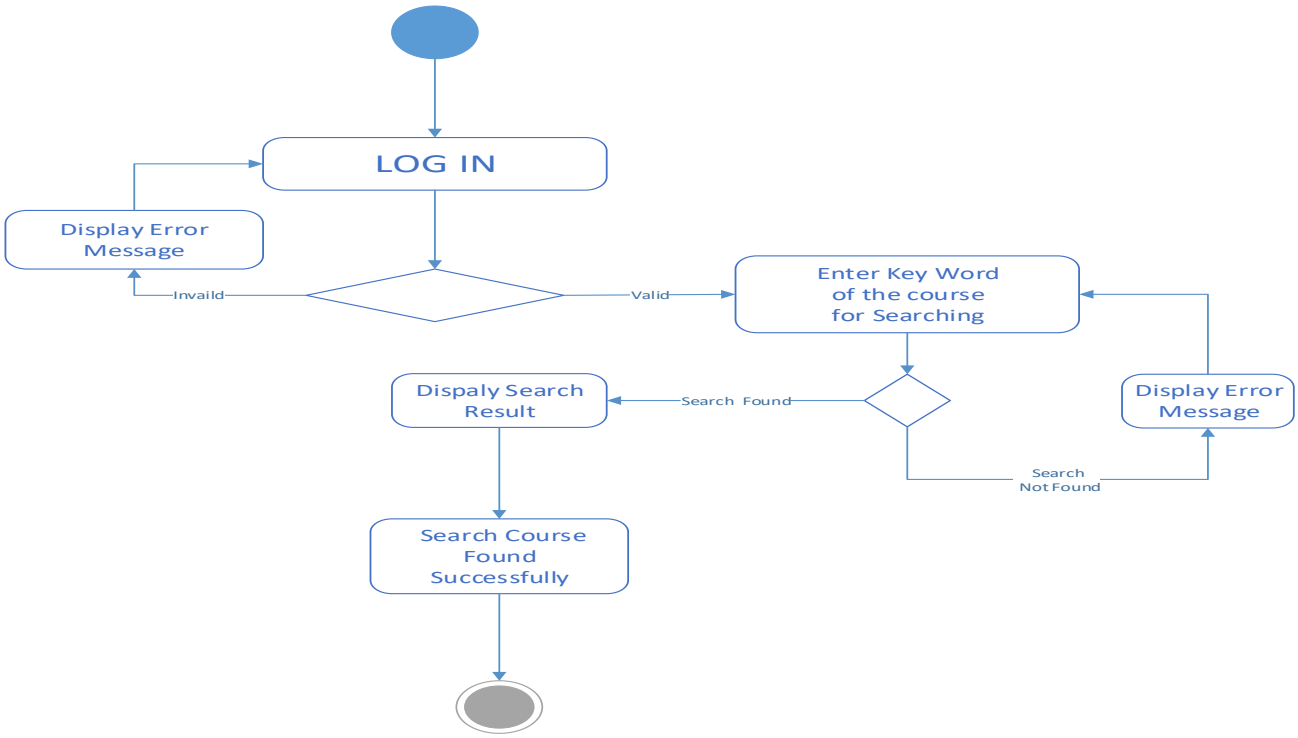


Figure 23: Search activity diagram

Add course material activity diagram

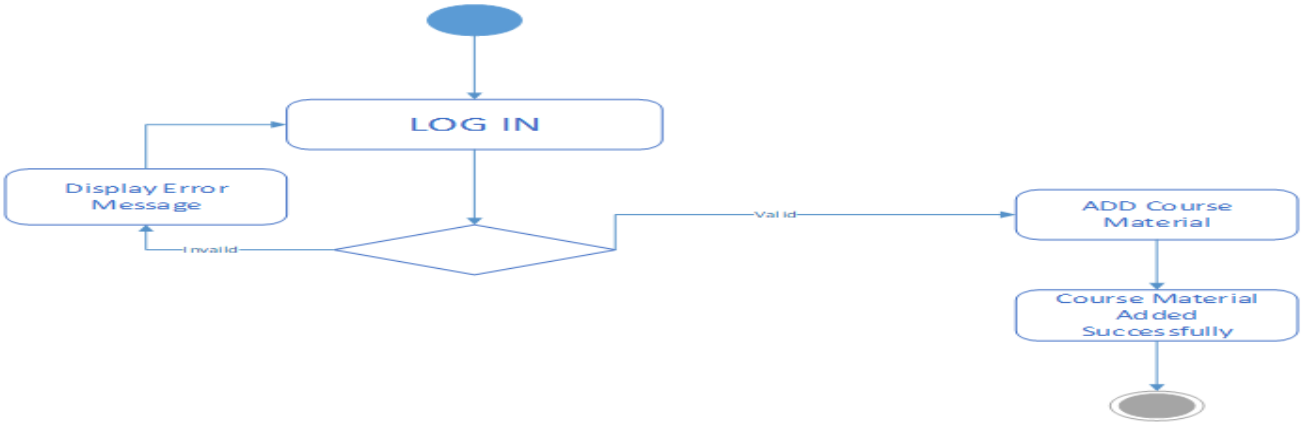


Figure 24: Add course material activity diagram

Delete course material activity diagram

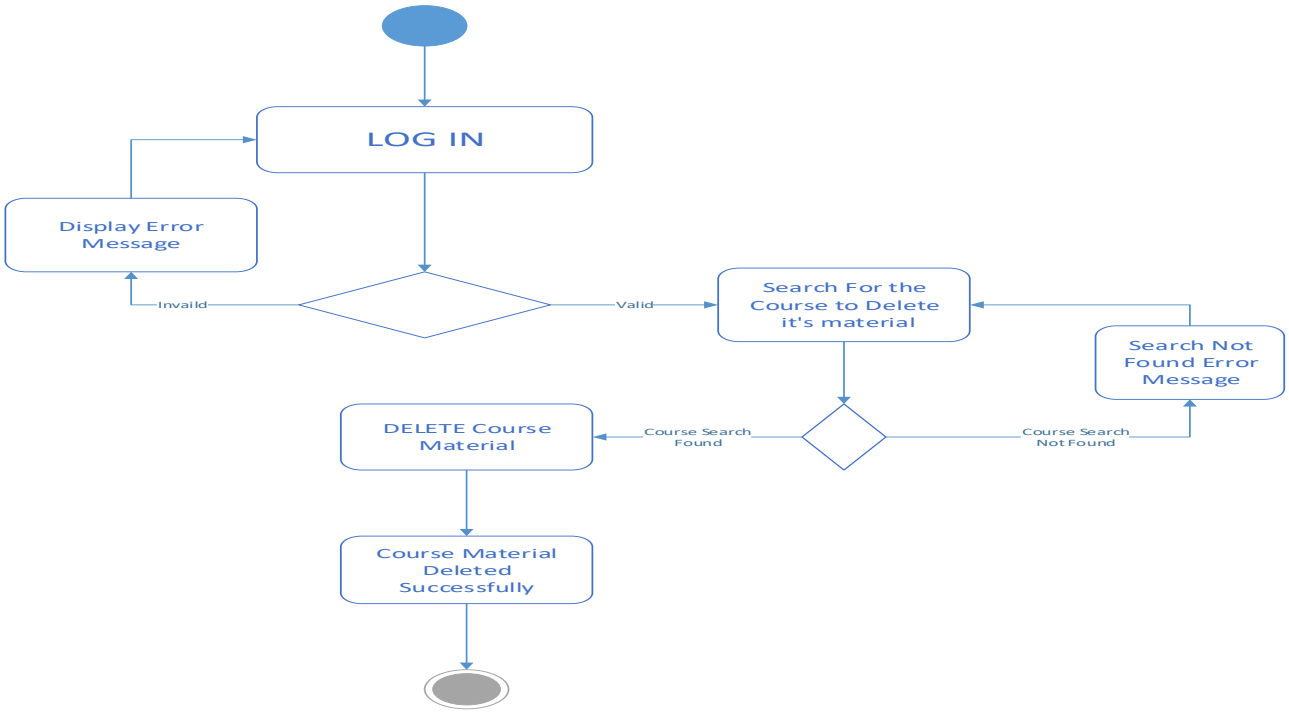


Figure 25: Delete course material activity diagram

Chatting activity diagram

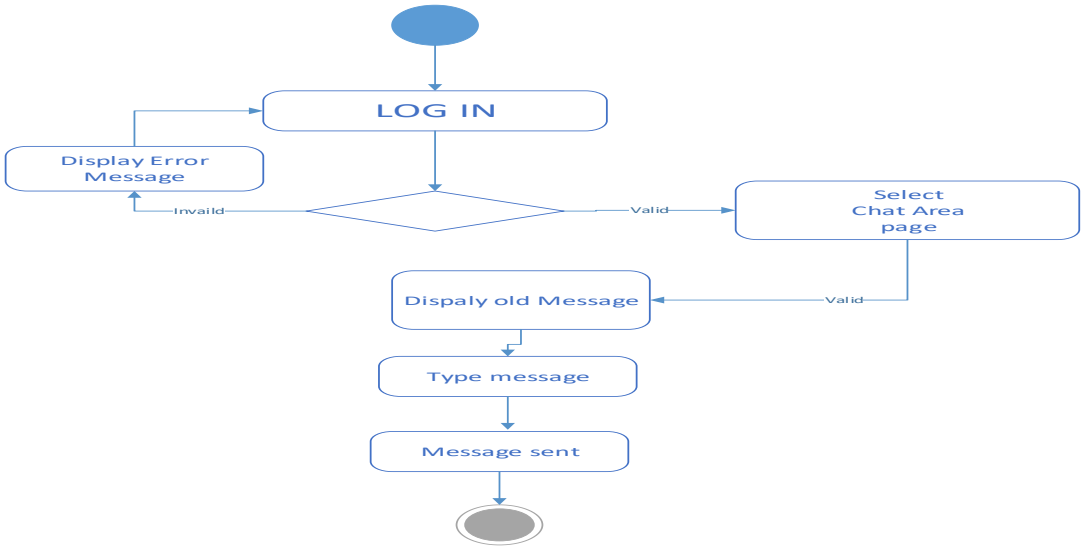


Figure 26: Chatting activity diagram

4.3.3. State chart diagram

A diagram that captures the behavior of an object by specifying the sequence of states it goes through during its lifetime in response to events; together with the responses to those events describe changes in state in a procedure also called state chart diagram or state machine diagram.

Follow the overall course state chart

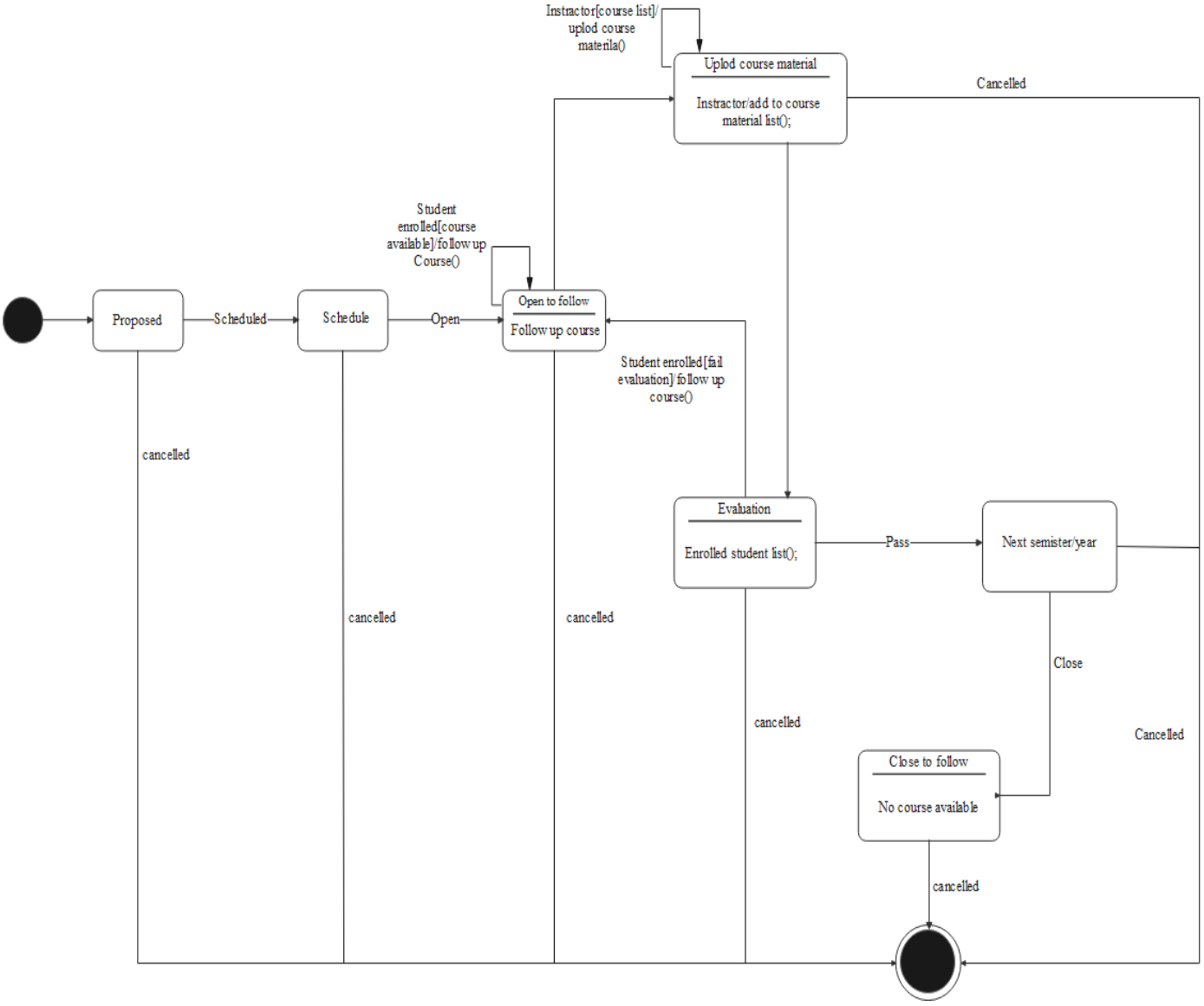


Figure 27: Course state chart diagram

CHAPTER 5

5. SYSTEM DESIGN

System design is the process of defining the components, modules, interfaces, and data for a system to satisfy specified requirements. Also system design is the process of creating or altering systems, along with the processes, practices, models, and methodologies used to develop them. Our system design is proposed solution for existing system problem. Our proposed system is divided into different categories in order to simplify the complexity of system when we providing a solution for the specific problem [4].

5.1. Design Goal

The objective of designing is to model a system and to provide a high-level overview of the system architecture, and describe the qualities of the system that the developers should consider. In order to produce a good system, it is very significant to have a good design and the design must have a goal, which states the needed quality that must incorporated.

Performance

Response time is the total amount of time it takes to respond to a request for service. So, in our system, the service time is the time it takes to do the work you requested generally the response time in our system is mostly short.

Dependability

Dependability is the ability to provide services that can defensibly be trusted within a time-period and also encompass mechanisms designed to increase and maintain the dependability of a system or software. So our system dependability is effectively high.

Maintainability

The system should be maintainable. It facilitates the correction of errors and deficiencies and it can expand or upgraded to satisfy new changes in requirement.

End user

This is to mean that the user of the system after the completion of the system how the system is used friendly manner for both the experienced and inexperienced user using the user interface and this is clearly stated in the user's documentation.

5.2. Proposed system Architecture

Our proposed system is consisting of 3-tier architecture namely Presentation layer, Application layer, and Database layer. 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database.

Presentation layer: is client layer (User Interface) and topmost layer of the application. This is the layer we see when we use this system. It is the interface to our system which takes information from the user. The main functionality of this layer is to communicate with the application layer. This layer passes the information which given in terms of keyboard action and mouse click to the application layer.

Application (Middle) layer: it is application layer which interacts with data layer and sends information retrieved from database toward to presentation layer.it act as the mediator between presentation layer and data layer. From above example once, the user clicks on login button application layer interact with the database and send information towards to presentation layer.

Database (Data) layer: which used to store data entered by the user? In general client of our system use browser to access the system using the internet. In this case when the user enters input and takes certain action application server process client request to interacting with the database server.

Below figure is proposed system architecture:

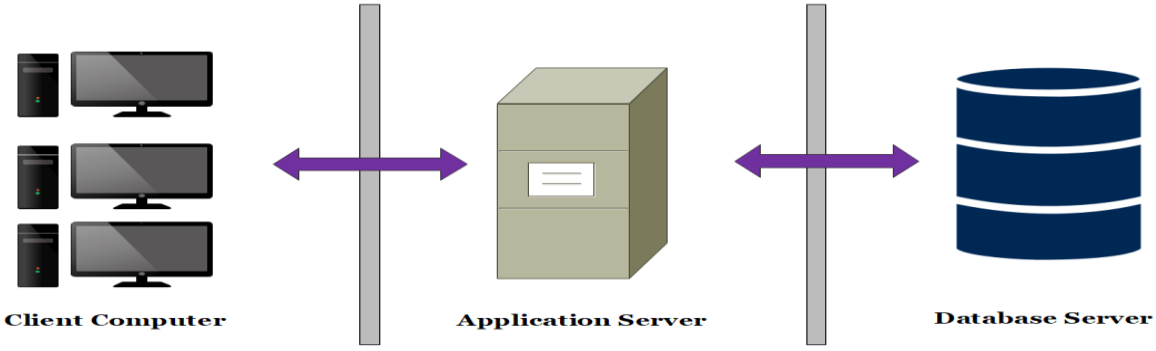


Figure 28: Proposed system architecture

5.2.1. Sub system decomposition and Description

Sub-system decomposition is the processes of breaking down the system into sub-systems, and each sub-systems analysis separately. The main aim is to minimize the complexity of the system. Subsystems could consider as packages holding related classes/objects. The following are sub systems.

User account Management subsystem: this subsystem performs the following tasks.

- Create user account
- user account

Course management subsystem: this subsystem performer the following tasks.

- Add course
- Delete course
- Update course

Instructor management subsystem: this subsystem performs the following operations.

- Add Instructor
- Delete Instructor

Course material management subsystem: this subsystem performs the following operations.

- Upload course material
- Delete course material

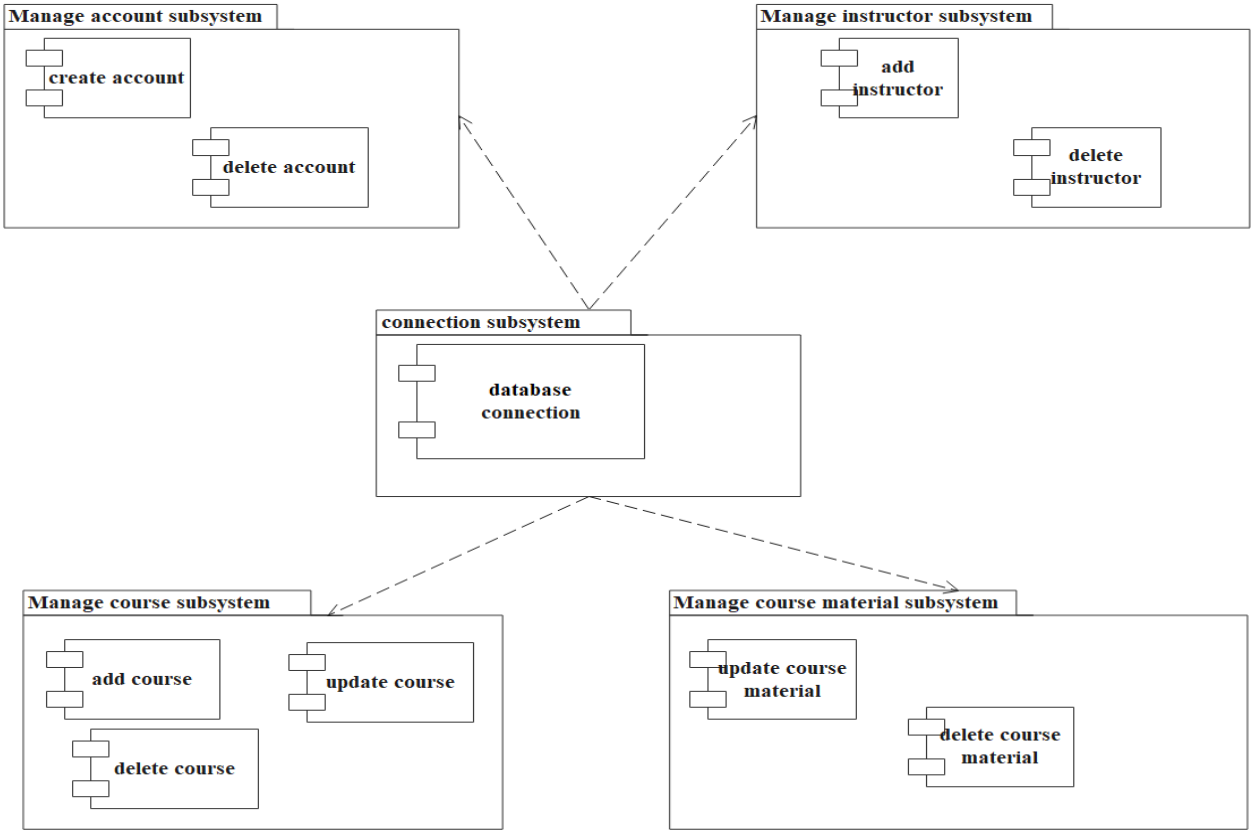


Figure 29: Sub system decomposition

5.2.2. Hardware /Software Mapping

The one and the main diagram, which shows this hardware and software mapping, is deployment diagram. Deployment diagram depicts a static view of the run-time configuration of processing nodes and the components that run on those nodes. In other words, deployment diagrams show the hardware for your system, the software installed on that hardware, and the middleware used to connect the disparate machines to one another.

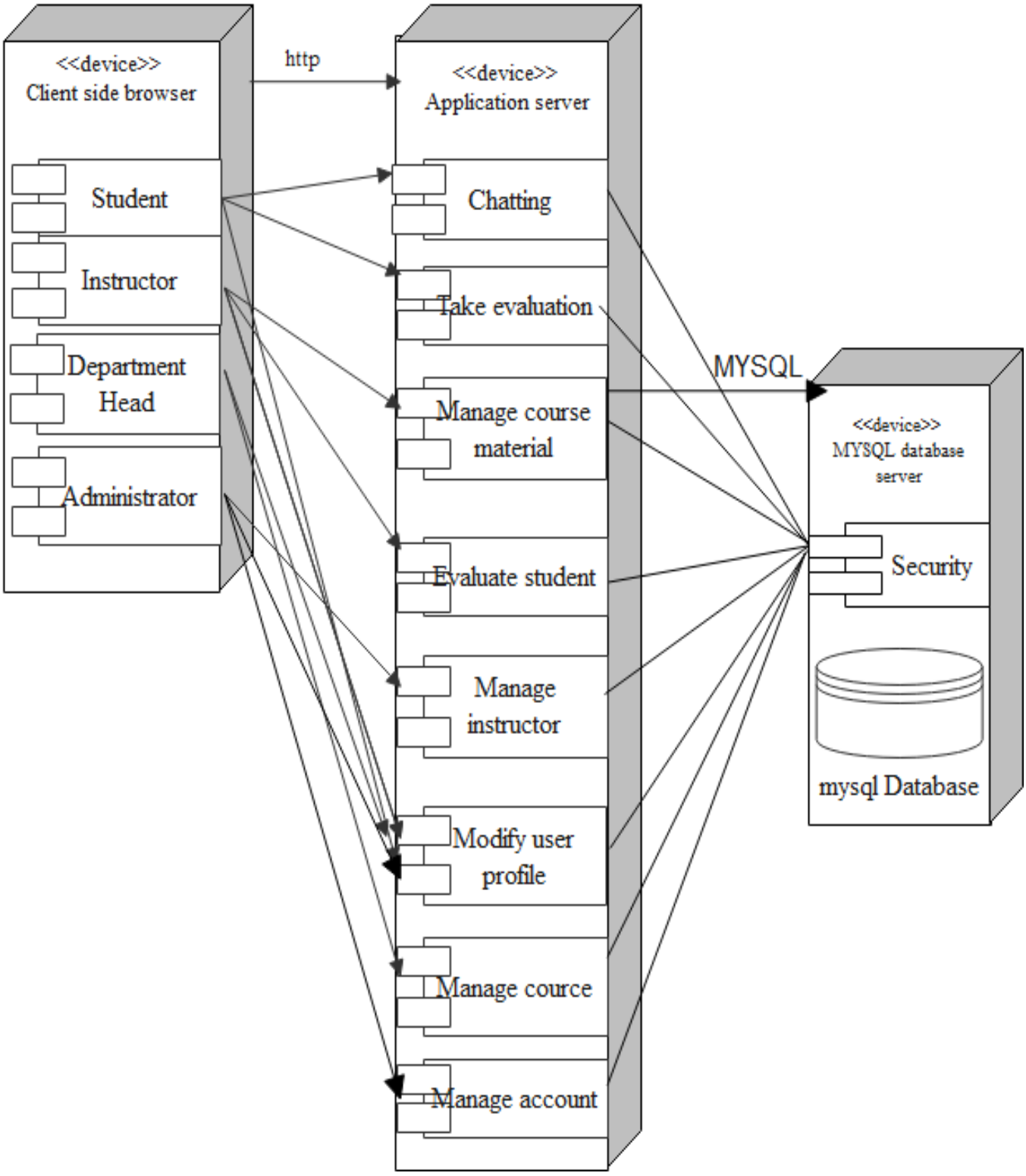


Figure 30: Hardware /Software Mapping

5.2.3. Detailed Class Diagram

Detailed Class Diagram Detail class diagram is class diagram that include attributes, methods, attribute data types, visibility of attributes and methods, inheritance, association, aggregation, composition, dependency, and municipality (cardinality and optimality. Below we have depicted our project class diagram using table and figure.

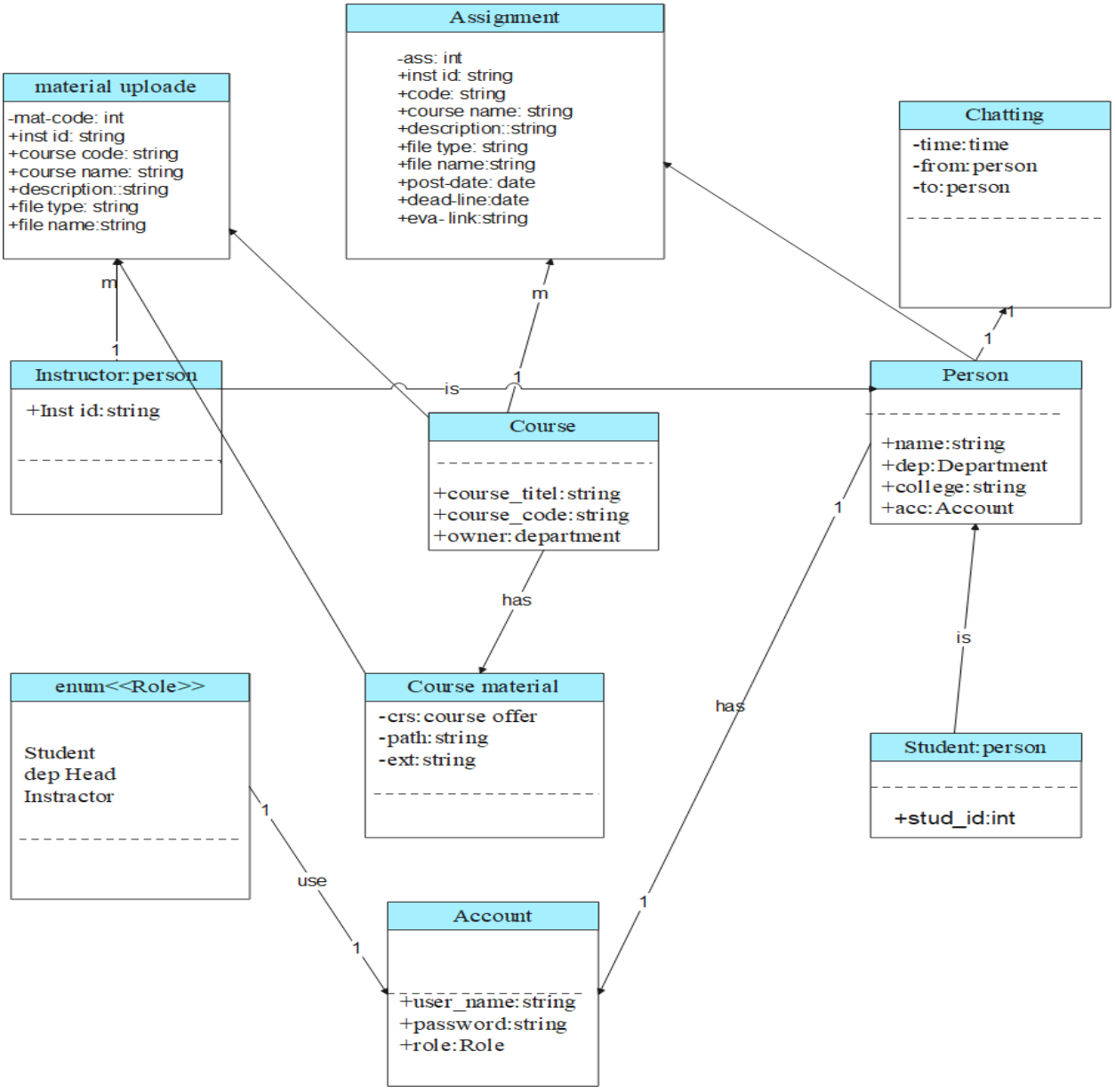


Figure 31: Detailed Class Diagram

5.2.4. Persistent Data Management

Persistent data management deals with how the persistent data such as file, database, etc. are stored and managed and to outlive a single execution of the system. This Persistent data management describes the persistent data stored by the system and the data management infrastructure required for it. The proposed system will use MySQL database for storing data. This will allow the database to be easily integrated and accessible by the rest of the system.

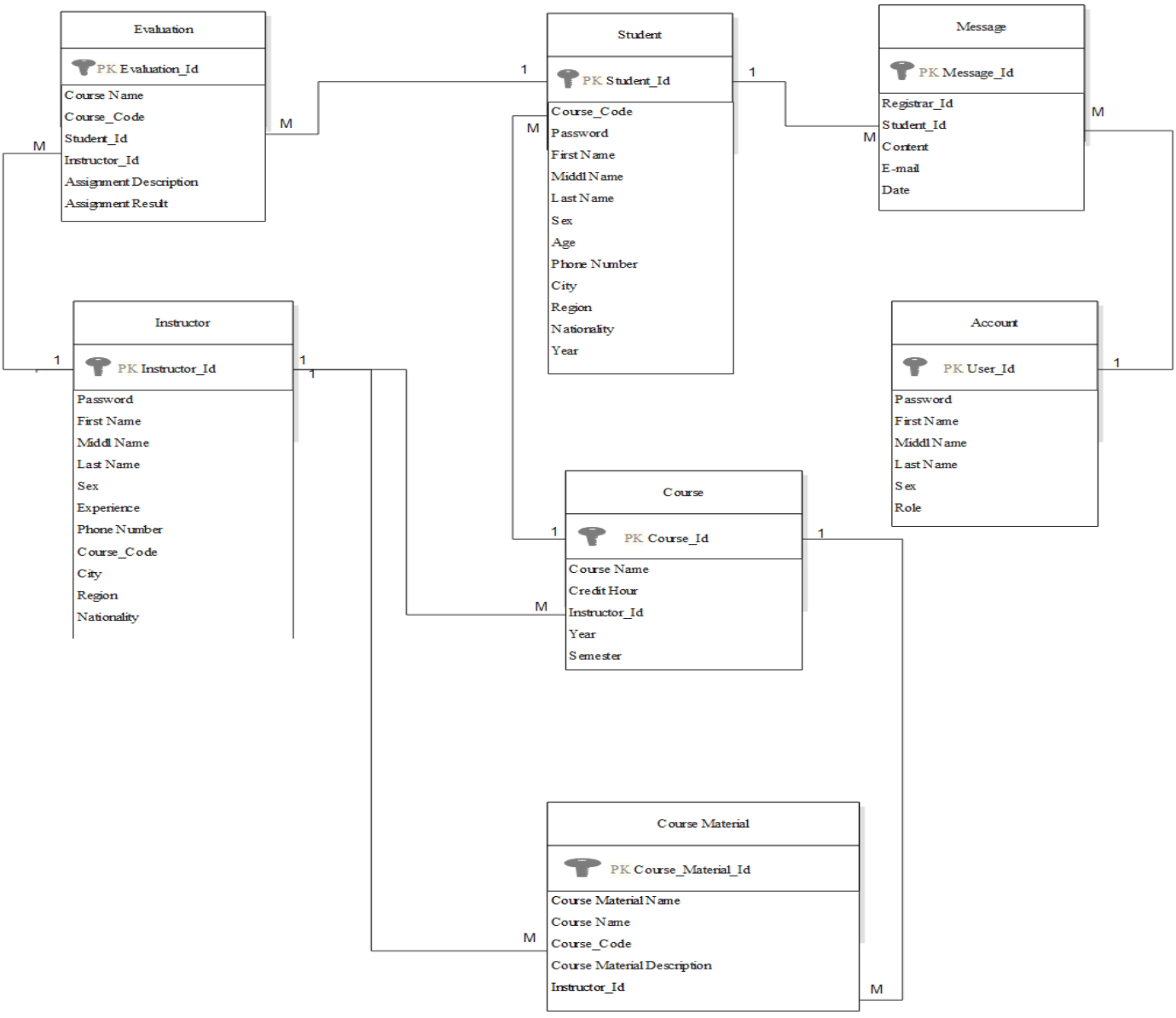


Figure 32: Persistent Modeling/ Database Design

5.2.5. Access Control and Security

In our project, different actors have access to different information and data. Access control and security specifies what the user can access or what cannot perform by some users. This access control is verified by ID and password. System administrator represents an authenticated user. In our system only authenticated user can access our system and only permitted user can use our system. The proposed system follows multi user system. In multi user system, different actors have access to different functionality and data. Then it must be having:

- Confidentiality: Only authorized person can see the information. Private data is kept private and personal privacy is respected.
- Availability: The system is available at all times to authorized users.

Table 15: Access Control and Security

Actor	**Operation**			
	Manage account	**View**	**Manage Course**	**Manage Instructor**
Admin	Create Update Delete	------	-------	Manage instructor
Department head	Update	------	Update Assign	------
Instructor	Update	Notification Course Student	Offer	-------
Student	Update	Course Notification	Take	-------

5.3. Packages

A package is an organized and functionality-based set of related interfaces and classes. Packages organize classes that belong to the same category or provide similar functionality. We categorized our system into course management package, database management package, instructor management package, account management package, security management package and course material management package.

- Database management package: The database subsystem will be implemented by relational database management system which is used to store the persistent data.
- Instructor management package: It is a subsystem that helps students of our system to offer course material.
- Course management package: It is a sub system that helps to department head to assign course to teachers
- Account management package: It is a subsystem that helps system administrator to manage account.
- Security management package: It is a subsystem that helps to secure the proposed system. The following diagram is shows the package diagram of our system.

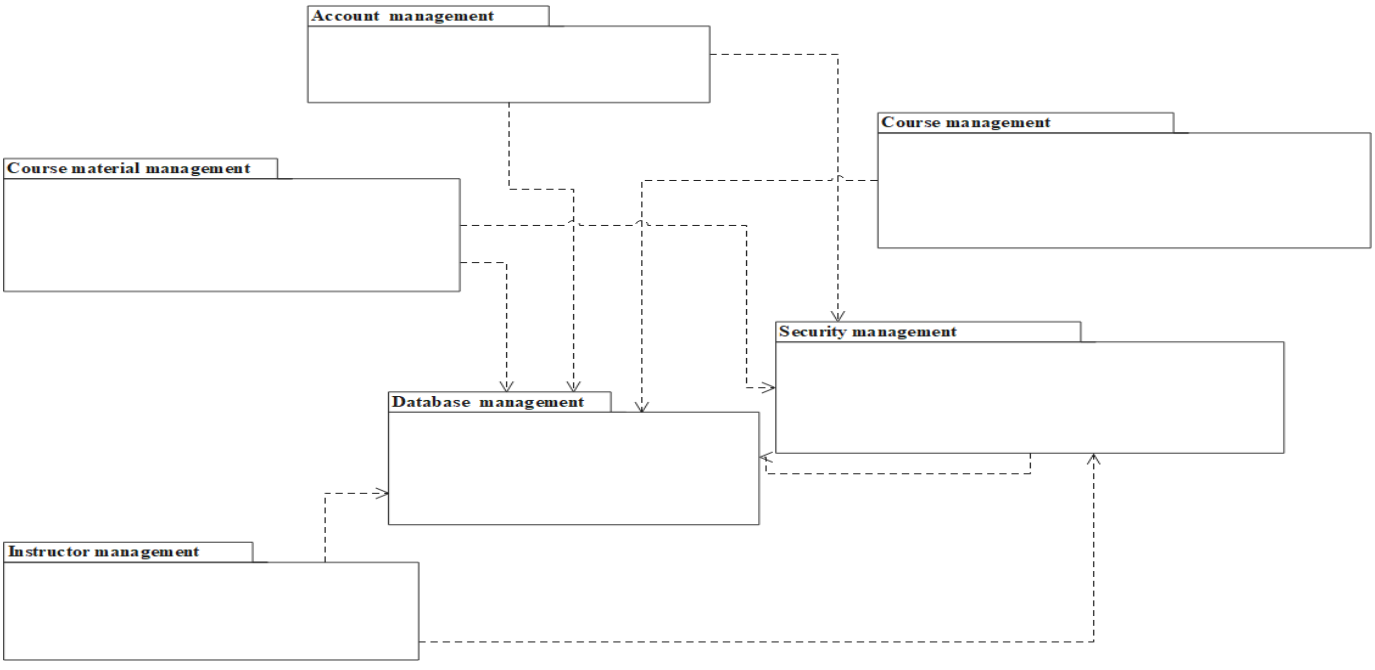


Figure 33: Packages

5.4. Algorithm Design

Algorithms are designed to show the flow of programs in the system. They are semantic driven rather than syntax driven. That means, the rule of syntax is not respected as other programming language but it has a complete meaning as that of syntax-based programming language.in addition, Algorithms show the flow and steps of logic in each function. This design part is important in the coding part of implementation. Some of the algorithms are listed below. Example:

Login

- Home page displayed.
- User click on sign in link
- System display sign in page.
- User enters user id and password.
- **If** user id and password is correct, **then**
- **If** user is user, **then** system display user page.
- **Else if** user is student**, then** system display student page.
- **Else if** user is instructor**, then** system display instructor page.
- **Else if** user is admin, **then** system display admin page.
- **Else if** user is department head, **then** system display department head page.
- **Else** user id and password is not correct, then system display error message and redisplay login page.

Create account

- Home page displayed.
- User click on sign up
- System display sign up page.
- Users fulfill its information.
- **If** entered information is correct, **then**
- **If** user name is verified, t**hen** user information registered correctly.
- **Else** user information is not correct, **then** system display your input is incorrect message**.**

5.5. User Interface Design

User interface design (UI) is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design) so in our project we have designed user interfaces that increase the user experience.

Home User Interface

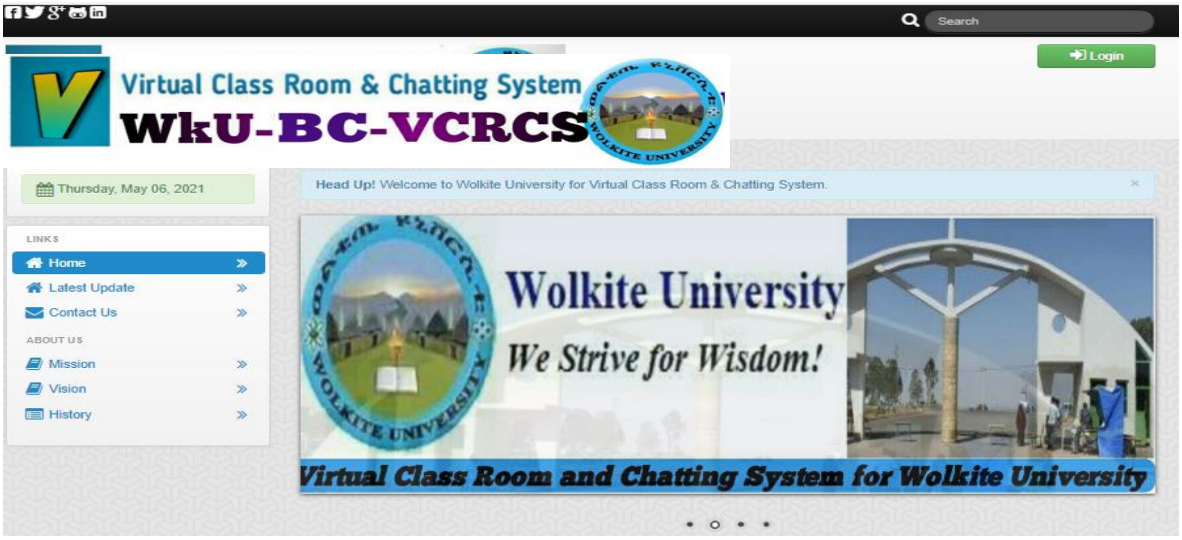


Figure 34: home page user interface diagram

Login User Interface

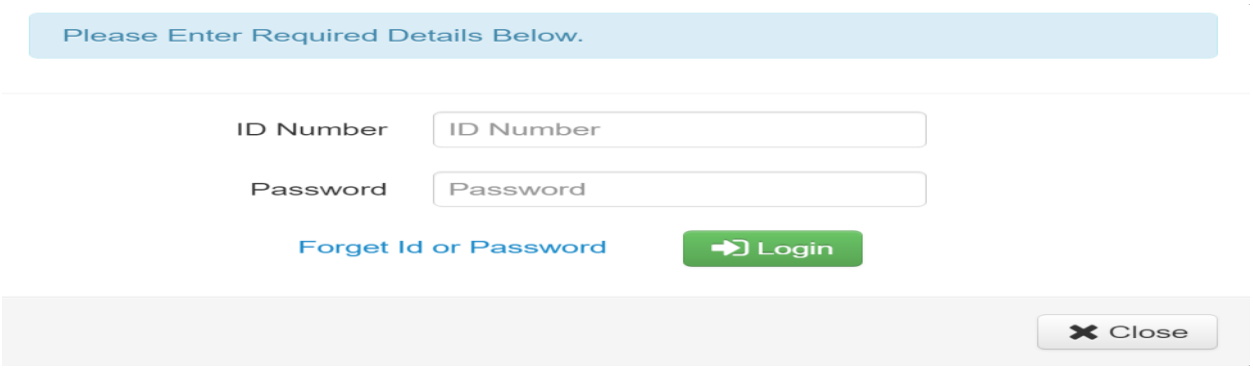


Figure 35: login page user interface diagram

CHAPTER 6

6. IMPLEMENTATION AND TESTING

In this chapter we discuss about implementation of database, detailed class diagram, application server, application security.

6.1 Implementation of the Database

We use MYSQL database to manage our system which assists to insert, update, delete and view data which is stored in the database.

We use MySQL because it can be deployed and run on any operating system. MySQL is easy, fast and can be used for any type of database whether it is relational or simple, large or small.

-- Table structure for table `account`

```
CREATE TABLE IF NOT EXISTS `account` (
  `user_id` varchar(30) NOT NULL,
  `f_name` varchar(20) NOT NULL,
  `m_name` varchar(20) NOT NULL,
  `l_name` varchar(20) NOT NULL,
  `user_type` varchar(30) NOT NULL,
  `e_mail` varchar(40) NOT NULL,
  `phone_no` int(10) NOT NULL,
  `pro_image` varchar(200) NOT NULL,
  `about_me` varchar(10000) NOT NULL,
  `password` varchar(50) NOT NULL,
  `status` varchar(10) NOT NULL,
```

```
  PRIMARY KEY (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-- Dumping data for table `account`

```
INSERT INTO `account` (`user_id`, `f_name`, `m_name`, `l_name`, `user_type`, `e_mail`, `phone_no`, `pro_image`, `about_me`, `password`, `status`) VALUES()
```

-- Table structure for table `course`

```
CREATE TABLE IF NOT EXISTS `course` (
  `course_code` varchar(10) NOT NULL,
  `course_name` varchar(10000) NOT NULL,
  `credit_hour` int(11) NOT NULL,
  `year` varchar(20) NOT NULL,
  `semester` varchar(20) NOT NULL,
  `description` varchar(20000) NOT NULL,
  `inst_id` varchar(30) NOT NULL,
  PRIMARY KEY (`course_code`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-- Dumping data for table `course`

```
INSERT INTO `course` (`course_code`, `course_name`, `credit_hour`, `year`, `semester`, `description`, `inst_id`) VALUES()
```

-- Table structure for table `materialupload`

```
CREATE TABLE IF NOT EXISTS `materialupload` (
  `mat_code` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `inst_id` varchar(30) NOT NULL,
  `course_code` varchar(30) NOT NULL,
  `course_name` varchar(100) NOT NULL,
  `description` varchar(5000) NOT NULL,
  `file_type` varchar(100) NOT NULL,
  `file` mediumtext NOT NULL,
  `date` date NOT NULL,
  PRIMARY KEY (`mat_code`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

-- Table structure for table `assignment`

```
CREATE TABLE IF NOT EXISTS `assignment` (
  `ass_id` varchar(20) NOT NULL,
  `inst_id` varchar(20) NOT NULL,
  `code` varchar(20) NOT NULL,
  `description` varchar(300) NOT NULL,
  `file` mediumtext NOT NULL,
  `post_date` varchar(100) NOT NULL,
  `dead_line` varchar(100) NOT NULL,
  `eval_link` varchar(40) NOT NULL,
  `status` varchar(20) NOT NULL,
  PRIMARY KEY (`ass_id`)
```

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table `assignment`

INSERT INTO `assignment` (`ass_id`, `inst_id`, `code`, `description`, `file`, `post_date`, `dead_line`, `eval_link`, `status`) VALUES()

- Table structure for table `submit`

CREATE TABLE IF NOT EXISTS `submit` (

`assignment_id` varchar(100) NOT NULL,

`course_code` varchar(20) NOT NULL,

`inst_id` varchar(20) NOT NULL,

`stud_id` varchar(20) NOT NULL,

`file` varchar(10000) NOT NULL,

`date` date NOT NULL,

`status` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Dumping data for table `submit`

INSERT INTO `submit` (`assignment_id`, `course_code`, `inst_id`, `stud_id`, `file`, `date`, `status`)

-- Table structure for table `total_credit`

CREATE TABLE IF NOT EXISTS `total_credit` (

`year` varchar(50) NOT NULL,

`semester` varchar(50) NOT NULL,

`semi_cr` varchar(50) NOT NULL,

```
  `total_cr` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table `total_credit`

INSERT INTO `total_credit` (`year`, `semester`, `semi_cr`, `total_cr`)
```

6.3 Configuration of the Application Server

We use WAMPS application server because WAMPS is simple, lightweight Apache distribution it is extremely easy to create a local web server for testing and deployment purposes. Everything you needed is to set up a web server – server application (Apache), database (Maria DB), and scripting language (PHP).

6.4 Configuration of Application Security

Our system validates all input by returning error message and suggesting to try again message when invalid message occur. The system has three roles and the roles are defined clearly and the user login to their specified page

6.5 Implementation of User Interface

We implement the user interface by considering the users of our system and easy for use
Interface of our system is developed using English language. Because the student can understand simple way by using bootstrap a CSS.

Login sample code

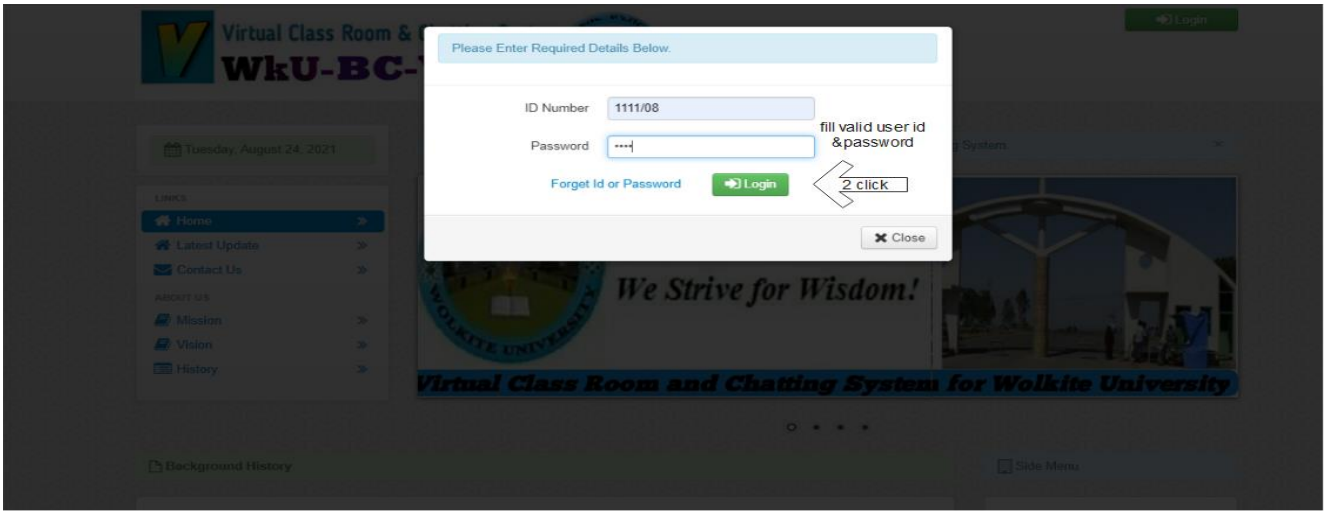
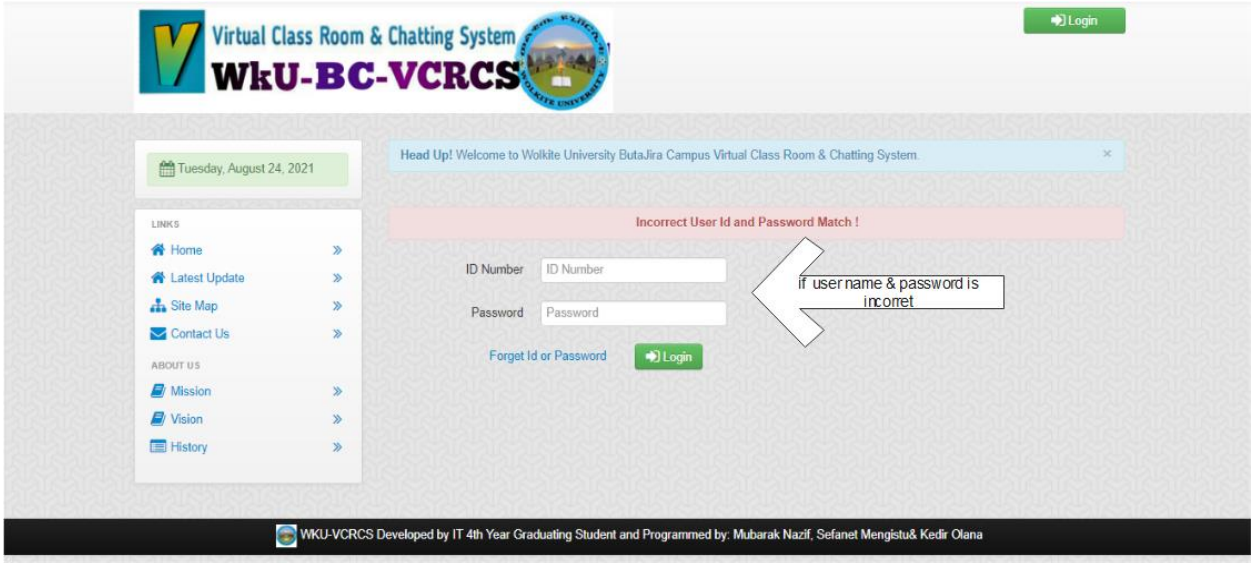


Figure 36 Login sample code




```
<?php
include('dbcon.php');
if (isset($_POST['login'])){
session_start();
$user_id = trim($_POST['user_id']);
$password = $_POST['password'];
$e_pass=$password;
$query = "SELECT * FROM account WHERE user_id='$user_id' AND password='$e_pass' ";
```

```
$result = mysql_query($query)or die(mysql_error());
$num_row = mysql_num_rows($result);
$row=mysql_fetch_array($result);
$row_id=$row['user_id'];
$status=$row['status'];
$abena = mysql_query("SELECT * FROM Assignment where status='New'");
while($keb=mysql_fetch_array($abena)){
$date_co=trim($keb['dead_line']);
$c_co=trim($keb['code']);
$dat=date('Y-m-d');
echo $date_co;
echo $dat;
if($dat>$date_co){
mysql_query("Update Assignment set status='Pass' where dead_line='$date_co'");
}}
$aben = mysql_query("SELECT * FROM Schedule where status='New'");
while($kebe=mysql_fetch_array($aben)){
$date_come=$kebe['date'];
$date=date('Y-m-d');
if($date>$date_come){
mysql_query("Update Schedule set status='Pass' where date='$date_come'");
}}
$ab = mysql_query("SELECT * FROM Withdrawal where status='Accept'");
while($kebo=mysql_fetch_array($ab)){
$year=$kebo['year'];
$da=date('Y');
$stud_id=$kebo['stud_id'];
if($da>$year){
mysql_query("Update Account set status='Active' where user_id='$stud_id'");
mysql_query("Update Student set status='Active' where stud_id='$stud_id'");
}}
```

```
$user_type =$row["user_type"];
$row=mysql_fetch_array($result);
if( $num_row > 0 ) {
if($user_type=="Registrar"){
header('location:Registrar/index.php');
$_SESSION['id']=$row_id;
}
else if($user_type=="Dep_Head"){
header('location:Dep_Head/index.php');
$_SESSION['id']=$row_id;
}
else if($user_type=="Instructor"){
header('location:Instructor/index.php');
$_SESSION['id']=$row_id;
}
else if($user_type=="Student"){
if($status=="Active"){
header('location:Student/index.php');
$_SESSION['id']=$row_id;
}
else{
echo"<script>alert('Your account has been temporally Deactivated')</script>";
echo "<script>windows:location='index.php'</script>";
}}
else{
header('location:access_denied.php');
}}
else{
header('location:access_denied.php');
}
}?>
```

Upload material

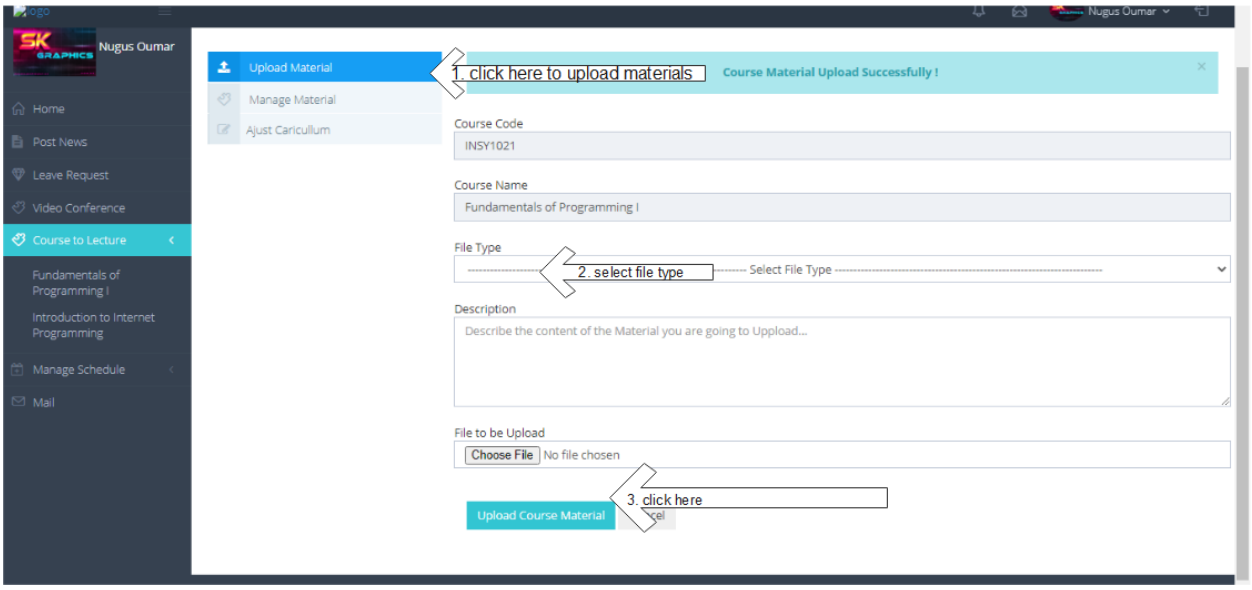


Figure 37: Upload material

Download Uploaded materials

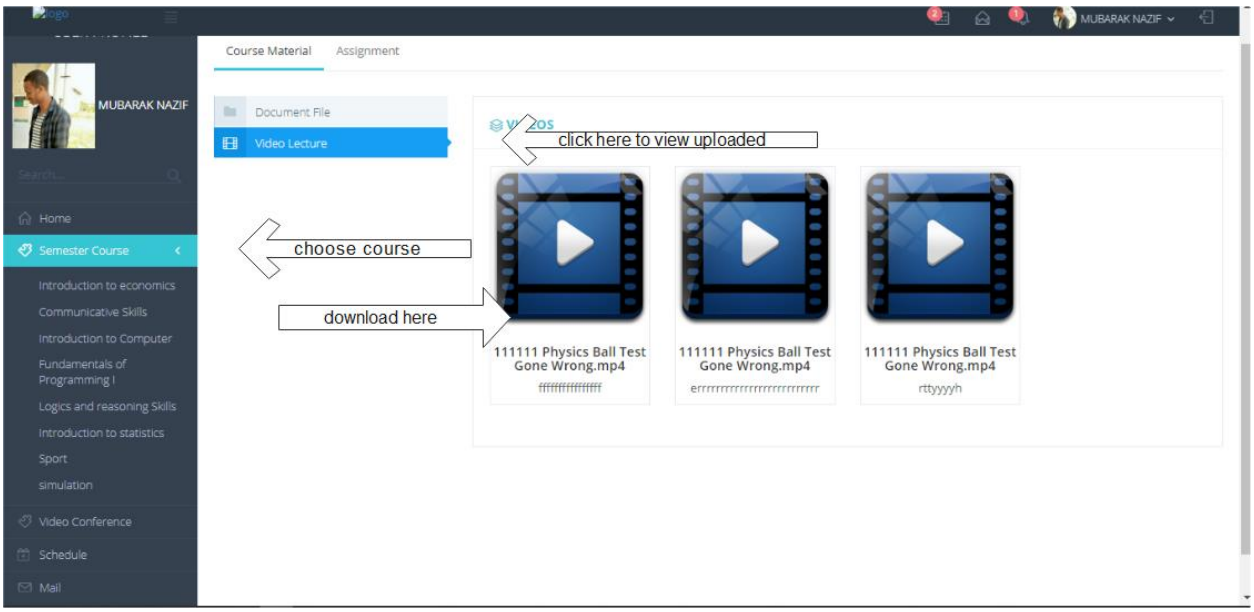


Figure 38: Download uploaded materials

Download Uploaded Assignments

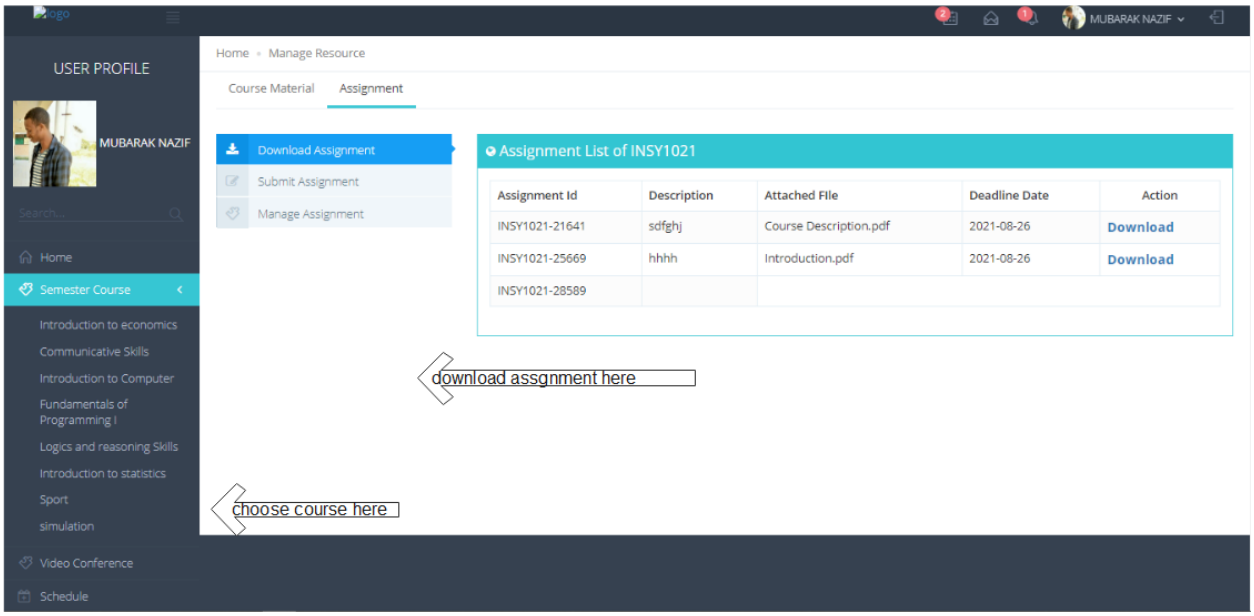


Figure 39: Download Uploaded Assignments

6.6 Testing

6.6.1 Testing Tools and Environment

Test Environment consists of elements that support test execution with software and hardware. Test environment configuration must mimic the production environment in order to uncover any environment.

A typical Environmental Configuration for a web-based application is given below: For web server we use Apache server, for Database MySQL, we use widow operating system, and we use chromes browser.

6.6.2 Unit Testing

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use

Unit testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the program's correctness by using JavaScript.

In this level of testing process, virtual classroom and chatting system for wku test the different sub procedures, functions and tested. Sample Tests Check whether the return type of the functions is correct.

1. Check how the sub procedures or functions are call correctly.
2. Check if the correct output is produced for different inputs.
3. Check the input data that we write on the GUI must be submitted to the data base.
4. Check the GUI can access the privileged data from the data base

6.6.3 System Testing

In this level of testing process, we have examined how the whole subsystems of virtual class room and chatting system work together to achieve the desired goal (user's requirements of the system). Generally, under this testing is mainly concerned with areas such as performance, security and validation of the System. But we will more focus only on function validation and performance.

Sample Tests

1. Evaluate the functionality of subsystem after combination of individual sub system weather it works correctly or not.
2. Check the coherence and coupling of each subsystem.
3. Check the overall functionality of the System that achieves the user's requirement.
4. Measure the system boundary which is beyond the goal or not.
5. Measure the weakness and the strength of the system using different metrics.
6. Check the interaction of each subsystem that performs the specified business process.
7. Verify the system completeness-based the requirement of end user.

6.6.4 Integration Testing

In this level of testing we have examined how the different procedures work together to achieve the goal of the sub system. The type of integration testing that we have followed is bottom up. Since our system is web-based system each and every access is depending on hypertext transfer protocol (HTTP). So, we integrate each component from single functionality (individual interface) to the main function incrementally step by step through link tag by using test driver. Sample Tests are check the interaction between individual functionality which performs the specific tasks, evaluate the functionality of subsystem after combination all individual functionality.

6.6.5 Acceptance Testing

According to system requirements and other resources (documentation, source code, user manual) test cases are generated to determine (validation and verification) whether the system satisfies users need and expectation to maintain the reliability of our system and also meet the user's requirements.

CHAPTER 7

7. CONCLUSION AND RECOMMENDATION

7.1 Conclusion

This project which has two phases; the first phase concerned with the analysis phase of the life cycle, the design phase and the next phase is about implementation. As the end of the first phase, we need to review that we have covered in accordance with what we have planned at the beginning. We began our work by identifying the significance of automated system for the store and the overall techniques to be used in the development process. This involved defining the system development methodology, identifying process. This involved defining the system development methodology, identifying resource and cost requirements, and setting the deliverable and scheduled for the project.

The analysis helps the team to well understand the major functional areas and processes of the system. Through this method we evaluate the existing system weakness and strength.

7.2 Recommendation

The system that we have tried to automate is not the whole system of Virtual Class Room & Chatting System of Wolkite University Students. Because of time limitation, we can't develop all parts of the system, but we tried to automate some sub systems and functionalities.

Therefore, others are interested individuals to develop Virtual Class Room & Chatting System can get some initial idea about the system and no need of more data gathering process the only need will be improving the system.

- Serious consideration should be given for the introduction of the new system.
- The staffing division should have adequate computer facilities for the introduction of the new system.
- Since the system is more users and students every user can access it without any trouble So we recommended using this system to the university.

- Maintenance is needed for the new system, so concerned body should take in consideration for maintaining the system.
- Budget should be allocated to take specific measure.
- To have standardize database.

REFERENCES

[1] Albrecht, Enriching student experience through hybrid learning, Washington, DC: Educause Center for Applied Research., 2006.

[2] Allen, I. E., & Seaman, J, Entering the mainstream: The quality and extent of online education, United States: Wellesley, MA: Sloan Consortium., 2003 and 2004.

[3] Akridge, J., DeMay, L., Braunlich, L., Collura, M., & Sheahan, Retaining adult learners in a high-stress distance education learning environment, Purdue University: The executive MBA , (2002).

[4] D. S. Gedera, "student' experiences of learning in a virtual classroom, International Journal of Education and Development using Information and communication Technology," *(IJEDICT),* Vols. Vol,10, no. Issues 4, pp. pp.93-101, 2014.