

**Bahir Dar University**

**Bahir Dar Institute of Technology**

**Faculty of Computing**

**Final Project on Online Distance Education Management System for Bahir Dar University**

**Submitted to Faculty of Computing in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Information System**

**Group members**

**Name ID Number**

**1. Kassahun Abebe 0701012**

**2. Matusala Asrat 0701021**

**3. Zemenay Alene 0701150**

**Advisor: Muluken B.**

**2018**

Bahir Dar, Ethiopia

# **Declaration**

The Project is our own and has not been presented for a degree in any other university and all the sources of material used for the project have been duly acknowledged.

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

#### Name Signature

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

#### Name Signature

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

#### Name Signature

**Faculty:** Computing

**Program**: Information System

**Project Title**: **Online Distance Education Management System for Bahir Dar University**

This is to certify that I have read this project and that in my supervision and the students’ performance, it is fully adequate, in scope and quality, as a project for the degree of Bachelor of Science.

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

#### Name of Advisor Signature

|  |  |  |
| --- | --- | --- |
| **Examining committee members** | **signature** | **Date** |
| 1. Examiner 1 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| 2. Examiner 2 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |

It is approved that this project has been written in compliance with the formatting rules laid down by the faculty.

# **Roles and Responsibilities of the Group Members**

|  |  |  |  |
| --- | --- | --- | --- |
| List of Tasks | List of members | | |
| Kassahun Abebe | Matusala Asrat | Zemenay Alene |
| Requirement gathering |  |  |  |
| Use case drawing and documentation |  |  |  |
| Activity diagram and sequence diagram drawing |  |  |  |
| Analysis class model |  |  |  |
| CRC drawing |  |  |  |
| Component diagram drawing |  |  |  |
| Deployment diagram drawing |  |  |  |
| Drawing design class diagram |  |  |  |
| user interface design |  |  |  |
| Persistence model |  |  |  |

# **Acknowledgment**

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 muluken Birara, 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. Then we would like to thanks instructor of faculty of computing who helps as by giving information about the project. Finally we would like to thank our classmates and friends.

# **List of Acronyms**

|  |  |
| --- | --- |
| Abbreviation | Description of abbreviation |
| **BDU** | Bahir Dar University |
| **CDE** | Continues and Distance Education |
| **Btn** | Button |
| **AD** | Activity Diagram |
| **DB** | Data Base |
| **FREQ** | Functional Requirement |
| **UC** | Use Case |
| **BR** | Business Rule |
| **UML** | Unified Modeling Language |
| **GB** | GigaByte |
| **DBMS** | Data Base Management System |
| **GHZ** | GigaHertz |
| **IE** | Internet Explorer |
| **HTTP** | HyperText Transfer Protocol |
| **CRC** | Class Responsibility Collaboration |
| **HTML** | HyperText Markup Language |
| **PHP** | Hypertext Pre-processor |
| **CSS** | Cascading Style Sheet |
| **PC** | Personal Computer |
| **CD** | Compact Disc |
| **ID** | Identification |
| **MGT** | Management |
| **AL** | algorithm |
| **NFR** | Non Functional Requirement |
| **MS** | Microsoft |
| **WAMP** | Window Apache MySQL PHP |

# **List of Figures**

[Figure 1 Use case diagram 17](#_Toc505561669)

[Figure 2 User Interface prototype 31](#_Toc505561670)

[Figure 3 Activity Diagram login 33](#_Toc505561671)

[Figure 4 Activity Diagram logout 33](#_Toc505561672)

[Figure 5 Activity Diagram add student 34](#_Toc505561673)

[Figure 6 Activity Diagram download module 35](#_Toc505561674)

[Figure 7 Activity Diagram view notice 36](#_Toc505561675)

[Figure 8 Activity Diagram view course result 37](#_Toc505561676)

[Figure 9 Activity Diagram view course 38](#_Toc505561677)

[Figure 10 Activity Diagram register Instructor 39](#_Toc505561678)

[Figure 11 Activity Diagram Assign Instructor 40](#_Toc505561679)

[Figure 12 Activity Diagram view notice 41](#_Toc505561680)

[Figure 13 Activity Diagram add course 42](#_Toc505561681)

[Figure 14 Activity Diagram create account 43](#_Toc505561682)

[Figure 15 Activity Diagram deactivate user account 44](#_Toc505561683)

[Figure 16 Activity Diagram upload learning material 45](#_Toc505561684)

[Figure 17 Activity diagram add department 46](#_Toc505561685)

[Figure 18 Sequence diagram login 48](#_Toc505561686)

[Figure 19 Sequence diagram logout 49](#_Toc505561687)

[Figure 20 Sequence diagram download 49](#_Toc505561688)

[Figure 21 Sequence diagram create account 50](#_Toc505561689)

[Figure 22 Sequence diagram upload learning material 51](#_Toc505561690)

[Figure 23 Sequence diagram add dept. 52](#_Toc505561691)

[Figure 24 Sequence diagram add course 53](#_Toc505561692)

[Figure 25 Sequence diagram register Instructor 54](#_Toc505561693)

[Figure 26 Sequence diagram Assign Instructor 55](#_Toc505561694)

[Figure 27 Sequence diagram view notice 56](#_Toc505561695)

[Figure 28 Sequence diagram view result 56](#_Toc505561696)

[Figure 29 Analysis Class Model 58](#_Toc505561697)

[Figure 30 Component modeling 68](#_Toc505561698)

[Figure 31 Deployment Modeling 69](#_Toc505561699)

[Figure 32 Design class model 71](#_Toc505561700)

[Figure 33 Persistent model 73](#_Toc505561701)

[Figure 34 User Interface Design of home page 74](#_Toc505561702)

[Figure 35 User Interface Design login page 74](#_Toc505561703)

[Figure 36 User Interface Design academic dean home page 75](#_Toc505561704)

[Figure 37 User Interface Design system admin home page 75](#_Toc505561705)

# **List of Tables**

[Table 1 FREQ upload course materials 12](#_Toc505561585)

[Table 2 FREQ login 13](#_Toc505561586)

[Table 3 FREQ manage user account 13](#_Toc505561587)

[Table 4 FREQ course material download 13](#_Toc505561588)

[Table 5 FREQ view course result 13](#_Toc505561589)

[Table 6 FREQ add student 14](#_Toc505561590)

[Table 7 FREQ Add course result 14](#_Toc505561591)

[Table 8 FREQ Register Instructor 14](#_Toc505561592)

[Table 9 FREQ Assign Instructor 14](#_Toc505561593)

[Table 10 FREQ Add notice 15](#_Toc505561594)

[Table 11 FREQ Add course 15](#_Toc505561595)

[Table 12 FREQ logout 15](#_Toc505561596)

[Table 13 Use case documentation of login 18](#_Toc505561597)

[Table 14 Use case documentation of logout 19](#_Toc505561598)

[Table 15 Use case documentation of download module 19](#_Toc505561599)

[Table 16 Use case documentation of view notice 20](#_Toc505561600)

[Table 17 Use case documentation of view Result 21](#_Toc505561601)

[Table 18 Use case documentation of view course 21](#_Toc505561602)

[Table 19 Use case documentation of Register Instructor 22](#_Toc505561603)

[Table 20 Use case documentation of Assign Instructor 22](#_Toc505561604)

[Table 21 Use case documentation of Add notice 23](#_Toc505561605)

[Table 22 Use case documentation of Add course 23](#_Toc505561606)

[Table 23 Use case documentation of Remove notice 24](#_Toc505561607)

[Table 24 Use case documentation of Add student 24](#_Toc505561608)

[Table 25 Use case documentation of create account 25](#_Toc505561609)

[Table 26 Use case documentation of deactivate user account 26](#_Toc505561610)

[Table 27 Use case documentation of activate user account 26](#_Toc505561611)

[Table 28 Use case documentation of Add student result 27](#_Toc505561612)

[Table 29 Use case documentation of upload learning material 28](#_Toc505561613)

[Table 30 Use case documentation of add department 28](#_Toc505561614)

[Table 31 BR of provide educational report to register 29](#_Toc505561615)

[Table 32 BR of determine the need of ID 29](#_Toc505561616)

[Table 33 BR of user account 29](#_Toc505561617)

[Table 34 BR of determine educational fee 30](#_Toc505561618)

[Table 35 BR of accessing module 30](#_Toc505561619)

[Table 36 BR of upload learning material 30](#_Toc505561620)

[Table 37 NFR Usability 63](#_Toc505561621)

[Table 38 NFR Availability 63](#_Toc505561622)

[Table 39 NFR Performance 63](#_Toc505561623)

[Table 40 NFR Security 64](#_Toc505561624)

[Table 41 NFR Documentation 64](#_Toc505561625)

Table of Contents

[Declaration i](#_Toc505561500)

[Roles and Responsibilities of the Group Members ii](#_Toc505561501)

[Acknowledgment iii](#_Toc505561502)

[List of Acronyms iv](#_Toc505561503)

[List of Figures v](#_Toc505561504)

[List of Tables vi](#_Toc505561505)

[Abstract ix](#_Toc505561506)

[Chapter One: Introduction 1](#_Toc505561507)

[1.1 Background of the Project 1](#_Toc505561508)

[1.2 Existing System Study 1](#_Toc505561509)

[1.3 Statement of the Problem 2](#_Toc505561510)

[1.4 Proposed System 2](#_Toc505561511)

[1.5 Objectives 3](#_Toc505561512)

[1.5.1 General Objective 3](#_Toc505561513)

[1.5.2 Specific Objectives 3](#_Toc505561514)

[1.6 Scopes and Limitations of the Project 3](#_Toc505561515)

[1.6.1 Scopes 3](#_Toc505561516)

[1.6.2 Limitations 4](#_Toc505561517)

[1.7 Methodology of the Project 4](#_Toc505561518)

[1.7.1 Data Gathering Method 4](#_Toc505561519)

[1.7.2 System Analysis and Design Methodologies 5](#_Toc505561520)

[1.7.3 Development Tools and Languages used 7](#_Toc505561521)

[1.7.3.1 Software Tools 7](#_Toc505561522)

[1.7.3.2 Hardware tools used 7](#_Toc505561523)

[1.8 Feasibility Study 7](#_Toc505561524)

[1.8.1 Economic feasibility 7](#_Toc505561525)

[1.8.2 Technical feasibility 8](#_Toc505561526)

[1.8.3 Operational feasibility 9](#_Toc505561527)

[1.9 Significance of the Project 9](#_Toc505561528)

[1.10 Organization of the project 10](#_Toc505561529)

[Chapter Two: System Features 11](#_Toc505561530)

[2.1 The Existing System 11](#_Toc505561531)

[2.2 Proposed System 11](#_Toc505561532)

[2.3 Requirement Analysis 12](#_Toc505561533)

[2.3.1 Functional Requirements 12](#_Toc505561534)

[2.3.2 System Use case 16](#_Toc505561535)

[2.3.3 Business Rule Documentation 29](#_Toc505561536)

[2.3.4 User Interface prototype 30](#_Toc505561537)

[2.3.5 Activity Diagram 32](#_Toc505561538)

[2.3.6 Sequence diagram 47](#_Toc505561539)

[2.3.7 Analysis Class Model 57](#_Toc505561540)

[2.3.8 Logic model/pseudo code 59](#_Toc505561541)

[2.4 Nonfunctional Requirement 63](#_Toc505561542)

[2.5 System Requirement 65](#_Toc505561543)

[2.5.1 Hardware requirements 65](#_Toc505561544)

[2.5.2 Software requirements 65](#_Toc505561545)

[2.6 Key abstraction with CRC analysis 65](#_Toc505561546)

[Chapter 3: System Design 67](#_Toc505561547)

[3.1 Architectural Design 67](#_Toc505561548)

[3.1.1 Component modeling 67](#_Toc505561549)

[3.1.2 Deployment Modeling 69](#_Toc505561550)

[3.2 Detail Design 70](#_Toc505561551)

[3.2.1 Design class model 70](#_Toc505561552)

[3.2.2 Persistent model 72](#_Toc505561553)

[3.3 User Interface Design 74](#_Toc505561554)

[3.4 Access control and security 76](#_Toc505561555)

[Chapter Four 78](#_Toc505561556)

[Implementation and Testing 78](#_Toc505561557)

[4.1 Implementation 78](#_Toc505561558)

[4.2 Testing 78](#_Toc505561559)

[Chapter Five 80](#_Toc505561560)

[Conclusions and Recommendations 80](#_Toc505561561)

[References 81](#_Toc505561562)

# **Abstract**

Bahir Dar University is one of the higher institutions involved in continuing and distance education. This project is done in Bahir Dar University continues and distance education. This document contains the brief description of background information of the universities and project, method of data collection and analysis, description of the existing system and modeling and design of the proposed system.

**This education system currently has many problems, even if it is semi-automated system, because of the teaching and learning system works manually. Due to this** we are going to develop online system to support Teaching and learning processes which means students can follow their education online through internet and get learning materials in the system.

# **Chapter One: Introduction**

## Background of the Project

Bahir Dar University is one of the higher institutions involved in continuing and distance education. Continuing and distance education mainly organizes activities for the society in the fields that have gained importance on the improvement of individuals with a view of career building, improving professional knowledge and skills, achieving new skills and supplying individual development needs. For this purpose centers have been established within different universities through the principle of making continuing and distance education accessible to everyone as a starting point for a lifelong learning process.

Each centers which aim at providing continuing education, implement more flexible programs compared to those executed in regular programs at higher education institutions. Owing to this fact, continuing and distance education program of BDU needs to be made accessible further for those who cannot get regular education and be consolidated in such a way that quality education can take place. In light of this, its provision has to be properly investigated and recommendations forwarded so that timely measures can be taken that can assure quality education [1].

## Existing System Study

The existing system of distance education of the University is work manual, even if it is semi-automated system. The distance program usually registers students and provides modules and tutorials face to face. Modules are prepared by distance education center of the University. After that tutorial sessions are organized by the University in different stations ranging from Bahir Dar to different branches. This is done on average twice a year. The constraints that are found in the systems are, no enough instructors, shortage of time (while instructors prepare materials and distribute also the manual way of data handling and running up of the activities is human power consuming, resource wasting and boring.

## Statement of the Problem

**Bahir Dar University of continuing and distance Education system currently has many problems, even if it is semi-automated system, because of the teaching and learning system works manually.** As we have analyzed the existing system has the following problems;

* It takes high cost to duplicate learning material
* Distribution of learning material is take high cost
* Student registration, to take tutorial or give tutorial and viewing grade reports consume time and cost.
* In order to viewing their results students must be come in the center
* Overburden work for the University during providing the service to the students like distributing materials, assignment in a manual way.
* Students don’t have the opportunity to get up to date information on time i.e. since the system is manual based the information that were distributed for students may be modified, deleted, or changed. Example, the registration date, exam date and receiving materials were make confusion.

## Proposed System

This project is aimed to automate the currently existing system. The proposed system can give good service for those who use the system. Everything that would be executed manually is changed into automated system except exam and payment. The problems in the existing system will be get solution in the proposed system. The proposed system will perform the following activities.

* Make good and easy module duplication and distribution mechanisms.
* The opportunity of get information up to date.
* Learning materials reach on time to students place that they are present.
* Saving time for all staff member and students.
* It enables handle information easy.

## Objectives

### **General Objective**

**The main objective of this project is to design online distance education management system for Bahir Dar University.**

### **Specific Objectives**

To achieve the general objective, the proposed system accomplishes

* Gathering required information for proposed system
* Identify the current system function.
* Analyzing the gathered information.
* Specifying functional and non-functional requirements of the proposed System
* Design the proposed system.
* Design a user interface for the proposed system.
* Implementing the proposed system.
* Testing the proposed system

## Scopes and Limitations of the Project

### Scopes

The scope of our project is limited to Bahir Dar University of continuing and distance education.

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.

**Due to the applicability the following activities will not include to be automated in the system.**

* Online examination
* **Online payment**
* **Vacancy.**

### Limitations

Challenges which are faced on us during the proposed system accomplishment

* Information availability is very difficult or getting necessary information is difficult.
* We have not enough time to gather requirements because of we have normal class over loaded.

## Methodology of the Project

### 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:**

* **Interviewing: As a method for the collection of data about the activities of University of continuing and 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.**
* **Direct Observation:** We observe the current system and identify the problems regarding to the working process .so it help us as easy way to understand the system and to develop the project. By observing their system structure we design a project to solve the problems..
* **Document Analysis: we analyze some documents and forms found in the office to get information about the current system.**

### System Analysis and Design Methodologies

**System Analysis:** is the process of breakdown an entire system into module, analyzing each module separately, and determining the relationship between them. In system development process we use system modeling i.e. creating model of the system. System modeling is the process of creating a model of system by analyzing and organizing the system elements. This is the phase where deeply understanding of the existing system problems and finding alternative solution. Models are:

***Use case diagram***: Use cases define interactions between external actors and the system to attain particular goals. There are three basic elements that make up a use case:

* **Actors**: Actors are the type of users that interact with the system.
* **System**: Use cases capture functional requirements that specify the intended behavior of the system.
* **Goals**: Use cases are typically initiated by a user to fulfill goals describing the activities and variants involved in attaining the goal

***Class diagram***: is a type of static structure diagram that describes the structure of a system by showing the system's [classes](https://en.wikipedia.org/wiki/Class_(computer_science)), their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) modeling. It is used for general [conceptual modeling](https://en.wikipedia.org/wiki/Conceptual_model) of the systematic of the application, and for detailed modeling translating the models into [programming code](https://en.wikipedia.org/wiki/Programming_code). Class diagrams can also be used for [data modeling](https://en.wikipedia.org/wiki/Data_modeling). The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

***Activity diagram***: An activity diagram visually presents a series of actions or flow of control in a system similar to a [flowchart](https://www.smartdraw.com/flowchart/) or a [data flow diagram](https://www.smartdraw.com/data-flow-diagram/). Activity diagrams are often used in business process modeling. They can also describe the steps in a [use case diagram](https://www.smartdraw.com/use-case-diagram/). Activities modeled can be sequential and concurrent.

***Sequence diagram***: Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. Like

* Model high-level interaction between active objects in a system
* Model the interaction between object instances within a collaboration that realizes a use case
* Model the interaction between objects within a collaboration that realizes an operation
* Either model generic interactions (showing all possible paths through the interaction) or specific instances of interaction (showing just one path through the interaction)

**System Design*: -*** to design the system the project team has choose Object Oriented Modeling techniques and Unified modeling language tools. Design models are:

***Deployment diagram***: are used to visualize the topology of the physical components of a system, where the software components are deployed.

***Component diagram***: are describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

Generally the model that we are describing above can:

* Enables us to comprehensively model a system before we develop it.
* Modification of the object implementation is easy because objects are loosely coupled.
* Understanding of the structure is easy because object oriented modeling and tools used to represent real world entities.
* Direct manipulation of architectural components is possible because several object oriented programming languages exist.

### Development Tools and Languages used

### Software Tools

**We used Microsoft Visio while we are designing our new system. The development tools that we will use are:**

* Server side scripting: PHP, we have select PHP for server side scripting because PHP is compatible with wide Varity of web server
* **Client side scripting: by considering the following characteristics we use java script. It can be embedded in HTML page and it is very popular in validation process.**
* **We use Wamp server to configure a MySQL database and to use PHP applications for easy configuration.**
* **MS-word and Power point: For document preparation and presentation.**
* **Static webpage: we used HTML since it is highly flexible with CSS to use different layouts**
* **Browser like moziliafirfox, spark, IE.**

#### **Hardware tools used**

* PC (laptop): to perform our task like writing documentation and codes.
* Flash disk: to store file temporarily and used to file transfer computer to computer
* CD (compact disk 700M): to make back up files or store files.
* Mobile camera: to capture image that we need to the project.

## ****Feasibility Study****

Feasibility study is the process of determination of whether or not a project is worth doing. There are many types of feasibility study

### Economic feasibility

The system which we are going to develop will have economic benefit. Those economic benefits may be tangible or intangible

#### Tangible Benefits

This is the concrete benefit that can be expressed in terms of birr. So the system proposed to develop will decrease a lot of birr that was expensive to buy and distribute hard copy document material. Also reduce the loss of data that means if it reduces the loss of data it also reduces the cost of replacing the lost data. For example if the University has 2,000 students to handle these amount data there must be many people to manage the data and a lot of paper. But after the system developed the data can be managed with one person and in one computer in very short time.

* Easy file management.
* Reduce cost for manual data management (Reduced expenses)
* Easy update & retrieval on stored records

#### Intangible Benefit

Intangible benefits are those benefits that cannot be expressed in terms of birr. Intangible benefit that the system will give is the following:

* Properly ordered files of employees which have stability means which is not easily lost.
* Good service to the office
* Little job burden to employees of office
* Small response time And many services
* Knowledge gain by project developer.

### Technical feasibility

Technical feasibility determines whether the work for the project can be done with the existing equipment, software technology and available users. Technical feasibility is concerned with specifying equipment and software that will satisfy the user requirement.

This project is feasible on technical because the proposed system can run on any computer with internet access so it would be technically feasible.

### Operational feasibility

Operational feasibility is a measure of how well the solution will work in the organization. Operational feasibility is dependent up on the human resources available for the system. This online system for distance education in Bahir Dar University 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.

## 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.

## Organization of the project

In these project documents we are discussing system details in each chapter. Before we begin to write chapter one the document includes acknowledgment, list of acronym, abstracts, table of contents, list of figure, and list of tables.

The first chapter of this document includes the Introduction part of the project which includes sub topics like background, existing system study, proposed system, general objective, and specific objective, significant of the system, scope and methodologies and tools.

The second chapter of the document includes system feature like existing system, problem of existing system, proposed system, requirement analysis like functional requirement system use case, use case documentation, business rules, user interface prototype, activity diagram, sequence diagram, analysis class model logical model, nonfunctional requirement and system requirement like software requirement and hard ware requirement.

The third chapter of this document describes system design architecture diagram such as component diagram and deployment diagram, detail design like design class diagram and persistent model, user interface design and access control and security.

The fourth chapter of this document describes as implementation and testing. And the last chapter includes conclusions and recommendations of the project.

# **Chapter Two: System Features**

## The Existing System

**As we have observed in the data collecting phase, the main problems in Bahir Dar University continuing and distance education is wastage of resources during preparing, duplicating and process of distributing modules.**

**The existing system of Bahir Dar University continuing and distance education is semi-automated but the operation some task is still manually. Different activities are performed as follow:**

**Distributing modules, giving tutorials and showing course results are performed by instructor and registration process is performed by registrar office. The overall teaching learning process is controlled by academic dean.**

**Overburden of work on the university’s employee is also seen on the current system of Bahir Dar University continuing and distance education during duplicating and distributing modules, since each course has its own module to be distributed per student.** Students also view result of the course such as quizzes, tests by going back to their instructor. Such process leads students to waste their time, effort, money and make them unsatisfied.

**Generally those problems can be defined as:**

* **Wastage of University resources**
* **Time consuming because of some tasks operated manually**
* **Work overload on University employees**
* **Less of students’ satisfaction.**

## Proposed System

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

* **Reduce the extravagancy of the University resources.**
* **Reduce the time and task required to perform the operation within the University.**
* **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.**

## Requirement Analysis

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish.

Requirements analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people [5].

### Functional Requirements

The functional requirement refers to the functionality of the new system, i.e. extends to what are Services it will provide to the user. Statement of service is that system should provide how the system should react to particular inputs and how the system should behave in particular situations.

The following table shows functional requirement of the proposed system.

|  |  |
| --- | --- |
| Requirement id | FREQ-1 |
| Source | Instructor |
| Requirement | The system shall allow instructor upload learning material |
| Description | Instructor uploads course materials for the student in the system |
| Category | Learning material upload |
| Priority | High |

Table 1 FREQ upload course materials

|  |  |
| --- | --- |
| Requirement id | FREQ-2 |
| Source | All user |
| Requirement | The system shall allow users to login |
| Description | User login to the system to perform their task |
| Category | login |
| Priority | High |

Table 2 FREQ login

|  |  |
| --- | --- |
| Requirement id | FREQ-3 |
| Source | Admin |
| Requirement | The system shall allow admin to manage employee account |
| Description | Admin manage account for employee in the organization |
| Category | **Manage user account**(create,reset,deactivate,activate) |
| Priority | High |

Table 3 FREQ manage user account

|  |  |
| --- | --- |
| Requirement id | FREQ-4 |
| Source | student |
| Requirement | The system shall allow student download learning material |
| Description | student downloads learning materials from the system uploaded by instructor |
| Category | Learning material download |
| Priority | High |

Table 4 FREQ course material download

|  |  |
| --- | --- |
| Requirement id | FREQ-5 |
| Source | Student |
| Requirement | The system shall allow student view result |
| Description | student view result from the system added by instructor |
| Category | Viewing result |
| Priority | medium |

Table 5 FREQ view course result

|  |  |
| --- | --- |
| Requirement id | FREQ-6 |
| Source | Registrar officer |
| Requirement | The system shall allow registrar officer add student |
| Description | Student registers to the system. |
| Category | Add student |
| Priority | High |

Table 6 FREQ add student

|  |  |
| --- | --- |
| Requirement id | FREQ-7 |
| Source | Instructor |
| Requirement | The system shall allow instructor add result |
| Description | instructor add student result to the system |
| Category | Add result |
| Priority | High |

Table 7 FREQ Add course result

|  |  |
| --- | --- |
| Requirement id | FREQ-8 |
| Source | Academic dean |
| Requirement | The system shall allow academic dean to register instructor |
| Description | Academic dean register instructor to the system |
| Category | Register instructor |
| Priority | High |

Table 8 FREQ Register Instructor

|  |  |
| --- | --- |
| Requirement id | FREQ-9 |
| Source | Academic dean |
| Requirement | The system shall allow academic dean to assign instructor |
| Description | Academic dean assign instructor in the system to classes |
| Category | Assign instructor |
| Priority | High |

Table 9 FREQ Assign Instructor

|  |  |
| --- | --- |
| Requirement id | FREQ-10 |
| Source | Academic dean |
| Requirement | The system shall allow academic dean to add notice |
| Description | Academic dean add notice to the system for users |
| Category | Add notice |
| Priority | Medium |

Table 10 FREQ Add notice

|  |  |
| --- | --- |
| Requirement id | FREQ-11 |
| Source | Academic dean |
| Requirement | The system shall allow academic dean to add course |
| Description | Academic dean add course to the system |
| Category | Add course |
| Priority | Medium |

Table 11 FREQ Add course

|  |  |
| --- | --- |
| Requirement id | FREQ-12 |
| Source | All user |
| Requirement | The system shall allow all user to logout |
| Description | All user leave the system after finish his/her task |
| Category | Logout |
| Priority | High |

Table 12 FREQ logout

### System Use case

#### Use case diagram

A[UML](https://en.wikipedia.org/wiki/Unified_Modeling_Language)use case diagram is the primary form of system/software requirements for a new software program under developed. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (such as UML). A key concept of use case modeling is that it helps us design a system from end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior [2].

The use case diagram shows the overall activities of these system users, it represents the functions that the actor should perform and this diagram also shows the relationships of actors that makes them common actors on specific use case. The diagram consist the system area in order to hold all use cases, which shows the boundaries of the system.

In the proposed system there are actors who make interact with the system via the use case. We include 5 actors in the proposed system

* **System admin:** Is Person who has a big authority on the system and managing exercise over the system and manages the user.
* **Instructor:** Is a person who has the privilege of uploading, and updating any learning resources and adds exercise for students.
* **Student:** Person who uses the system in order to download file and use what was uploaded by instructor.
* **Registrar officer:** Person who uses the system in order to recorded or approves student information.
* **Academic dean:** Person who controls the teaching and learning processes in the system.

The following figure shows use case diagram of the proposed system.



Figure 1 Use case diagram

#### Use case documentation

This sub topic describes in detail about the use case and actors in the use case diagram, it includes the name of use case, description about the use case, the name of the actor who act on the use case, what precondition before the actor acts on the use case, post condition after the actor acts on the use case, basic course of action which describes the interaction between the actor and the system when the actor acts on the use case, and finally alternative course of action.

The following table shows use case documentation of the system.

|  |  |  |
| --- | --- | --- |
| Use case number | UC 01 | |
| Use case name | Login | |
| Actor | Academic dean, instructor, students , registrar officer and 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 13 Use case documentation of login

|  |  |  |
| --- | --- | --- |
| Use case number | UC 02 | |
| Use case name | Logout | |
| Actor | Academic dean, instructor, students , registrar officer and 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 14 Use case documentation of logout

|  |  |  |
| --- | --- | --- |
| 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 15 Use case documentation of download module

|  |  |  |
| --- | --- | --- |
| Use case number | UC 04 | |
| Use case name | View notice | |
| Actor | Students and instructor. | |
| Description | The user looks all the information added by the academic dean in the system. | |
| Precondition | The notice must be posted before | |
| Post condition | The user observes the notices. | |
| Basic course of action | User action | System response |
| 1.The user opens the system   1. The user click on view notice link.   5. the user view the posted notice  6. End use case. | 2. The system displays the home page.  4. The system displays available notice that is posted by the academic dean. |
| Alternative course of action | If connection is failed before open the notice, the system displays connection fail. Then turn back to step 3 basic course of action. | |

Table 16 Use case documentation of view notice

|  |  |  |
| --- | --- | --- |
| 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 17 Use case documentation of view Result

|  |  |  |
| --- | --- | --- |
| Use case number | UC 06 | |
| Use case name | View course | |
| Actor | Students. | |
| Description | The user looks course added by the academic dean in the system. | |
| Precondition | The user must be login to the system and the course must be added before. | |
| Post condition | The user access and know the course. | |
| Basic course of action | User action | System response |
| 1. The user click on view course link.   3.The user view course  4. End use case. | 2. The system displays available course that is added by the academic dean. |
| Alternative course of action | If course is not added, the system displays the message “there is no course available”. | |

Table 18 Use case documentation of view course

|  |  |  |
| --- | --- | --- |
| Use case number | UC 07 | |
| Use case name | Register instructors | |
| Actor | Academic dean | |
| Description | Registering instructor in the system | |
| Precondition | Academic dean must be login to the system. | |
| Post condition | The instructor successfully register in the system by academic dean | |
| Basic course of action | User action | System response |
| 1. The user clicks register link.  3.The user fill the form and click register button  5. End use case. | 2. The system displays registration form.  4. The system registers instructor |
| Alternative course of action | If the user fill incorrect info, the system displays an error message, then go back to step 3 of basic course of action. | |

Table 19 Use case documentation of Register Instructor

|  |  |  |
| --- | --- | --- |
| Use case number | UC 08 | |
| Use case name | Assign instructor | |
| Actor | Academic dean | |
| Description | assigning instructor to class in the system | |
| Precondition | Academic dean 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 Academic dean 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 academic dean 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 20 Use case documentation of Assign Instructor

|  |  |  |
| --- | --- | --- |
| Use case number | UC 9 | |
| Use case name | Add notice | |
| Actor | Academic dean | |
| Description | Posting of notice in the system | |
| Precondition | Academic dean must be login to the system | |
| Post condition | The notice successfully post in the system. | |
| Basic course of action | User action | System response |
| 1. The user clicks post notice link.  3. The user attaches notice and click add notice button.  5. End use case. | 2. The system displays notice post area.  4. The system posts notice |
| Alternative course of action | IF connection is fail, try again. | |

Table 21 Use case documentation of Add notice

|  |  |  |
| --- | --- | --- |
| Use case number | UC 10 | |
| Use case name | Add course | |
| Actor | Academic dean | |
| Description | Adding course in the system | |
| Precondition | Academic dean must have user name and password | |
| Post condition | The course successfully added in the system by academic dean | |
| Basic course of action | User action | System response |
| 1.The academic dean 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 academic dean 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 22 Use case documentation of Add course

|  |  |  |
| --- | --- | --- |
| Use case number | UC 11 | |
| Use case name | Remove notice | |
| Actor | Academic dean | |
| Description | Removing of notice in the system that the notice is expired | |
| Precondition | Academic dean must have user name and password and the notice has been posted before | |
| Post condition | The expired notice is successfully remove from the system by academic dean | |
| Basic course of action | User action | System response |
| 1.The academic dean login to the system  3. The user clicks notice link.  5. The user click remove button.  7. End use case. | 2. The system displays the academic dean page.  4. The system displays post notice.  6. The system remove notice |
| Alternative course of action | IF connection is fail, try again. | |

Table 23 Use case documentation of Remove notice

|  |  |  |
| --- | --- | --- |
| Use case number | UC 12 | |
| Use case name | Add student | |
| Actor | Registrar officer | |
| Description | Adding of student to the system they can pass entrance exam | |
| Precondition | Registrar officer must be login to the system and the student must pass the entrance exam | |
| Post condition | The student is added in to the system by registrar officer | |
| Basic course of action | User action | System response |
| 1. The user clicks add link.  3. The user fill student and click add button.  5. End use case. | 2. The system displays the registration form.  4. The system adds the student |
| Alternative course of action | If the user misses some information to fill, error message display and turn back to step 3 basic course of action | |

Table 24 Use case documentation of Add student

|  |  |  |
| --- | --- | --- |
| Use case number | UC 13 | |
| Use case name | Create account | |
| Actor | System Administrator | |
| Description | The system administrator creates user account to students, instructors, registrar officer and academic dean to give authorization. | |
| Precondition | Users registered to the system | |
| Post condition | Users account Created. | |
| User Action | System Response |
| Basic course of action | 1. The system Administrator login to the system  3.The system administrator click on create account link  5. System Administrator Fill create account form.  6. click on create button  8. End of use case. | 2. The system display system administrator home page.  4. The system display creates account form.  7. The system displays create successful message. |
| Alternative course of action | If the system administrator fills incorrect information, the system displays error message, and go back to step 5 of basic course of action. | |

Table 25 Use case documentation of create account

|  |  |  |
| --- | --- | --- |
| Use case number | UC 14 | |
| Use case name | Deactivate user account | |
| Actor | System administrator | |
| Description | The system administrator block accounts of the academic dean, students, registrar officer 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 26 Use case documentation of deactivate user account

|  |  |  |
| --- | --- | --- |
| Use case number | UC 15 | |
| Use case name | Activate user account | |
| Actor | System administrator | |
| Description | The system administrator unblocks or activate user accounts of the academic dean, students, registrar officer 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 27 Use case documentation of activate user account

|  |  |  |
| --- | --- | --- |
| 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 28 Use case documentation of Add student result

|  |  |  |
| --- | --- | --- |
| 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 29 Use case documentation of upload learning material

|  |  |  |
| --- | --- | --- |
| Use case number | UC 18 | |
| Use case name | Add department | |
| Actor | Academic dean | |
| Description | Adding department in the system | |
| Precondition | Academic dean must have user name and password | |
| Post condition | The department successfully added in the system by academic dean | |
| Basic course of action | User action | System response |
| 1.The academic dean 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 academic dean 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 | |

Table 30 Use case documentation of add department

### Business Rule Documentation

A business rule defines or constrains one aspect of the business that is intended to assert business structure or influence the behavior of the business. Business rules often focus on access control issues. Business rules may also pertain to business calculations [9].

In business rule documentation the idea includes

**Name**: The name should give you a good idea about the topic of the business rule.

**Identifier:** it shows the id of the business rules

**Description**: The description defines the rule exactly.

**Source (optional):** The source of the rule is indicated so it may be verified (it is quite common that the source of a rule is a person, often one of the project stakeholders, or a team of people).

The following table shows business rule of the project.

|  |  |
| --- | --- |
| Identifier | BR 001 |
| Name | every applicant must provide his/her educational report to register |
| Description | Any applicant who wishes to join Bahir Dar University should provide the necessary educational information. |
| Source | University |

Table 31 BR of provide educational report to register

|  |  |
| --- | --- |
| Identifier | BR 002 |
| Name | Determine the need of ID |
| Description | Any student who is learning in the Bahir Dar university distance and continuing education should have a unique ID. |
| Source | University |

Table 32 BR of determine the need of ID

|  |  |
| --- | --- |
| Identifier | BR 003 |
| Name | User must have his/her account |
| Description | Every user must have his/her user account in order to access the service and perform their task. |
| Source | Group member |

Table 33 BR of user account

|  |  |
| --- | --- |
| Identifier | BR 004 |
| Name | Determine educational fee should be paid before due date |
| Description | Every student should pay his/her educational fee before the due date to be unpunished. |
| Source | University |

Table 34 BR of determine educational fee

|  |  |
| --- | --- |
| Identifier | BR 005 |
| Name | Students may access modules |
| Description | Student must login to the system with the correct user name and password and pay his/her fee in order to access the uploaded material like module, tutor, assignments etc. |
| Source | Group member |

Table 35 BR of accessing module

|  |  |
| --- | --- |
| Identifier | BR 006 |
| Name | Instructors are allowed to upload learning material. |
| Description | instructors must be login first in order to upload learning materials for the student in the system |
| Source | Group member |

Table 36 BR of upload learning material

### User Interface prototype

### 

The user interface prototype is built early, before the whole system is analyzed, designed and implemented.

The main purpose of creating a user-interface prototype is to be able to expose and test both the functionality and the usability of the system before the real design and development starts. This way, you can ensure that you are building the right system, before you spend too much time and resources on development.

The figure shows the diagram of user interface prototype.



Figure 2 User Interface prototype

### Activity Diagram

UML Activity diagrams are used to document the logic of a single operation or method, a single use case (may be the basic course of action or the alternate course of action) or the flow of a logic of a business process. They are the object oriented equivalent of flow charts and data flow diagrams in the structured development approach.

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows).Activity diagrams show the overall flow of control [3].

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

* rounded rectangles represent actions;
* diamonds represent decisions;
* bars represent the start (split) or end (join) of concurrent activities;
* a black circle represents the start (initial node) of the workflow;
* An encircled black circle represents the end (final node).

The following figure shows activity diagram of each activity in the system.

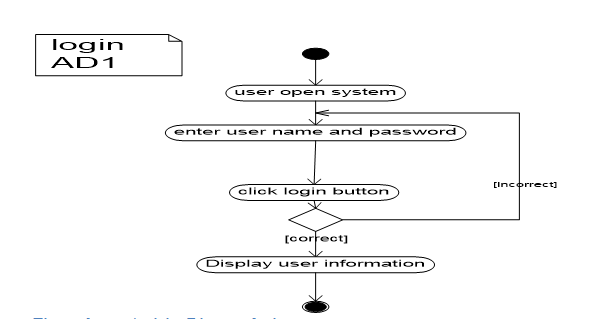


Figure 3 Activity Diagram login

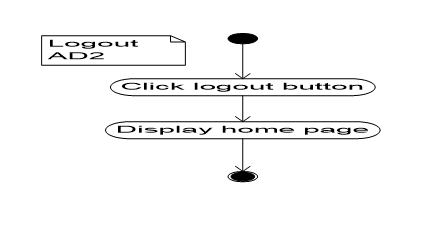


Figure 4 Activity Diagram logout



Figure 5 Activity Diagram add student



Figure 6 Activity Diagram download module



Figure 7 Activity Diagram view notice



Figure 8 Activity Diagram view course result



Figure 9 Activity Diagram view course



Figure 10 Activity Diagram register Instructor



Figure 11 Activity Diagram Assign Instructor



Figure 12 Activity Diagram view notice



Figure 13 Activity Diagram add course



Figure 14 Activity Diagram create account



Figure 15 Activity Diagram deactivate user account

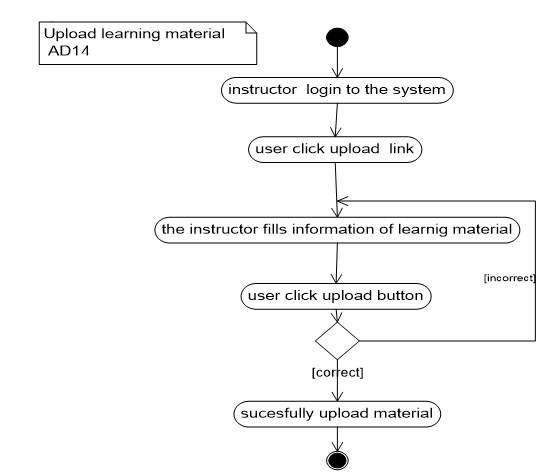


Figure 16 Activity Diagram upload learning material

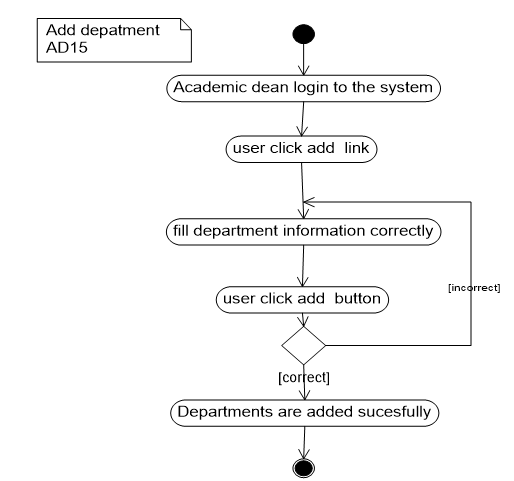


Figure 17 Activity diagram add department

### Sequence diagram

A sequence diagram is an [interaction diagram](https://en.wikipedia.org/wiki/Interaction_diagram) that shows how objects operate with one another and in what order. It is a construct of a [message sequence chart](https://en.wikipedia.org/wiki/Message_sequence_chart).

A 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 scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams **[4]**.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

The following figure shows the sequence diagram.



Figure 18 Sequence diagram login

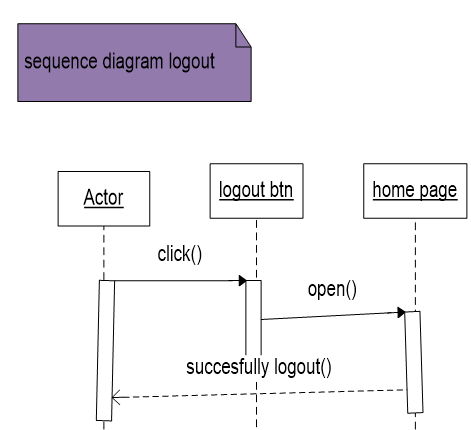


Figure 19 Sequence diagram logout



Figure 20 Sequence diagram download



Figure 21 Sequence diagram create account



Figure 22 Sequence diagram upload learning material



Figure 23 Sequence diagram add dept.



Figure 24 Sequence diagram add course



Figure 25 Sequence diagram register Instructor



Figure 26 Sequence diagram Assign Instructor



Figure 27 Sequence diagram view notice



Figure 28 Sequence diagram view result

### Analysis Class Model

Analysis Class model/diagram is static model that shows the classes and the relationships among classes that remain constant over the time. Class is the main building block of class diagram, which stores and manages information in the system.

The figure shows analysis class models.



Figure 29 Analysis Class Model

### Logic model/pseudo code

Pseudo code is a simple way of writing programming code in English. Pseudo code is not actual programming language. It uses short phrases to write code for programs before you actually create it in a specific language. Once you know what the program is about and how it will function, then you can use pseudo code to create statements to achieve the required results for the program.

**ID: AL 01**

Name: Pseudo code **for Login**

**Login ()**

**Begin**

**{**

**Enter user name;**

**Enter password;**

**Connect to database**

**If (user name&&password==true)**

**Display home page;**

**Else**

**Display error massage;**

**}**

**End**

**ID: AL 02**

Name: Pseudo code **register instructor**

**Register instructor ()**

**Begin**

**{**

**Login ()**

**Enter instructor information;**

**Connect to database**

**If (instructor information ==true)**

**Register successfully;**

**Else**

**Display error massage;**

**}**

**End**

**ID: AL 03**

Name: Pseudo code **for Create Account**

**Create user account ()**

**Begin**

**{**

**Enter user information;**

**Validate input**

**Connect to Database**

**If (information==true)**

**Insert to Database;**

**Display success massage;**

**Else**

**Display error message**

**}**

**End**

**ID: AL 04**

Name: Pseudo code **for add course**

**Add course ()**

**Begin**

**{**

**Enter course information;**

**Connect to database**

**If (course information==true)**

**Display message “successfully added”;**

**Else**

**Display error massage;**

**}**

**End**

**ID: AL 05**

Name: Pseudo code **for Remove notice**

Remove notice ()

Begin

{

Login ()

Enter a key word to search the notice;

Check if the key word is valid;

If (keyword==true) {

Click on delete button;

If (delete success) {

Message 'deleted';

}

Else {

Message 'not deleted' try again;

}

}

Else

{

Message 'please enter valid key word';

}

}

**End**

**ID: AL 06**

Name: Pseudo code **for add department**

**Add department ()**

**Begin**

**{**

**Enter department information;**

**Connect to database**

**If (department information==true)**

**Display message “successfully added”;**

**Else**

**Display error massage;**

**}**

**End**

#### ID: AL07

#### Name: Pseudo code for Upload

#### Upload ()

**Begin**

**{**

Select course

Enter module name;

If (filled info==true)

{

Add module to DB is successful;

}

Else

{

“Fill the valid information”

**}**

**End**

## Nonfunctional Requirement

Non-functional requirements are requirement that can support and give more quality for the system. The proposed system has the following Non-Functional Requirements to achieve its functionality.

The following table shows nonfunctional requirement of the system.

|  |  |
| --- | --- |
| ID | NFR-1 |
| Source | user |
| Requirement | Usability |
| Description | The system is easy to learn and operate. The User interface for this system will be simple and clear. Web based distance learning services are easy to gain and use i.e. the service doesn’t require special training. |
| Priority | Medium |

Table 37 NFR Usability

|  |  |
| --- | --- |
| ID | NFR-2 |
| Source | user |
| Requirement | Availability |
| Description | This system is available in everywhere (where internet/intranet service reach) and at all time for those who have access to use the system. |
| Priority | Medium |

Table 38 NFR Availability

|  |  |
| --- | --- |
| ID | NFR-3 |
| Source | user |
| Requirement | Performance |
| Description | The system will have good performance i.e. fast response time and optimal workload. |
| Priority | Low |

Table 39 NFR Performance

|  |  |
| --- | --- |
| ID | NFR-4 |
| Source | Developer |
| Requirement | Security |
| Description | Use very strong user name and password in order to secure the system. And also encrypts user’s password on database. So it is designed to be very secure by providing a login feature which authenticates the user by means of a user name and password with which user will be able to login to his/her respective pages and use the system as required. |
| Priority | High |

Table 40 NFR Security

|  |  |
| --- | --- |
| ID | NFR-3 |
| Source | Developer |
| Requirement | Documentation |
| Description | The system contains the required documents needed to implement the project. |
| Priority | Low |

Table 41 NFR Documentation

## System Requirement

### Hardware requirements

The following hard ware requirement is needed for the system

* Processor speed=3.0 GHZ and above
* RAM=4 GB and above
* Hard disk space=250GB and above

### Software requirements

* Web server: Apache HTTP server
* Database management system(DBMS): MYSQL database
* Operating system: Windows (7, 8, 10), Linux.
* Client side application (Browser): currently available/functional browsers (like Moziliafirfox, spark, internet explorer).

## Key abstraction with CRC analysis

A Class Responsibility Collaborator (CRC) model is a collection of standard index cards that have been divided into three sections. They are class, responsibility and collaboration. A class represents a collection of similar objects, a responsibility is something that a class knows or does, and a collaborator is another class that a class interacts with to fulfill its responsibilities.

The table shows the CRC of a class.

|  |  |
| --- | --- |
| System admin | |
| Id  Name  Age  Sex  E\_mail  Manage Account () | Student  Instructor  Academe dean  Registrar office |

|  |  |
| --- | --- |
| Instructor | |
| Id  Name  Age  Sex  E\_mail  View notice ()  Upload module ()  Add student result () | System admin  Course  Module  Academic dean |

|  |  |
| --- | --- |
| Student | |
| Id  Name  Age  Sex  E\_mail  Register ()  View result  View course  View notice  Download module | System adimn  Instructor  Register office  Course  Module  Department |

|  |  |
| --- | --- |
| Registrar office | |
| Id  Name  Age  Sex  E\_mail  Approve student () | Student  System admin |

|  |  |
| --- | --- |
| Academic dean | |
| Id  Name  Sex  Age  E\_mail  Register instructor ()  Assign instructor ()  Add notice ()  Add course ()  Update notice ()  Add department ()  Delete department () | System admin  Course  Department  Instructor |

|  |  |
| --- | --- |
| Module/material | |
| M\_id  M\_title  Author |  |

|  |  |
| --- | --- |
| department | |
| Id  Name |  |

|  |  |
| --- | --- |
| course | |
| C\_id  C\_name  C\_ch |  |

|  |  |
| --- | --- |
| Schedule | |
| Dept.  Year  Semester  date |  |

# **Chapter 3: System Design**

Systems design is the process of defining elements of a system like architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization

## Architectural Design

Systems architecture is the [conceptual model](https://en.wikipedia.org/wiki/Conceptual_model) that defines the [structure](https://en.wikipedia.org/wiki/Structure), [behavior](https://en.wikipedia.org/wiki/Behavior), and more [views](https://en.wikipedia.org/wiki/View_model) of a [system](https://en.wikipedia.org/wiki/System). An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the [structures](https://en.wikipedia.org/wiki/Structure) and [behaviors](https://en.wikipedia.org/wiki/Behavior) of the system.

A system architecture can comprise system [components](https://en.wikipedia.org/wiki/System), the expand systems developed, that will work together to implement the overall system.

### Component modeling

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams. It does not describe the functionality of the system but it describes the components used to make those functionalities.

In the [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language), a component diagram depicts how [components](https://en.wikipedia.org/wiki/Component_(UML)) are wired together to form larger components or [software systems](https://en.wikipedia.org/wiki/Software_system).

The figure shows the component diagram of a system.



Figure 30 Component modeling

### Deployment Modeling

Deployment diagram is structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deploymenttargets. Artifacts represent concrete elements in the physical world that are the result of a development process.

A deployment diagram in the [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language) models the physical deployment of [artifacts](https://en.wikipedia.org/wiki/Artifact_(UML)) on [nodes](https://en.wikipedia.org/wiki/Node_(UML)).

The figure shows below are deployment diagram of the system.



Figure 31 Deployment Modeling

## Detail Design

### Design class model

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.

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed [6].

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved.

In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the attributes of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes [7].

The figure shows below are design class diagram of the system.



Figure 32 Design class model

### Persistent model

Persistence refers to the characteristic of state that outlives the process that created it. This is achieved in practice by storing the state as data in computer data storage. Programs have to transfer data to and from storage devices and have to provide mappings from the native programming-language data structures to the storage device data structures [9].

Designing of a Persistence Model can be accomplished through model-driven design or database modeling

The figure shows below are Persistent model of the system.



Figure 33 Persistent model

## User Interface Design

User interface design (UI design) refers to the design of various types of software and hardware interfaces through which users interact with computers and other technologies [10].

User interface design (UI) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices with the focus on maximizing usability and 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 goal [11].

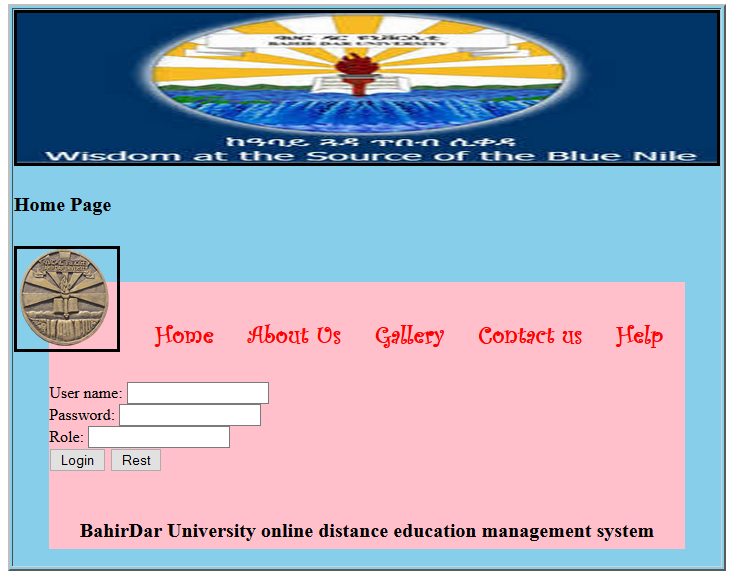


Figure 34 User Interface Design of home page

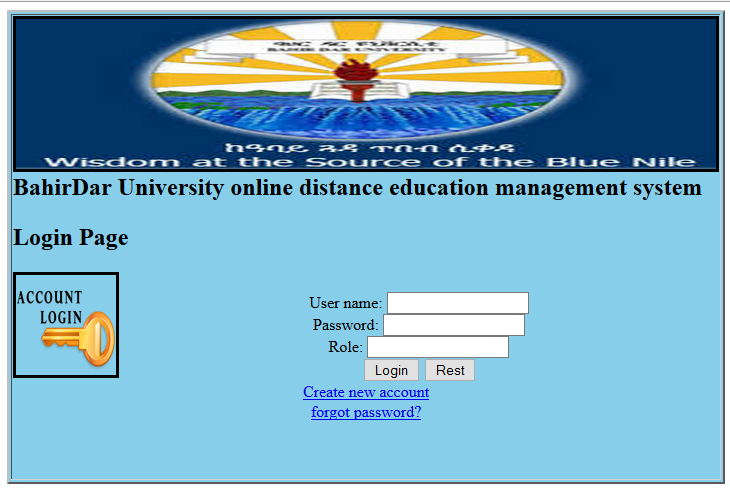


Figure 35 User Interface Design login page

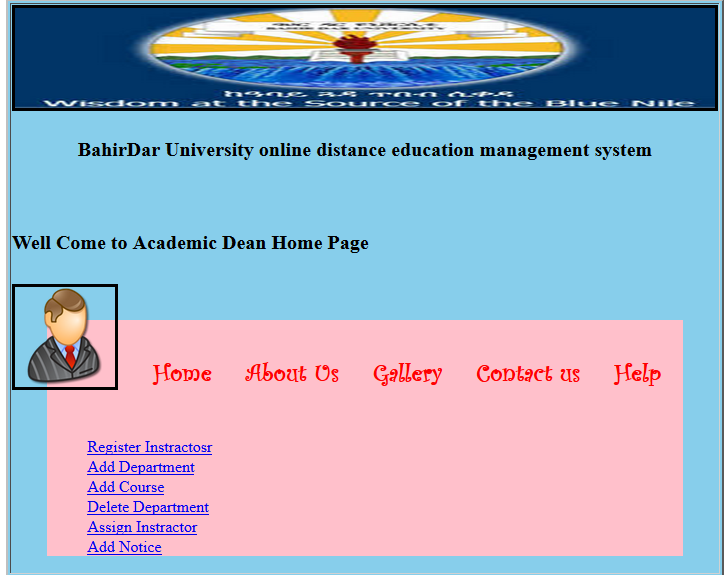


Figure 36 User Interface Design academic dean home page

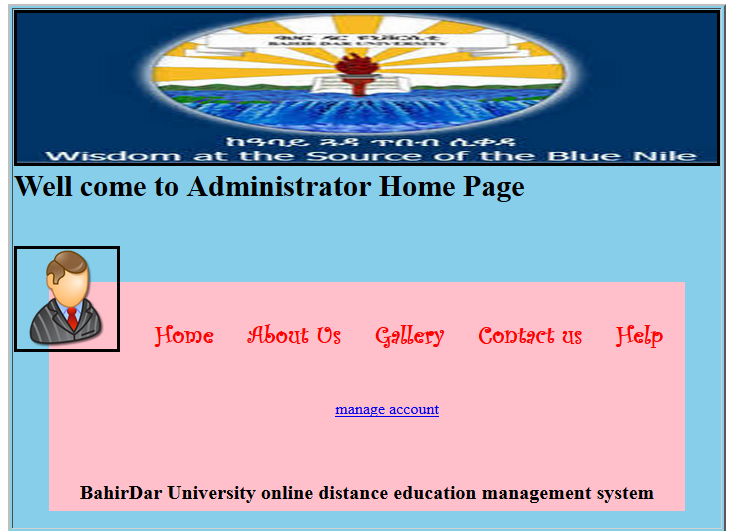


Figure 37 User Interface Design system admin home page

## Access control and security

Access controls are security features that control how users and systems communicate and interact with other systems and resources. Access is the flow of information between a subject and an object. It is all about the act of ensuring that an authenticated user accesses only what they are authorized to. In the systems, different actors have access to different functionality and data. The project is divided into five modules namely,

1. Administrator Module

The administrator is the person who is responsible for security issue of the system. He has the authority of manage and control user accounts. He can create new user account, deactivate and activate user account.

1. Instructor Module

An instructor is the person who is responsible for inserting student data into the database like course result, uploading learning materials to the students. He has also access grant to view information in the system. To do this he/she must have his/her own account user name and password.

1. Student Module

A student is a user who has access grant to uses the system for down load learning material from the system. He has also right to view information’s about the university which concern them. To do this he/she must have his/her own account user name and password.

1. Academic dean module

Academic deanis the person who is responsible for controlling over all teaching and learning processes and can manage course, departments and instructors. To do this he/she must have his/her own account user name and password.

1. Registrar officer module

Registrar officeris the person who is responsible for controlling students like accepting and ignoring applicant students. He accept student they fulfill registration criteria and ignore if they not fulfill. To do this he/she must have his/her own account user name and password.

# Chapter Four

# Implementation and Testing

## Implementation

Implementation refers to the Coding of all documents gathered starting from requirement analysis to Design phase. In the implementation phase all the programs are written, database is created, user operational document is written, users are trained, and the system tested with operational data.

The system developers are in a position of converting all documents gathered and designed into the code so that the system will be implemented for the user to be used for the purpose it developed. This document will be implemented next phase that is project two. For the implementation of this system the user must have a server on which the system will be hosted.

## Testing

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. Testing’s should systematically uncover different classes of errors in a minimum amount of time with a minimum amount of efforts.

**Testing procedures**

Before directly deploying this system, the team will perform different testing for its functionality and meeting customers need. First the team tests each unit at each phase. So, if a problem is encountered it will immediately fixed. Then the team will perform an integration testing to check whether the system meets all the functional requirements. System will be tested using the following system testing procedures

**Unit testing**

Every module of the System is separately tested. I.e. the team tests every module by applying some selection mechanism. Through this mechanism every modules gets tested. If an error occurs correction will be taken without affecting another module. We have tried to test UI screens of our system that needs to verify screen elements that appears on the screen.

**Integrating testing**

In this testing part, all the modules will be combined together and tested it for its fitness with each other and with the systems functionality. If error occurs in combining them, the module with problem will be identified and recombined. Both units testing and integrated testing are performed by all team members at the work place.

**System Testing**

Here we compile the whole system staring from initial and proceed testing the whole system to check out for the errors and flow control of the system.

# Chapter Five

# Conclusions and Recommendations

The development and advancement of computer technology makes computers to be a part of everyday human life activities. Education is an area where the human is involved in a day to day activity of his life. It is an area which requires due attention, for it deals with behavioral, attitude and skill changes. The same is true for the use of computer in education. This project has enabled the delivery of learning materials and teaching processes to be efficient and it has also achieved interactivity among students and instructors. This project is going to develop using the PHP web technology. This technology choice has enabled the work to have portability, extendibility and security. The portability enables the work to be deployed on a given platform. The extendibility can be expressed as features for the work to tolerate the future expansions on the area. The security features of the PHP language can be incorporated to the level of requirement in need.

The system that we have tried to develop is not the whole system of the university .Because of time and budget limitation we can’t develop all parts of the system, but we have tried to automate some sub systems and functionalities. The following functionalities can’t be automated because of time and budget limitations that we have discussed above.

* **Online examination**
* **Job Vacancy**
* Online payment etc.

**Therefore, others who are interested to develop on this system of the university can get some initial idea about the system will improve the system.**

# References

[1] http://bdu.edu.et/cde/?q=content/background/referenced date 15/11/2017

[2]https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/referenceddate25/11/2017

[3] https://en.wikipedia.org/wiki/Activity\_diagram/referenced date10/12/2017

[4]https://en.wikipedia.org/wiki/Sequence\_diagram/refernced date10/12/2017

[5]http://searchsoftwarequality.techtarget.com/definition/requirements-analysis/referenceddate 28/12/2017

[6] https://en.wikipedia.org/wiki/Class\_diagram/referenceddate 28/12/2017.

[7]http://searchmicroservices.techtarget.com/definition/class-diagram/referenceddate 28/12/2017.

[8]http://agilemodeling.com/artifacts/businessRule.htm/referenceddate30/03/2010EC

[9] https://en.wikipedia.org/wiki/Persistence\_ (computer science) #cite\_note-1/referenced date 03/01/2018.

[10].https://www.techopedia.com/definition/22776/user-interface-design-ui-design/referenced date 03/01/2018.

[11]. https://en.wikipedia.org/wiki/User\_interface\_design/referenced date 15/01/2018.

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

13. Object Oriented System Analysis and Design learning materials prepared by Instructor asegahagn.