**Chapter Three**

**System Analysis**

## Introduction

In chapter two, we have discussed the different activities of existing system and the main processes involved in managing the continuing and distance office activities. In this chapter the functional requirements of the proposed system will be modeled using UML models with different types of diagrams.

## Use case Model

**To model a system the most important aspect is to capture the dynamic behavior. To clarify in details, dynamic behavior means the behavior of the system when it is running or operating. So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.**

**These internal and external agents are known as actors. So use-case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system or subsystem of an application. A single use case diagram captures a particular functionality of a system. So to model the system we used the following use case diagrams.**

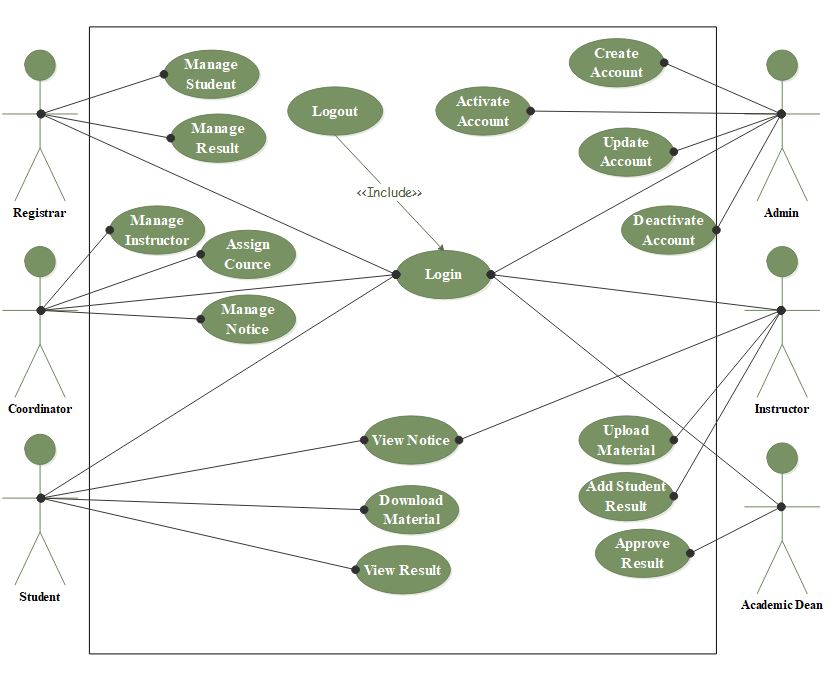
### ****Actor Identification****

**The purpose of actor analysis is to identify all of the actors that interact with the system. An actor has a role in that interacting with the system.**

**The actors that interact with the system are:-**

* Academic dean
* Instructor
* Student
* Registrar officer
* System Administrator
* Distance Coordinator

### ****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, distance coordinator 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 1 Use case documentation of login

|  |  |  |
| --- | --- | --- |
| Use case number | UC 02 | |
| Use case name | Logout | |
| Actor | Academic dean, instructor, students , registrar officer, distance coordinator 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 2 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 distance coordinator 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 | Distance Coordinator | |
| 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 | Admin | |
| 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 | Distance Coordinator | |
| 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

## 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.

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;
* Black circle represents the start (initial node) of the workflow;
* Encircled black circle represents the end (final node).

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

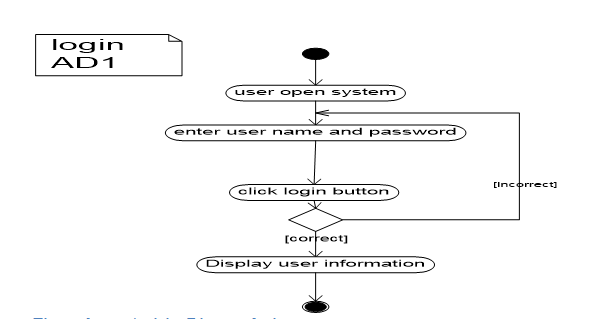


Figure 3 Activity Diagram login

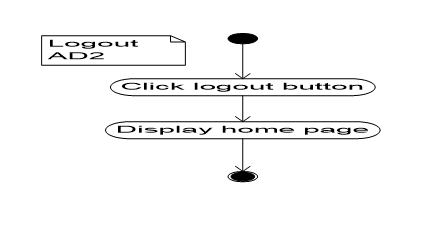


Figure 4 Activity Diagram logout



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

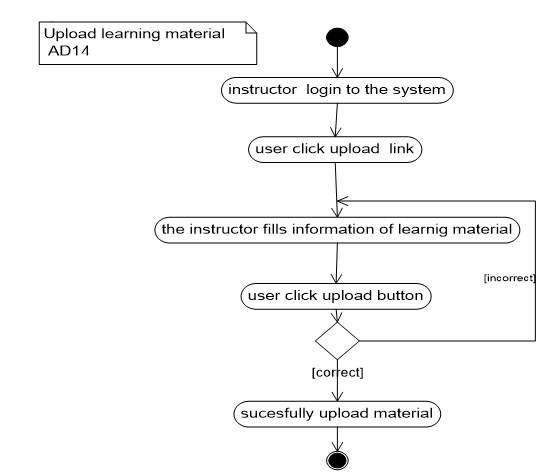


Figure 16 Activity Diagram upload learning material

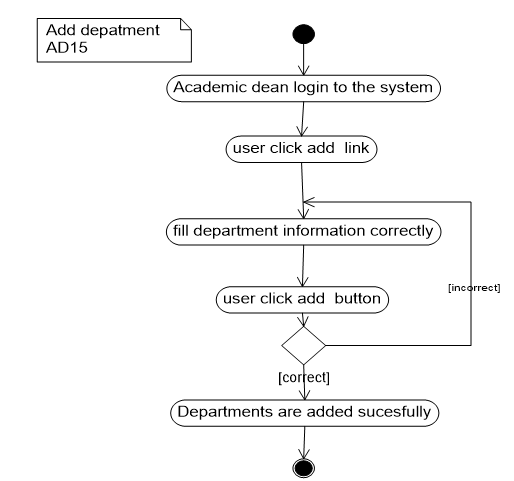


Figure 17 Activity diagram add department

## Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a 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.

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

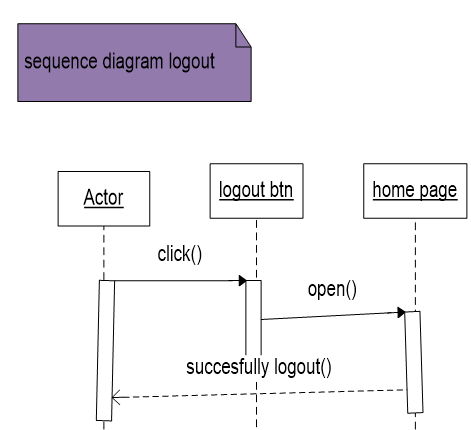


Figure 19 Sequence diagram logout

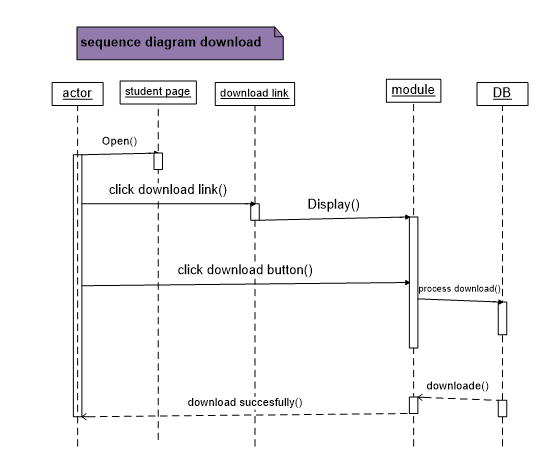


Figure 20 Sequence diagram download



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

**Chapter Four**

**System Design**

### Introduction

**System design is the transformation of the analysis model into a system design model. System design is the first part to get into the solution domain in a software development. This chapter focuses on transforming the analysis model into the design model that takes into account the nonfunctional requirements and constraints described in the problem statement and requirement analysis sections discussed earlier.**

### Design Goals

**Design goals describe the qualities of the system that the developers should consider. These goals can be drawn from the non-functional requirements already discussed. The design goals can be generally grouped into five categories. These are: Performance Criteria, Dependability Criteria, Cost Criteria, Maintenance Criteria, and End User Criteria.**

**Performance: The system should respond fast with high throughput, i.e. It should perform searching information, uploading and downloading modules, registration processing and generating report ina time less than a minute**.

**Dependability: The office needs the system to be highly dependable. The system should be robust (forceful) i.e. it should be able to carry on invalid user inputs, fault tolerant, reliable and available. The system shouldn’t allow non-authorized users to access students’ personal data or modify.**

Cost: **The system should be developed, deployed, administered and maintained with minimum cost possible.**

**Maintenance: The system should be easily extensible to modify the uploading materials, add new functionality, portable to different platforms. The code for the system should be easily readable, understandable and should be easily mapped to specific requirements.**

**End User Criteria: The system should have simple and understandable graphical user interface such as forms and buttons which have descriptive names. It should give reliable response for each user request at least before the session expires.**

**Usability: Usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. From the end users’ perspective the system should be designed in such a way that it is easy to learn and use, efficient and having few errors if any.**

## Architectural Design

Systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can comprise system components, 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, a component diagram depicts how components are wired together to form larger components or software systems.

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

## Design of Class Diagram

**Diagrams are used to represent the structure of the system in terms of objects, their notes and nature of relationship between classes. It shows the static features of the actors and do not represent any particular processing.**



## Collaboration Diagram

Collaboration diagram is another form of interaction diagram. It represents the structural organization of a system and the messages sent/received. Structural organization consists of objects and links.

The purpose of collaboration diagram is similar to sequence diagram. But the specific purpose of collaboration diagram is to visualize the organization of objects and their interaction.



Figure 4.4.1 Collaboration diagram for upload module



Figure 4.4.2 Collaboration diagram for user login



Figure 4.4.3 Collaboration diagram for Download module



Figure 4.4.4 Collaboration diagram upload assignment



Figure 4.4.5 Collaboration diagram for download assignment

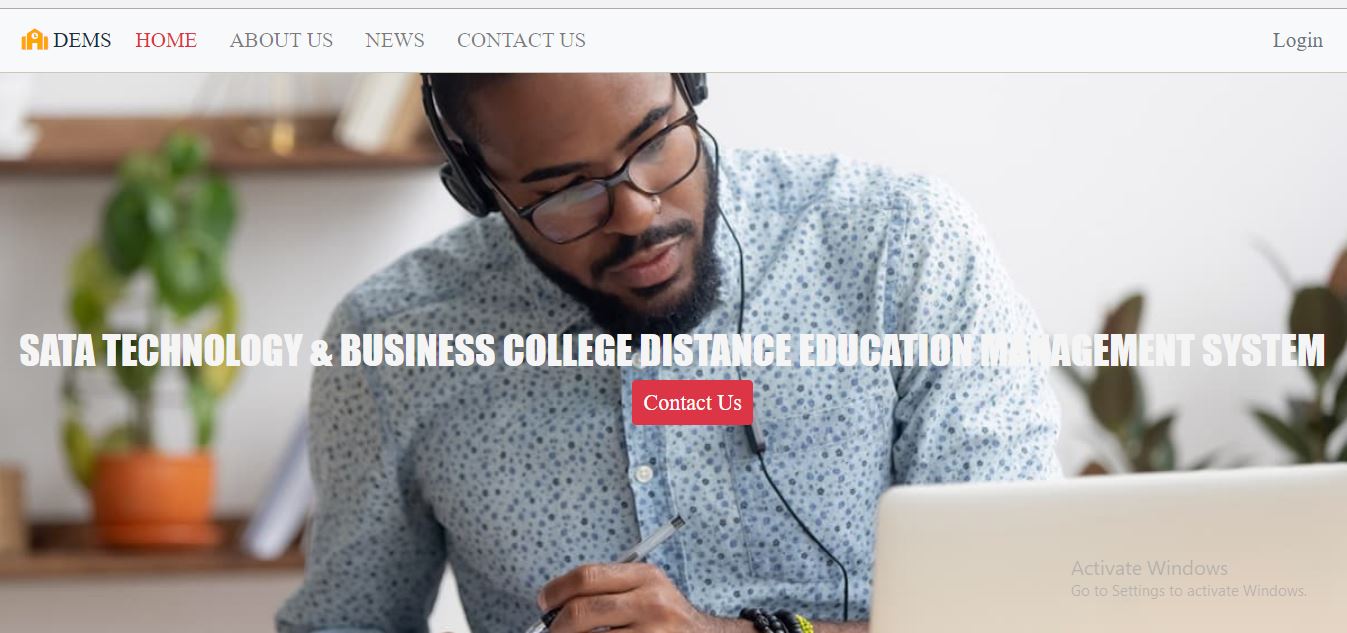


Figure 4.4.6Collaboration diagram for post information

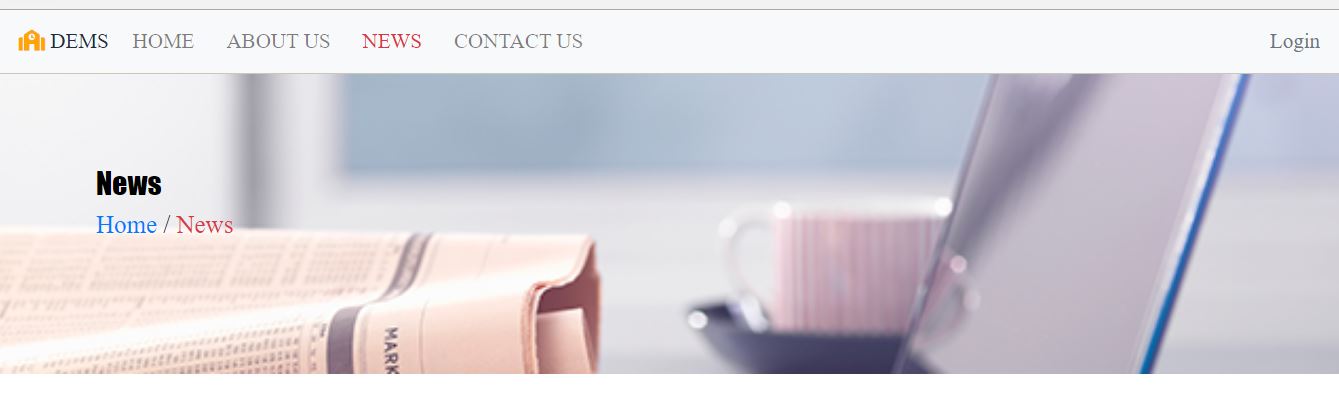
## User Interface Design

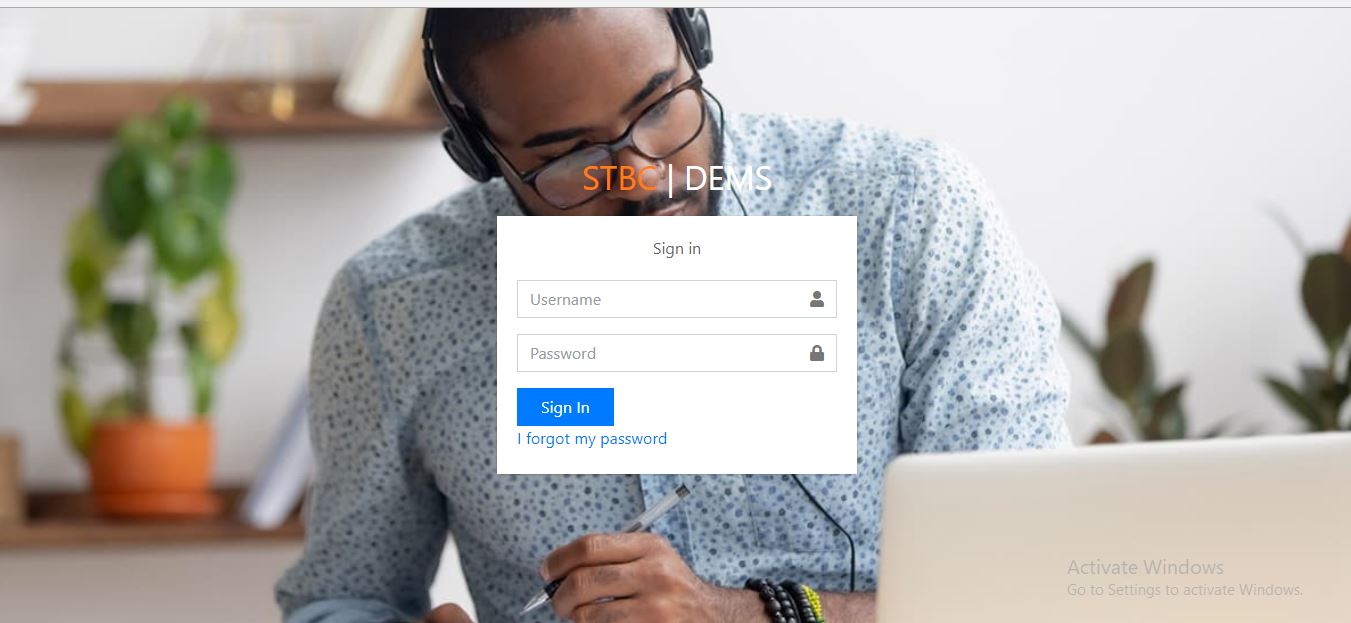
User interface design (UI design) refers to the design of various types of software and hardware interfaces through which users interact with computers and other technologies.

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.



**Figure** User Interface Design of Home

**Figure** *User Interface Design of Notice*



**Figure** *User interface design of login*

# Chapter Five

## 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 Six

## 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 college .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 college can get some initial idea about the system will improve the system.**

**REFERENCE**

1. Brueghel, Bernd (2000) ***Object oriented Software Engineering Conquering Complex and Changing System.*** Upper Saddle River: Prentic Hall.
2. **Books available in the library**
3. The ***Object primer: The application Developers Guide to Object Oriented and the UML***.2nd rev. Ed England: The Colombia University.
4. <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/referenceddate25/11/2017>
5. [https://en.wikipedia.org/wiki/Activity\_diagram/referenced date10/12/2017](https://en.wikipedia.org/wiki/Activity_diagram/referenced%20date10/12/2017)
6. [https://en.wikipedia.org/wiki/Sequence\_diagram/refernced date10/12/2017](https://en.wikipedia.org/wiki/Sequence_diagram/refernced%20date10/12/2017)
7. [http://searchsoftwarequality.techtarget.com/definition/requirements-analysis/referenceddate 28/12/2017](http://searchsoftwarequality.techtarget.com/definition/requirements-analysis/referenceddate%2028/12/2017)
8. [https://en.wikipedia.org/wiki/Class\_diagram/referenceddate 28/12/2017](https://en.wikipedia.org/wiki/Class_diagram/referenceddate%2028/12/2017).
9. [http://searchmicroservices.techtarget.com/definition/class-diagram/referenceddate 28/12/2017](http://searchmicroservices.techtarget.com/definition/class-diagram/referenceddate%2028/12/2017).
10. Agile modeling the Object primer 2nd edition Scott W.Ambler’s book.
11. Object Oriented System Analysis and Design learning materials prepared by Instructor asegahagn.