



WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION TECHNOLOGY

**Project Title: - School Management System for Aba Fransua
Preparatory School**

By

Team members.....Id No:

Damoze Bane..... CIR/180/10

Abebe Birhanu.....CIR/165/10

Beka Ahmed.....CIR/176/10

Project advisor Mr. Abraham Assefa

Wolkite University, Wolkite, Ethiopia

August 25, 2021

WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION TECHNOLOGY

**Project Title: - School Management System for Aba fransua
Preparatory School**

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

By

Team members.....Id No:

Damoze Bane..... CIR/180/10

Abebe Birhanu.....CIR/165/10

Beka Ahmed..... CIR/176/10

Project advisor Mr. Abraham Assefa

Wolkite University, Wolkite, Ethiopia

August 25, 2021

DECLARATION

This is to declare that this project work which is done under the supervision of Mr. Abraham Assefa, And having the title School Management System for Aba Fransua Preparatory School is the sole contribution of:

1. Damoze Bane
2. Abebe Birhanu
3. Beka Ahmed

No part of the project work has been reproduced illegally which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: -----

Group Members:

Full Name	Signature
Damoze Bane	-----
Abebe Birhanu	-----
Beka Ahmed	-----

APPROVAL FORM

This is to confirm that the project report entitled **School Management System for Aba Fransua Preparatory School** submitted to **Wolkite University, College of Computing and Informatics Department of Information Technology** by Damoze Bane, Abebe Birhanu and Beka Ahmed. This senior group project have been approved with the signature of advisor and Department of Information Technology.

Advisor Name	Signature	Date
Department Head Name	Signature	Date
Examiner 1 Name	Signature	Date
Examiner 2 Name	Signature	Date
Examiner 3 Name	Signature	Date

ACKNOWLEDGEMENT

First, our gratitude thanks to God for help us in this project starting from the beginning up to now. Secondly, we would like to express our deepest gratitude to our Advisor Mr. Abraham Assefa for his advice and constrictive commit in our project and we would like to thanks Computing and Informatics College for giving us this chance to do this final project. We would also like to thank to Aba Fransua Preparatory School for their willingness to interview and give us different forms and documents that help us to the achievement of our proposed project.

Table of Contents	Pages
DECLARATION	ii
APPROVAL FORM	iii
ACKNOWLEDGEMENT	iv
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF ABBREVIATION	xii
ABSTRACT.....	xiii
CHAPTER ONE	1
1. INTRODUCTION	1
1.1. Background of the Organization	1
1.2. Statement of the Problem.....	2
1.3. Objective of the Project	2
1.3.1. General Objectives.....	2
1.3.2. Specific Objectives	2
1.4. Feasibility of the Project	3
1.4.1. Technical Feasibility.....	3
1. 4.2. Operational Feasibility.....	3
1. 4.3. Economic Feasibility	3
1.5. Scope and Limitations of the Project	4
1.5.1. Scope of the project	4
1.5.2. Limitations of the Project.....	4
1.6. Significance & Beneficiaries of the Project.....	4
1. 6.1. Significance of the project	4
1.6.2. Beneficiaries of the Project.....	5

1.7. Functional and nonfunctional requirements.....	5
1.7.1. Functional requirements.....	5
1.7.2 Nonfunctional Requirements	5
1.8. Methodology	6
1.8.1. Data collection methodology	6
1.8.2. System analysis and design Methodology	6
1.8.3. System Development Models	7
1.9. Programming Development Languages and Tools.....	8
CHAPTER TWO	10
2. DESCRIPTION OF THE EXISTING SYSTEM	10
2.1. Introduction of the Existing System	10
2.2. Users of the Existing System	10
2.3. Major Function of the Existing System	10
2.5. Drawbacks of the Existing System	13
2.6. Business Rule of the Existing System	14
CHAPTER THREE	15
3. PROPOSED SYSTEM	15
3.1. Function Requirements	15
3.2. Nonfunctional Requirements	16
3.2.1. User interface and human factors	16
3.2.2. Hardware Consideration	16
3. 2.3. Security Issues	17
3. 2.4. Performance Consideration.....	17
3. 2.5. Error handling and Validation	17
3.2.6. Quality Issues.....	17

3.2.7. Backup and Recovery	18
3.2.8. Physical Environment	18
3.2.9. Resource Issues	18
3.2.10. Documentation	18
CHAPTER FOUR.....	19
4. SYSTEM ANALYSIS	19
4.1. System Model	19
4.1.1. Use case Model	19
4.2. Object model	29
4.2.1. Class Diagram	30
4.2.2 Data Dictionary	31
4.3. Dynamic Model	34
4.3.2. Activity Diagram	38
4.3.3 State Chart Diagram.....	45
CHAPTER FIVE	47
5. SYSTEM DESIGN	47
5.1 Design Goal	47
5.2 Proposed System Architecture	49
5.2.1 Subsystem Decomposition and Description	50
5.2.2 Hardware and software Mapping.....	52
5.2.3 Detailed Class Diagram	53
5.2.4 Persistent Data Management.....	54
5.2.5 Access control and Security	55
5.3 Package Design.....	56
5.4. Algorithm Design.....	57

5.5 User Interface Design	57
6. IMPLEMENTATION AND TESTING	60
6.1. Implementation of Database	60
6.1.1 Implementation of Class Diagram	63
6.1.2 Configuration of Application server	64
6.1.3 Configuration of Application Security.....	64
6.1.4 Implementation of User Interface	64
6.2 Testing.....	66
6.2.1 Testing tools and Environments.....	66
6.2.2 Unit Testing	67
6.2.3 System Testing.....	67
6.2.4 Integration Testing	68
6.2.5 Acceptance Testing	68
CHAPTER SEVEN	69
7. CONCLUSION AND RECOMMENDATION.....	69
7.1. Conclusion	69
7.2. Recommendation	69
REFERENCE.....	70

LIST OF TABLES

Table 3.1 Hardware Consideration	16
Table 4.2 Actors and use case table	20
Table 4.3 Use case description of login	23
Table 4.4 Use case description of create account	24
Table 4.5 Use case description of post information.....	25
Table 4.6 Use case description of insert mark	26
Table 4.7 Use case description of registration	27
Table 4.8 Use case description of view transcript	28
Table 4.9 Data dictionary for Administrator table.....	32
Table 4.10 Data dictionary for User account	32
Table 4.11 Data Dictionary for Teacher table	33
Table 4.12 Data Dictionary for Record officer table	33
Table 4.13 Data Dictionary for Student table	34
Table 5.14 Access control privileges in our System.....	55

LIST OF FIGURES

Figure 2.1 Student registration form	12
Figure 2.2 Student transcript sheet.....	12
Figure 2.3 Student attendance form	13
Figure 4.4 Use case Diagram	22
Figure 4.5 Class Diagram	31
Figure 4.6 Sequence diagram for Login Use case	35
Figure 4.7 Sequence Diagram for Create account use case	35
Figure 4.8 Sequence Diagram for registration Student Use case.....	36
Figure 4.9 Sequence Diagram for Post Information Use case	36
Figure 4.10 Sequence Diagram for view Transcript Use case	37
Figure 4.11 Sequence Diagram for Prepare Transcript Use case	37
Figure 4. 12 Sequence Diagram for Insert mark Use case.....	38
Figure 4.13 Activity diagram for Login Use case.....	39
Figure 4.14 Activity diagram for Post Information Use case	40
Figure 4.15 Activity diagram for registration Use case	41
Figure 4.16 Activity diagram for View Transcript Use case	42
Figure 4.17 Activity diagram for Insert Mark Use case	43
Figure 4.18 Activity diagram for Prepare Transcript Use case	44
Figure 4.19 State chart for Login	45
Figure 4.20 State chart for create account	45
Figure 4.21 State chart for deactivate account.....	46
Figure 5.22 Proposed system architecture	50
Figure 5.23 Subsystem Diagram.....	52
Figure 5.24 Deployment Diagram	53

Figure 5.25 Detailed class diagram.....	54
Figure 5.26 Persistence data management	55
Figure 5.27 Package diagram.....	56
Figure 5.28 Home page user interface	58
Figure 5.29 Login user interface	59
Figure 6. 30 Home page.....	65
Figure 6.31 Administrator Login page.....	65
Figure 6.32 Student Login page.....	66
Figure 6.33 registration with correct information.....	67
Figure 6.34 registration without correct information.....	67

LIST OF ABBREVIATION

AFP	Aba fransua Preparatory
BR	Business Rule
CPU	Central Processing Unit
CSS	Cascading Style Sheets
GB	Gigabyte
HTML	Hypertext Markup Language
MB	Megabyte
MySQL	My Structured Query Language
PHP	Hypertext Preprocessor
RAM	Random Access Memory
SDLC	Software Development Life Cycle
SMS	Short Message Service
SQL	Structured Query Language
UML	Unified Modeling Language
USB	Universal Serial Bus

ABSTRACT

School management system in developing countries lags seriously behind as we compared to the developed countries and existing school management system is not enough to support school function. This project emphasis on School management System for Aba Fransua Preparatory School. In this School many problems occurred. Like oldest documents decayed, and fade away, workload on workers, lack of data security, searching data is difficult, take more time, and need many workers. Because of this problem, we proposed web based student's information management system for Aba fransua preparatory school using interview, observation and document analysis for data collection and finding facts. The propose system is developed by using PHP language and very easy to operate and easily handle all the data and the work done of the school, and provide an efficient management and reduce the system complexity and results an easy implementation. This proposed system improves the task load of the school, minimize time and budget, easy way searching data and giving fast service for students and academic staffs of the school.

CHAPTER ONE

1. INTRODUCTION

Nowadays education plays a great role in development of any country. Many of education organizations try to increase education quality. One of the aspects of this improvement is managing of school resources.

Our system is a major tool through this managing process by making a continuous communication between director, teacher, and student. So in order to achieve that goal, we need a Web based student information management system that covers the needs of all users at the same time. For Students, they can view their subject's grades, contact with the director and teachers for any complaint, recommendation or absence permission, and they up to date with all school's news or posts that publish by the other users. For director, they have a full control on the system, as they can add a new teachers and students with their subjects. For Teachers, they can add student's grades or edit it for their own subjects only, and they have a direct connection with students All the system users can publish whatever they need within the educational process on the last updates section, and these posts is visible for all the users.

Now in Aba fransua preparatory school, these tasks done by mostly practiced manually, this manual processing has many problems and their own disadvantages. In order to facilitate and simplify these actions one of the major tools is to have a web based student information management system.

1.1. Background of the Organization

Aba fransua General Secondary & preparatory school was instantly established by local community participation and contribution in 1994 E.C. At first time, it takes only 120 students for grade 9, and in 1997, E.C. gives national university entrance examination. Today the school accepts from 9 to 12 grade students, and numbers of students in preparatory school are: - 510 grade 11 and 380 grade 12 totally 890 students are there. Aba fransua General Secondary & preparatory school found in Gubre town beyond commercial bank of Ethiopia.

Through its learning and teaching journey, about fourteen batches were taken grade twelve university entrance examination and much of them join universities. In our observation, the

teachers are more experienced. Student's information management system in the government schools of Aba fransua preparatory school is manual and the record officers generate transcripts manually.

1.2. Statement of the Problem

Aba fransua preparatory school facility several problems because of the manual system and they keeping the data by using excel. These problems are:

- To searching, the documents in the data storeroom are difficult and tedious.
- The documents, which are oldest, are decayed and fade away.
- Data security is weak.
- Increasing workload by data manipulators and forced to work over time.
- Takes many workers.
- Students cannot register on line and get information about their school at their school.

1.3. Objective of the Project

1.3.1. General Objectives

The main objective of this project is to develop web based student information management system for Aba fransua preparatory school.

1.3.2. Specific Objectives

To achieve the general objective of this project the proposed system consists of the following specific objectives

- To understand the existing system.
- To identify and gather requirement of the existing system.
- To analyses and specify the requirement of the existing system.
- Design and model of the proposed system
- To implement the proposed system
- To test the proposed system.

1.4. Feasibility of the Project

Feasibility study is used to investigate the proposed system in multiple dimensions. It used to indicate whether the system feasible or not. The proposed system can be seen according to the following literals.

1.4.1. Technical Feasibility

Technical feasibility is the measure of practicality of the specific technical solution and the availability of technical resources and expertise. This project can be easily maintaining and repair without requiring high experts or technical assistants; because the system is will be developed by familiar programming language. The technology that we were used to designing and implementing are currently available and we can get them easily, so our project is technically feasible.

1. 4.2. Operational Feasibility

This system brings better achievement for the operations performed by the school by providing efficient registration and storage of students information, easily updating, deletion, modification etc. This increases the efficiency of work in the school, our project is operationally feasible.

1. 4.3. Economic Feasibility

Our project is economically feasible to the organization because the project we were developed was economically feasible than the manual system that the organization uses now. It is cost-effective in the sense that it eliminates the paperwork, and the proposed system will solve the business and time problem for the organization.

Tangible Benefits

- Minimize registration cost.
- Minimize demands of paper, inks.
- Easily access student information throughout the school

Intangible Benefits

- Increased flexibility
- Increase speed of activity
- Improves the confidence of the system users.

- Users save their time

1.5. Scope and Limitations of the Project

1.5.1. Scope of the project

This project design, develop and implement web based for Aba Fransua school and mainly focused on the following: -

- The system we proposed is only for preparatory students or grade 11 and 12.
- Support online registration for students.
- School administrator: they have a full control on the system; they can add, update and delete student's data, and teachers with their subjects.
- Students able to view transcript and report online.
- Teacher can able to prepare student report card.
- Record officer can able to prepare student transcript.
- The proposed system use English language.

1.5.2. Limitations of the Project

- The system we proposed is does not include information about employee such as guards and secretary of the school, because of lack of time.
- The system we proposed is does not generate student's status to the family such as notification and SMS due to lack of time.

1.6. Significance & Beneficiaries of the Project

1.6.1. Significance of the project

The development of this project has the following significance

- This system makes the school data management (keeping of data computerized).
- The system is user friendly.
- The system makes easy communication between student and school management.
- Reduce time, cost and redundancy of information or data.
- Improve efficiency, control and security of the existing system

1.6.2. Beneficiaries of the Project

The beneficial of this system are -

- **School:** This system in the school decreases the amount of paper, pen, certificate and printing cost.
- **Teachers:** they can add student's grades or edit it for their own subjects only, and they have a direct connection with students.
- **School administrator:** they have a full control on the system; they can add a new teachers and students with their subjects.
- **Students:** they can view their subject's grade; get information about their school and can access their profile in easy and simple way.

1.7. Functional and nonfunctional requirements

1.7.1. Functional requirements

- Enable record officers, and students to view student transcript online.
- Enable students, and teachers to view student comment.
- Displays new posted information.
- Enable record officer search the student file from the database
- Allow record officer to registration student.
- Record officer to prepare transcript.
- The administrator enables to post new information.
- Modifying the data

1.7.2 Nonfunctional Requirements

This project includes the following nonfunctional requirements

Efficient

- Searching a student history should not take more time
- The system should be user friendly.

Security

- Administrator, Record officer and Teachers will have its own username and password but not parent and student. Only use student ID.

- Central server was placed in the administrator office secured with key, user name and password.

Maintenance: The system should be easy to maintain and update.

Performance: The system should meet the following performance criteria's

Response Time: - Upon request for user inquiry, the system under normal condition will display results as quickly as possible.

Concurrent Processing: - the system will support multiple users at a time.

Availability: All data in the system will be available all the time.

1.8. Methodology

1.8.1. Data collection methodology

There are many techniques of data collection. Among them, we used the following methods for collecting data.

Interviewing: to get the basic and background information about the existing system we were interview school administrator especially record officer and director.

Observation: By visiting, the school staffs while they are working, for our project we observed the existing system, the environment of classrooms and the procedures of teacher student relationships throughout teaching learning processes that will provide primary information for our project.

Document analysis: from existing document of the school collected more information about the existing system relevant to develop the propose system.

1.8.2. System analysis and design Methodology

Among the different methodologies available, we will use object-oriented system analysis and design methodology for proposed project system development, because it has better advantages over structural system analysis and design methodology.

Some benefits of using the object-oriented paradigm as follows:

- **System stability**-object-oriented techniques tend to produce systems that are resilient to change, where changes can be made without major disruption, with minimal time and effort, and with little fear of disturbing something else in the system.

- **Maintainability**-object-oriented methods produce systems that can be maintain and enhanced more readily.
- **Reality-based systems**– the techniques used with object-oriented methods give a far more accurate picture of the users' business operation and its information needs, leading to a final system that is closer to what the customer wants.
- **Data accessibility**–object-oriented methods tend to lead to better design of databases; this data is more accessible and usable.

1.8.3. System Development Models

The SDLC gives structure to the challenge of transitioning from the beginning to the end of your project without forgetting a step. We are going to use the Agile model which quickly delivers a working product and is considered a very realistic development approach. It produces ongoing releases, each with small, incremental changes from the previous release. Emphasize interaction as the customers, developers, and testers work together via the project. It can adapt well to the changes that occur via the life of the project. It is also communicative, means welcoming feedback from the end-user, and encourages changing requirements as needed.

1.8.4. System Testing Methodology

This methodology is used to test a particular product to ensure that the product fits its purpose. It involves testing that the product works following its specification, has no undesirable side effects when used in ways outside of its design parameters. Ensure that the developed software successfully operates in all the anticipated environments with the required usability and security. Consider both functional Testing and non-functional Testing, functional testing is to ensure that all requirements should be met and the proper working of all the functionalities of an application. In addition, Non-functional is to ensure security and establish the reliability of the application and the application developed is robust enough to work.

Unit Testing: -

- Unit testing is the testing of individual software modules/components that make up an application/system.
- To ensure that all code meets quality standards before it is deploy.

- Save time and money.
- Check how the sub procedures or functions are called correctly.

Integration Testing: -

- Integration Testing is the testing of the different modules/components that have been successfully unit tested when integrated to perform specific tasks and activities
- Create automated tests for specific integrated components.
- Examine how different procedures work together to achieve the goal of individual system

System Testing: -

- Testing the entire system for errors and bugs.
- Examine how the individual components of AMS by face recognition work together to achieve the desired goal.

Performance Testing: - measuring how a system behaves under an increasing load.

Compatibility Testing: -tests that the application is compatible with all the specified operating systems, hardware platforms, web browsers, mobile devices, and other designed third-party programs.

1.9. Programming Development Languages and Tools

Programming language

We use the following languages:

PHP language because the following

- It is easy to learn and runs efficiently on the server side.
- It is a powerful tool for making dynamic and interactive Web pages.
- It supports a wide range of databases
- It is free (open source).

HTML: for display contents

JAVASCRIPT: -for form validation

MySQL: -use for the following reason

- To creating and manipulating databases.
- Easy to use, open Source, it is fast and secure.
- It runs on many operating systems.

UML: - helps to develop use case diagram

Hardware tools

We use desktop computer, USB or flash disk, printer etc.

Software tools; -we use the following software

Microsoft Office

- XAMPP control and WAMP Server
- Adobe Photoshop CS5 for edit image
- Text editor Note pad++, Sublime Text: for writing PHP implementation code
- Macro media Dreamweaver

Modeling tools

EdrawMax or star UML used for designing different models in our project.

CHAPTER TWO

2. DESCRIPTION OF THE EXISTING SYSTEM

2.1. Introduction of the Existing System

School management system on Aba Fransua preparatory school focuses on registering students, registering teachers, managing students and teacher's data, exchanging of reports between the offices of the school. Currently the Aba Fransua Preparatory School performs these activities manually. Manual system has different draw backs like need of a lot of paper every day, lack of space to store these old papers, the service is time taking, searching is not easy and retrieving student information is difficult. These and other related problems can be solved by the proposed system.

2.2. Users of the Existing System

The major users and their responsibilities in the existing system are the following: -

School Administrator: - can post school information and can create user account.

Teacher: - evaluate and who can process students mark, prepare report card, give comment to students and read students comment.

Student: - can register to the school, read a comment given to him/her, see his /her results.

Record officer: register students and prepare transcript.

2.3. Major Function of the Existing System

The major function of the Aba fransua preparatory school is to satisfy and provide service needed by the Aba fransua preparatory school users. In the existing system, major functions has done manually. The major functions in the existing system are as the following

Registration: Registering students and academic staff.

- They fill the appropriate information on the form and attach his/her photo and they submitted to the record officer.
- The record officer also checks whether they are valid or not.
- If the user fills all the necessary information, the record officer accepts him/her and give identification card to him/her.

Preparing transcripts.

- Teachers submit student's marks to record officers the record officers organize student's mark.
- After organizing students mark, the record officers prepare transcripts.
- Students take transcript and checks whether it is correct or not, if it is not correct students re submit to record officers.
- The record officers see and correct it and re give to the students.

Preparing students grade.

- Teachers give exam to students.
- Teachers collect and organize students mark.
- Students take mark from their class teachers.
- If there are any correction students, ask their teachers again to correct their mark.

Preparing reports

- School administrator prepare reports for teacher if there is a problem and do not attend the class.
- Teacher prepare report for student if the student does not attend the school.

Post news: Administrator post information if there are new ideas about teaching and learning process

2.4. Forms and Other Documents of the Existing System

The image shows a handwritten student registration form in Amharic. The form is organized into several columns. The first column is for the student's name. The second column is for the student's ID number. The third column is for the student's sex and age. The fourth column is for the student's grade and section. The fifth column is for the student's date of birth. The sixth column is for the student's date of registration. The seventh column is for the student's date of admission. The eighth column is for the student's date of graduation. The ninth column is for the student's date of completion. The tenth column is for the student's date of enrollment. The eleventh column is for the student's date of registration. The twelfth column is for the student's date of admission. The thirteenth column is for the student's date of graduation. The fourteenth column is for the student's date of completion. The fifteenth column is for the student's date of enrollment. The sixteenth column is for the student's date of registration. The seventeenth column is for the student's date of admission. The eighteenth column is for the student's date of graduation. The nineteenth column is for the student's date of completion. The twentieth column is for the student's date of enrollment.

Figure 2.1 Student registration form

subject											
Grade 12 th											
Abafransua Sec & preparatory school Mark Transf. Sheet 2013 E.C											
No	Full name	Sex	Sec	1st semester			2nd semester			Ave.	R.mark
				60%	40%	100%	60%	40%	100%		
1	Abdillewis Zeinu	M	A								
2	Abulfeta Mekki Ahmed	M	A								
3	Abdourahmou Hayder	M	A								
4	ABESE YIRGA NEDA	M	A								
5	ABRIHAM BELACHEWU	M	A								
6	ABRIHAM TESEMA MNUTA	M	A								
7	Amir Hadesa Hassen	M	A								
8	AMERA MEKA AHMED	F	A								
9	Behru Mohammed	M	A								
10	BELETECH LEGISTU ARGAW	F	A								
11	BEZAWIT FIKADU ANSA	F	A								
12	BIRUK DEMIS ZEPKE	M	A								
13	BIRUK TEGEMET TEBE	M	A								
14	Dagnachewu Mesale	M	A								
15	DAWIT GERRU WOLDE	M	A								
16	EDEN MULUGETA BOGALE	F	A								
17	Efrem Assela	M	A								
18	IKRAM ABDREMED TAMER	F	A								
19	Feysof Abdirahman	M	A								
20	GEDION GEBRE BERGA	M	A								
21	HABTAMU FIKRE	M	A								
22	HALID MEHDI YASIN	M	A								
23	Hanna Hassen	F	A								
24	Hambya Hussein	F	A								
25	Hassan Bebru	M	A								

Figure 2.2 Student transcript sheet

		Abafransua General Sec & preparatory school														12								
		Attendance																						
S.No.	Student Name	SEX	S/OC	M	T	W	TH	F	S	M	T	W	TH	F	S	M	T	W	TH	F	S	Total	Absent	
1	Abdilkerim Zeinu	M	A																					
2	Abdifeta Meki Ahmed	M	A																					
3	Abduruaf Hayder	M	A																					
4	ABEBE YIRGA NEDA	M	A																					
5	ABRRHAM BELACHEWU	M	A																					
6	ABRRHAM TESEMA MNUTA	M	A																					
7	Amir Hadra Nasir	M	A																					
8	AMIRA MEKA AHMED	F	A																					
9	Behru Mohammed	M	A																					
10	BELETECH TIGISTU ARGAW	F	A																					
11	BEZAWIT FIKADU ANSA	F	A																					
12	BIRUK DEMIS ZEPRE	M	A																					
13	BIRUK TEWAGI NEMA	M	A																					
14	Dagnachewu Mesele	M	A																					
15	DAWIT GEBRU WOLDE	M	A																					
16	EDEN MULUGETA BOGALE	F	A																					
17	Efram Assefa	M	A																					
18	EKRAM ABRSEMED UMER	F	A																					
19	Feysel Abdirahman	M	A																					
20	GEDION GEBRE BERGA	M	A																					
21	HABTAMU FIKRE	M	A																					
22	HALID MEHDI YASIN	M	A																					
23	Halima Hassen	F	A																					
24	Hamdya Hussien	F	A																					
25	Hamid Nesiru	M	A																					

Figure 2.3 Student attendance form

2.5. Drawbacks of the Existing System

The existing system has the following problems: -

Performance: - It is time consuming, and less performance since every task is done manually and it requires a long period to complete the task.

Security and control: - Every record of data in the existing system is stored the manual way, so, it is difficult to control and secure these manual records, since it does not have any authentication and authorization system.

Efficiency: - It refers perform the task using little time and resource, but the current system requires much resources like paper and time since the existing system is inefficient.

Data storage problem

- Data are not easily accessible due to place in different location.
- Difficult to change and edit.
- Data redundancy that leads to inconsistency

Redundancy: - Data is redundant because data is stored in paper and MS Excel the data may be record or store many times. If someone want to find one's data, it is difficult to capture the correct data in short period of time because the data is stored manually.

2.6. Business Rule of the Existing System

The business rule that are involved in current System are -

BR1: Record officers registered student's data.

BR2: Student must finish grade 10 and pass national exam

BR3: Student must have average above 50% and if all subjects 50% and above, Student must have average above 52% and if one subjects below 50%, Student must have average above 53% and if two subjects below 50%, Student must have average above 54% and if three subjects below 50%, if four and above subjects below 50% cannot register

BR4: Students fill the registration form.

BR5: Each student must wear a uniform clothe when come to the school.

BR6: Each student must have identification card

BR7: If the student lost the school, educational material will pay.

BR8: Student assigned to classroom for their teachers

BR9: Teachers register with their subject

CHAPTER THREE

3. PROPOSED SYSTEM

The proposed system is a Web-Based system that was developed by using PHP language. The static page is designed by using HTML for the layout of the web page, JavaScript for validation and CSS for more attractive Web site while the server-side programming will be done by using PHP and MYSQL as a database.

The proposed system is very easy to operate and easily handle all the data and the work done of the school, and provide an efficient management and reduce the system complexity and results an easy implementation. This system is mainly concerned with how administrator record officers are working their position. This proposed system improves the following: -

- It minimizes the task load of the school.
- Fast and easy registration.
- Creating effective functioning method in managing school.
- Minimizing time and budget.
- Easy way in searching, viewing, updates transcript.
- Teachers can easily prepare students mark.
- Giving fast service to the students and academic staffs of the school
- Easily creating account for teachers and record officers.

3.1. Function Requirements

The functional requirements that are included upon the system by rendering from the user's requirements. This includes operations performed upon the system by using the system going to be designed. The new system performs tasks and duties done by the current system without changing the business rule of the existing system and in a cost-effective way. The new system is used to manage all activities of the Aba Fransua Preparatory School.

The following are functional requirements expected from the new system.

- The systems admin manages user account.
- The system should allow deactivating users to account.
- The systems manage, teachers and record officer and student's information
- The system should allow updating, view any information.

- Enable record officers, and students to view student transcript online.
- Enable students, and teachers to view student comment.
- Enable record officer search the student file from the database
- Allow record officer to registration student.
- Record officer prepare transcript.
- The system assigns teachers and students section.
- The system manages posts and deletes news, announcement.
- The system should allow seeing the rule and regulation of the school.

3.2. Nonfunctional Requirements

Nonfunctional requirements are not directly concerned with the specific service delivered to by the system to its users. It describes aspects of the system that are concerned with how the system provides the functional requirements. Non-functional requirement support and give more quality for the system. These are: -

3.2.1. User interface and human factors

The user interface of our system should have a user-familiar and attractive interface so the system should easy and understandable for every user. To increase understandably of the system, we will use bootstrap (icon), button, text field, and preferable and attractive color user interface.

3.2.2. Hardware Consideration

The hardware that required developing the system is the following:

Table 3.1 Hardware Consideration

Number	Items	Specification
1	CPU	32-bit Core i3(TM) 3.30GHz
2	RAM	Minimum of 2GB
3	HARD DISK	Minimum of 22GB and others
4	FLASH DISK	8GB

5	CD	700MB
---	----	-------

3. 2.3. Security Issues

Our system is protected against internal and external intrusions or against an authorized user by using the following security controls.

User authentication: - The system should authenticate the users when they try to access by asking user name and passwords.

The system should keep the personal information protection guideline during authentication.

User accessibility control: - User can access only their privilege concern areas of the system components.

3. 2.4. Performance Consideration

Response Time: - Upon request for user inquiry, the system under normal condition will display results as quickly as possible.

Processing Time: - Since the system will be, develop with efficient programming language and database upon request for user's Activities, the system under normal condition will process the request as quickly as possible by using multi-tier architectures.

Scalability: - the system will support multiple users at a time.

3. 2.5. Error handling and Validation

Our system has some rules (validation issue) that means server with PHP and client side with JavaScript validation and error handling methods to reduce input fault, the system will allow the user interface generate Error message should contain necessary information about the error. The error message is be concise and a short description that is meaningful for user and gives a clear idea of the problem and how to resolve it. Enable the user to confirm before creation, deletion or modification occurs. Respond to error inputs by asking the user to re-enter data in the correct format. Therefore, our system has high levels of error correction and input validation by using database constraint and datatypes that entered to the system as input.

3.2.6. Quality Issues

Availability: - Our system will be available for 24 hours, if intranet network available.

Usability: - concerned with the following

User operability: -The system will offer simple navigation any user with basic computer knowledge so, can operate function.

Reliability: - The proposed system minimize crash during its runtime, since more than one user could use the system simultaneously.

- Good validations for user input will be done.
- Avoid incorrect storage of records.

3.2.7. Backup and Recovery

The proposed system will have two databases for recovery and backup. One for a copy of the original database used for a backup purpose that is used for recovery if the original database fails.

3.2.8. Physical Environment

The system is deployed or installed on the server computer, but for more feature we recommend that the system to deploy on the cloud that is free from any disaster.

3.2.9. Resource Issues

We develop the system to minimize wastage of resource like reducing human power. In the future, the system use computer, flash, server etc.

3.2.10. Documentation

Since we develop user manual documentation, the user that use the system cannot face any difficulty concerning how to use the system. The user guide will aid users who work on the system to have a detailed and clear understanding of the system steps and functionalities. The developed system has full documentation if some failure occurred could maintain easily.

CHAPTER FOUR

4. SYSTEM ANALYSIS

4.1. System Model

System model is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system. System modelling has now come to mean representing a system using some kind of diagram notation, which is always based on notations in Unified Modeling Language.

4.1.1. Use case Model

Use case model is composed of a use case diagram and the accompanying documentation describing the use cases, actors, and associations. In addition, a use case model is the representation of a system intended functions and its environment.

4.1.1.1. Use case Diagram

Use case diagram define the different types of users of a system and the various ways that they interact with the system. They include actor, use case, system boundary and relationship. Use cases represent system functionality, the requirements of the system from the user's perspective. Actors represent the people or systems that provide or receive information from the system they are among the stakeholders of a system. Use Case diagrams, therefore, show which actors initiate use cases they also illustrate that an actor receives information from a use case. In essence, a Use Case diagram can illustrate the requirements of the system Actor is a person, organization, or external system that plays a role in one or more interactions with the system. The actors and the use case that participate in our project are listed in the following table.

Table 4.2 Actors and use case table

Actor	Use case under identified actors	Use case ID
System Administrator	Login <ul style="list-style-type: none"> ▪ Logout 	01
	Manage system <ul style="list-style-type: none"> ▪ Create user account ▪ Update user account ▪ Delate user account ▪ View user account ▪ Active account ▪ Deactivate account 	02
	Send Report	03
	Send feedback	04
Record officer	Login <ul style="list-style-type: none"> ▪ Logout 	01
	Register Student	05
	Prepare transcript	06
	Check student list	07
	Send report	08
	Send feedback	09
Teacher	Login <ul style="list-style-type: none"> ▪ Logout 	01
	Evaluate student <ul style="list-style-type: none"> ▪ Add student mark 	

	<ul style="list-style-type: none"> ▪ Update student mark ▪ Delete student mark 	10
	Send report	11
	Send feedback	12
Student	Login <ul style="list-style-type: none"> ▪ Logout 	13
	Registered	14
	View mark	15
	View report	16
	View comment	17

Use case:

- **Create account:** - School administer can create account for users.
- **Login:** - administrator, teacher, student & record officer enjoyed the system first login by their username and password
- **Post information:** - Administrator easily to add information, update information, delete information and view information using this system.
- **Register student:** - record officer can easily register parent and students using this system
- **Prepare report card:** - Homeroom teacher prepare report card for student mark.
- **Insert student's mark:** - teacher insert student mark easily by using this system
- **View report card:** - students can easily see their transcript and report card by using this system
- **Prepare transcript:** - record officer can prepare student transcript with this system.

- **Read comment:** - student can able to view comments using this system.

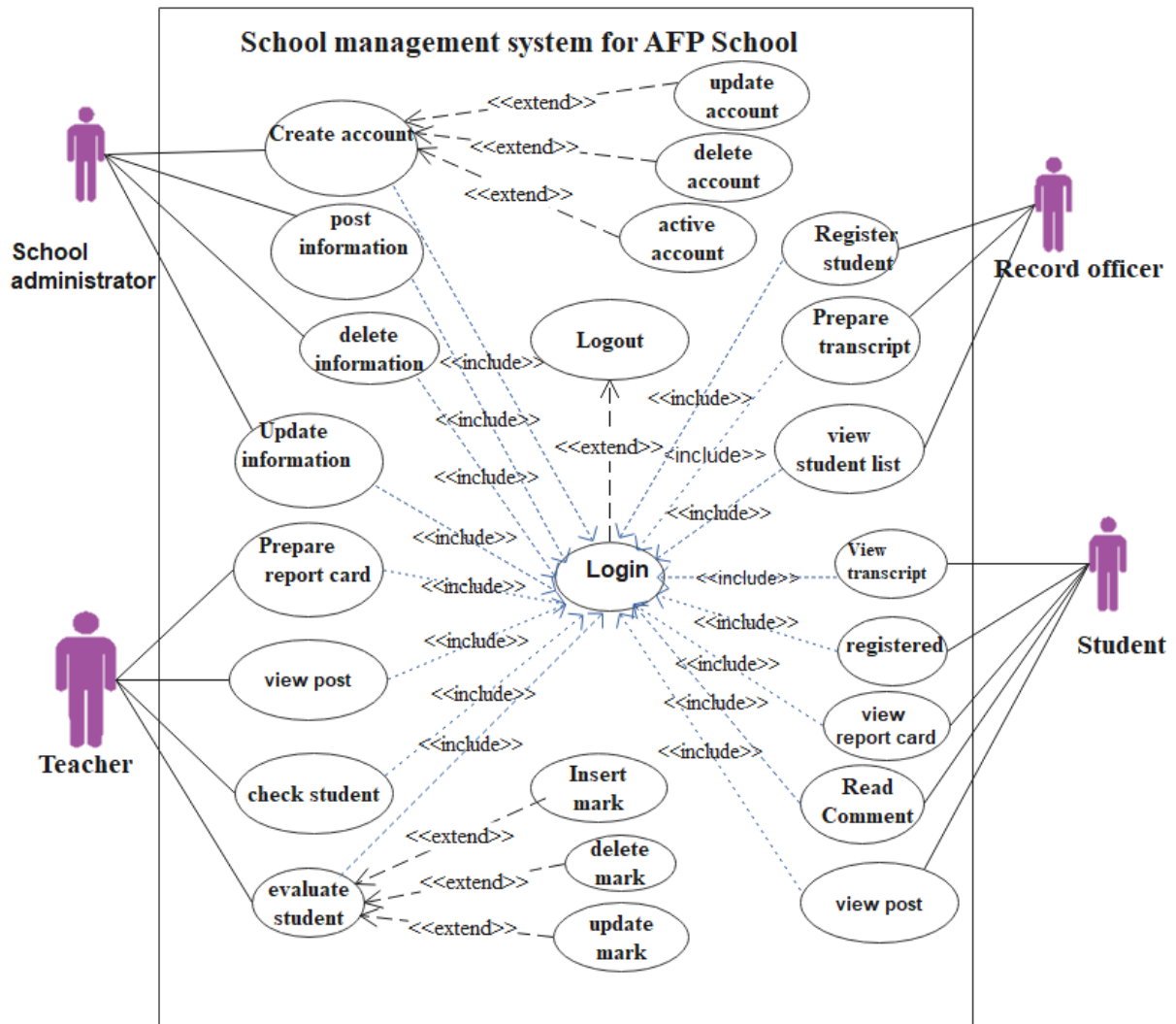


Figure 4.4 Use case Diagram

4.1.1.2 Use case Description

The following tables show use cases description based on the system business or context of our system.

Table 4.3 Use case description of login

Use case name	Login	
Use case identifier	UCID-01	
Actors	Administrator, Teacher, Record officer, Student	
Description	Allows user's login into the system and access their privilege in the system	
Precondition	The users must have username and password (user account).	
Basic course of action	Actors action	System Response
	<p>Step1: Open system webpage</p> <p>Step3: click the login on home page.</p> <p>Step5: enter username and password then click login button.</p>	<p>Step2: displays home page</p> <p>Step4: displays login interface for the user.</p> <p>Step6: validates user name and password. [Alt. Course A]</p> <p>Step7: displays a page with privileges of a user's use case end. [Alt. Course B]</p>
Alternative course of action	<p>Alt. Course A: If the input data (username and password) invalid. StepA.1: The system displays error message and returns to step4.</p> <p>Alt. Course B: if a user wants to sign out from a privilege. StepB.1: a user clicks on logout link then use case end.</p>	
Post condition	The user logged into a system (system display users' privilege) then user sign out after performing their intended actions.	

Table 4.4 Use case description of create account

Use case name	Create Account	
Use case identifier	UCID-02	
Actors	Administrator	
Description	System allows Administrator to create account for the users.	
Precondition	The Administrator is authenticated and login to the system.	
Basic course of action	Actors action	System Response
	<p>Step1: The administrator selects create account link</p> <p>Step3: fills the required information and submits it.</p>	<p>Step2: displays create account page.</p> <p>Step4: validates the information.</p> <p>Step5: Registers the users into the database and display a message successfully registered</p>
Alternative course of action	<p>If the information admin fill is invalid information</p> <p>The system displays error message and Go to step3.</p>	
Post condition	The account is successfully created	

Table 4.5 Use case description of post information

Use case name	Post information	
Use case identifier	UCID-03	
Actors	Administrator	
Description	Add necessary information to the system.	
Precondition	Should have a legal account (username and password).	
Basic course of action	Actors action	System Responses
	<p>Step1: Login to the system by their own username and password.</p> <p>Step4: write information</p> <p>Step6:click logout</p>	<p>Step2: Check that person is legal or not to post information.</p> <p>Step3: displays post information page</p> <p>Step5: display information is posted</p> <p>Step7: use case end</p>
Alternative course of action	<p>If administrator enter wrong username and password.</p> <p>The system displays please enters a correct user name and password and The use case continuous to step2</p>	
Post condition	Administrator successfully post information.	

Table 4.6 Use case description of insert mark

Use case name	Insert mark	
Use case identifier	UCID-4	
Actors	Teacher	
Description	Teacher should has privileges to insert student mark in to the system	
Precondition	Teacher should have a legal user name and password	
Basic course of action	Actors action	System Responses
	<p>Step1: Log into the system</p> <p>Step3: enter username and password</p> <p>Step5: click Insert mark link on the page</p> <p>Step7: enter student ID to the search box</p> <p>Step9: Teacher insert mark</p>	<p>Step2: The system displays home page.</p> <p>Step4: verify username and password</p> <p>Step6: System display search box</p> <p>Step8: check the value is valid or not</p> <p>Step10: display successfully inserted</p>
Alternative course of action	<p>Step4.1: If a teacher entering their incorrect username and password, the system displays please enter a correct user name and password and use case go to step3</p> <p>Step8.1: If a teacher entering incorrect student ID</p> <p>The system display please enter a correct student ID and use case go to step7</p>	
Post condition	Student mark will successfully insert	

Table 4.7 Use case description of registration

Use case name	Registration student	
Use case identifier	UCID-5	
Actors	Record officer	
Description	Record officer should has privileges to register student.	
Precondition	Record officer should have his or her own legal account.	
Basic course of action	Actors action	System Responses
	<p>Step1: open system web page.</p> <p>Step3: enter username and password</p> <p>Step5: fills the required information and submit</p>	<p>Step2: display record officer page.</p> <p>Step4: validate and display student registration form.</p> <p>Step6: validates the information and save to database</p> <p>Step7: registered successfully</p>
Alternative course of action	<p>Step4.1: If a record officer makes a mistake during entering their own user name and password,</p> <p>The system displays please enter correct user name and password and use case go to step3</p> <p>Step.6: fill information is incorrect, the system displays please fill correct information and use case go to step5</p>	
Post condition	Record officer will success to register students using this system.	

Table 4.8 Use case description of view transcript

Use case name	View transcript	
Use case identifier	UCID-06	
Actor	Student	
Description	Student should have privilege to view transcript	
Precondition	Student must have legal account	
Basic course of action	Actors action	System Responses
	Step1: Student want to view transcript Step3: enter username and password Step5: enter student Id Step8:view and logout	Step2: The system displays home page. Step4: verifies the entered username and password and display view transcript link Step6: verifies the entered ID. Step7: display transcript value
Alternative course of action	If the Users made a mistake during entering student ID Step6.1: enter correct ID Step6.2: use case continuous to Step 5	
Post condition	Users will success to view transcript	

4.1.1.3. Use case Scenario

A use case is an abstraction that describes all possible scenarios involving the described functionality. A scenario is an instance of a use case describing a concrete set of actions, when and where they are working, why they are using the system, and what they need the system to do for them. Scenarios used as example for illustrating common use cases. The following are describing scenario of how the user use the system.

Scenario name: Login

Actor: Administrator, Teacher, Student and record officer as user he/she wants to login to the system first open browse the home page then clicks login link from home page. After that the

system display login form then fill login form and click login button if the data is invalid the system display error message else user page is display next perform his authenticated operation and logout.

Scenario name: Create account

Actor: Administrator

Administrators as user she/he wants to create account at first time. First browse home page then clicks manage account link from home page, the system displays the manage account form. After this click create account link and the form becomes display. Then fill correctly and click create account button if the data is not correctly fill the system display error message else create account is success.

Scenario name: Registration

Actor: record officer

Record officer as user he/she wants to register for new Student at first time. First browse home page then clicks student register link from home page, the system display the registration form then fill correctly and click student registration button if the data is invalid the system display error message else student registration is success.

Scenario name: Update Account

Actor: Administrator

Administrator as user she wants to update account the first time. First browse home page then clicks manage account link from home page, the system displays the manage account form. After click on update account button and the form becomes display. Then fill correctly and click update account button if the data is not correctly fill the system display error message else update account is success.

4.2. Object model

Describe the system in terms of object classes and their associations, with common attributes and the services (operations) provided by each object.

4.2.1. Class Diagram

A Class is a category or group of things that has similar attributes and common behavior. A Rectangle is the icon that represents the class it is divided into three areas. The upper most area contains the name, the middle; area contains the attributes and the lowest areas show the operations.

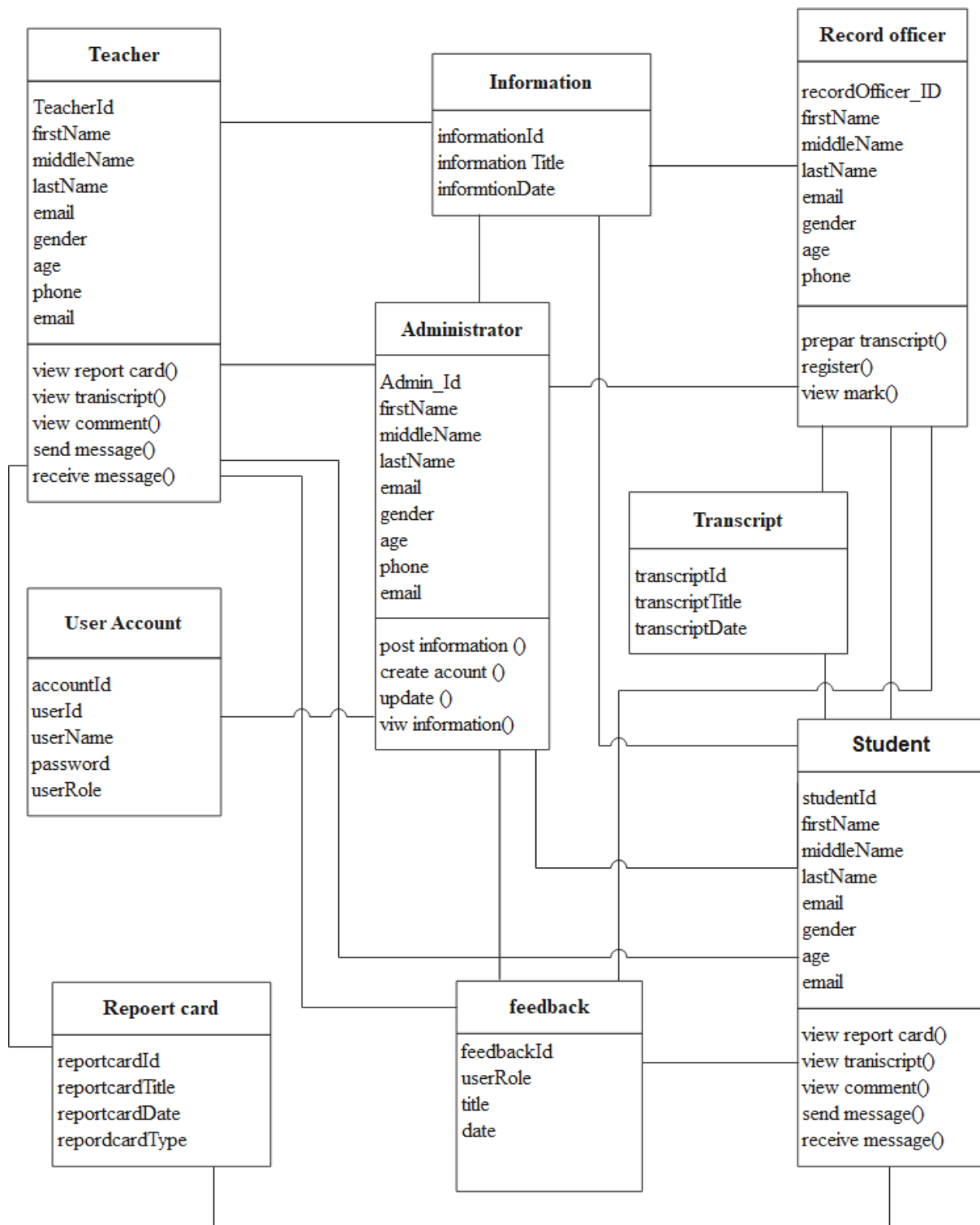


Figure 4.5 Class Diagram

4.2.2 Data Dictionary

Data dictionary is a set of information describing attributes data type, data size, and key constraints of the identified entities.

Table 4.9 Data dictionary for Administrator table

Field Name	Caption	Example	Data Type	Constraint
AdminId	Administrator ID	Admin/123	Varchar(50)	Primary key
firstName	First Name	Abebe	Varchar(50)	Not null
middleName	Middle Name	Biruk	Varchar(50)	Not null
lastName	Last Name	Yimer	Varchar(50)	Not null
Gender	Sex	Male	Varchar(50)	Not null
Age	Age	20	int	Not null
Phone	Phone number	+251917292225	Varchar(50)	Null
Email	E-mail	dam@gmail.com	Varchar(50)	Null

Table 4.10 Data dictionary for User account

Field name	Caption	Example	Data Type	Constraint
accountId	Account ID	acc/01	Varchar(50)	Primary key
userId	User ID	Teach/343	Varchar(50)	Foreign key
userName	User name	Beka Ahmed	Varchar(50)	Not null
Password	Password	Bek123	Varchar(50)	Not null
userRole	User Role	Teacher	Varchar(50)	Not null

Table 4.11 Data Dictionary for Teacher table

Field Name	Caption	Example	Data Type	Constraint
teacherId	Teacher ID	teach/123	Varchar(50)	Primary key
firstName	First Name	Ali	Varchar(50)	Not null
middleName	Middle Name	Biruk	Varchar(50)	Not null
lastName	Last Name	Yimer	Varchar(50)	Not null
Gender	Sex	Male	Varchar(10)	Not null
Age	Age	22	int	Not null
Phone	Phone number	+251917292225	Varchar(50)	Null
Email	E-mail	bek@gmail.com	Varchar(50)	Null

Table 4.12 Data Dictionary for Record officer table

Field Name	Caption	Example	Datatype	Constraint
recordOfficerId	Record officer ID	recoff/123	Varchar(50)	Primary key
firstName	Fist Name	Alem	Varchar(50)	Not null
middleName	Middle Name	Jamal	Varchar(50)	Not null
lastName	Last Name	Yimer	Varchar(50)	Not null
Gender	Sex	Female	Varchar(10)	Not null
Age	Age	19	int	Not null
Phone	Phone number	+251917292276	Varchar(50)	Null

Table 4.13 Data Dictionary for Student table

Field Name	Caption	Example	Data Type	Constraint
studentId	Student ID	stud/123	Varchar(50)	Not null
firstName	First Name	Tesfa	Varchar(50)	Not null
middleName	Middle Name	Alemayehu	Varchar(50)	Not null
lastName	Last Name	Kedir	Varchar(50)	Not null
Gender	Sex	Male	Varchar(10)	Not null
Age	Age	19	int	Not null
phone	Phone number	+251917292345	Varchar(50)	Null
Email	E-mail	tesf@gmail.com	Varchar(50)	Null

4.3. Dynamic Model

The dynamic model represents the time–dependent aspects of a system. It is concerned with the temporal changes in the states of the objects in a system. It includes support for sequence diagrams, activity diagrams and state diagrams.

4.3.1 Sequence Diagram

Sequence diagrams are one of two types of interaction diagrams. They illustrate the objects that participate in a use case and the messages that pass between them over time for one use case. A sequence diagram is a dynamic model that shows the explicit sequence of messages that are passed between objects in a defined interaction. Because sequence diagrams emphasize the time-based ordering of the activity that takes place among a set of objects, they are very helpful for understanding real-time specifications and complex use cases.

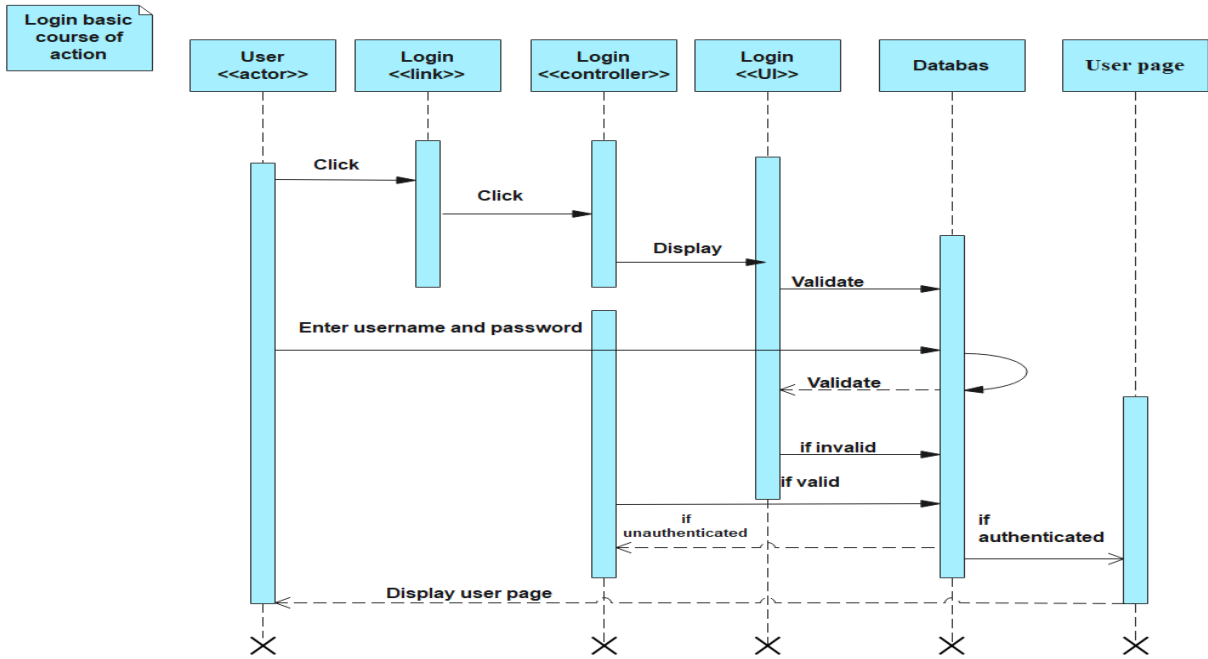


Figure 4.6 Sequence diagram for Login Use case

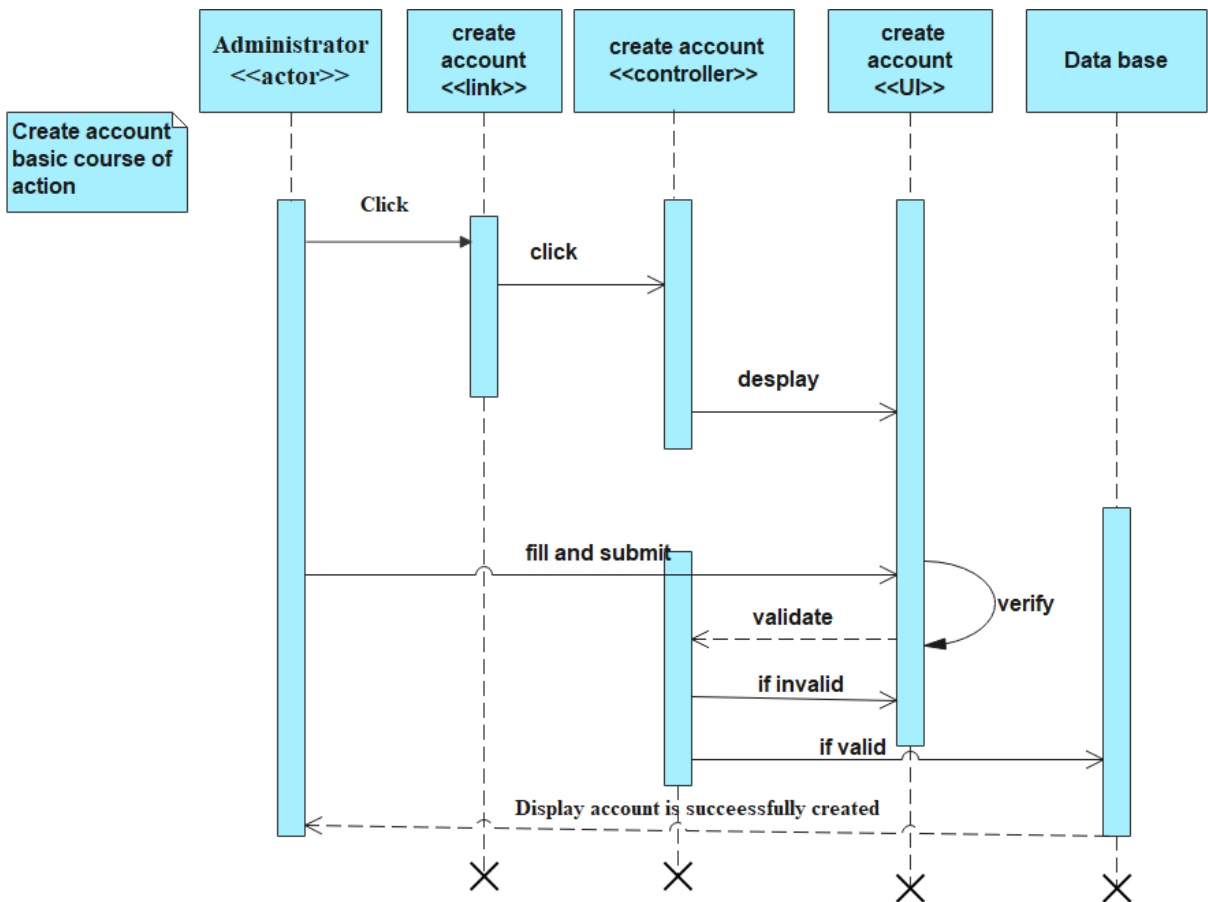


Figure 4.7 Sequence Diagram for Create account use case

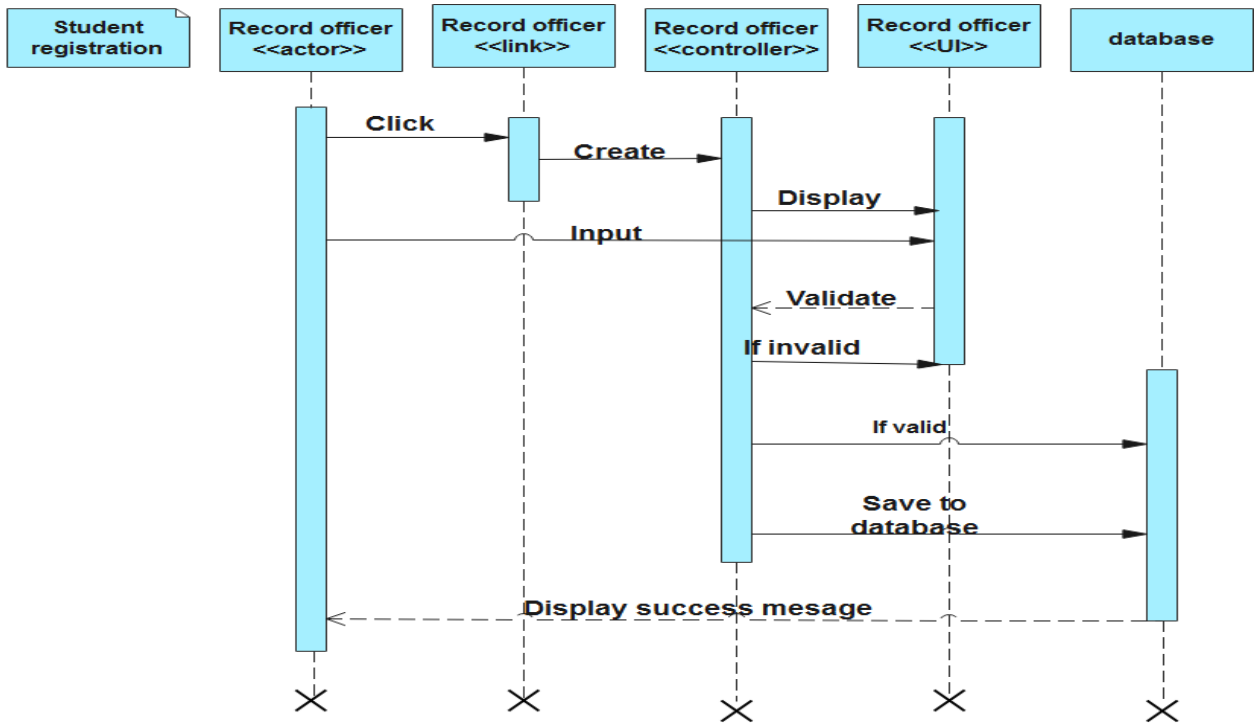


Figure 4.8 Sequence Diagram for registration Student Use case

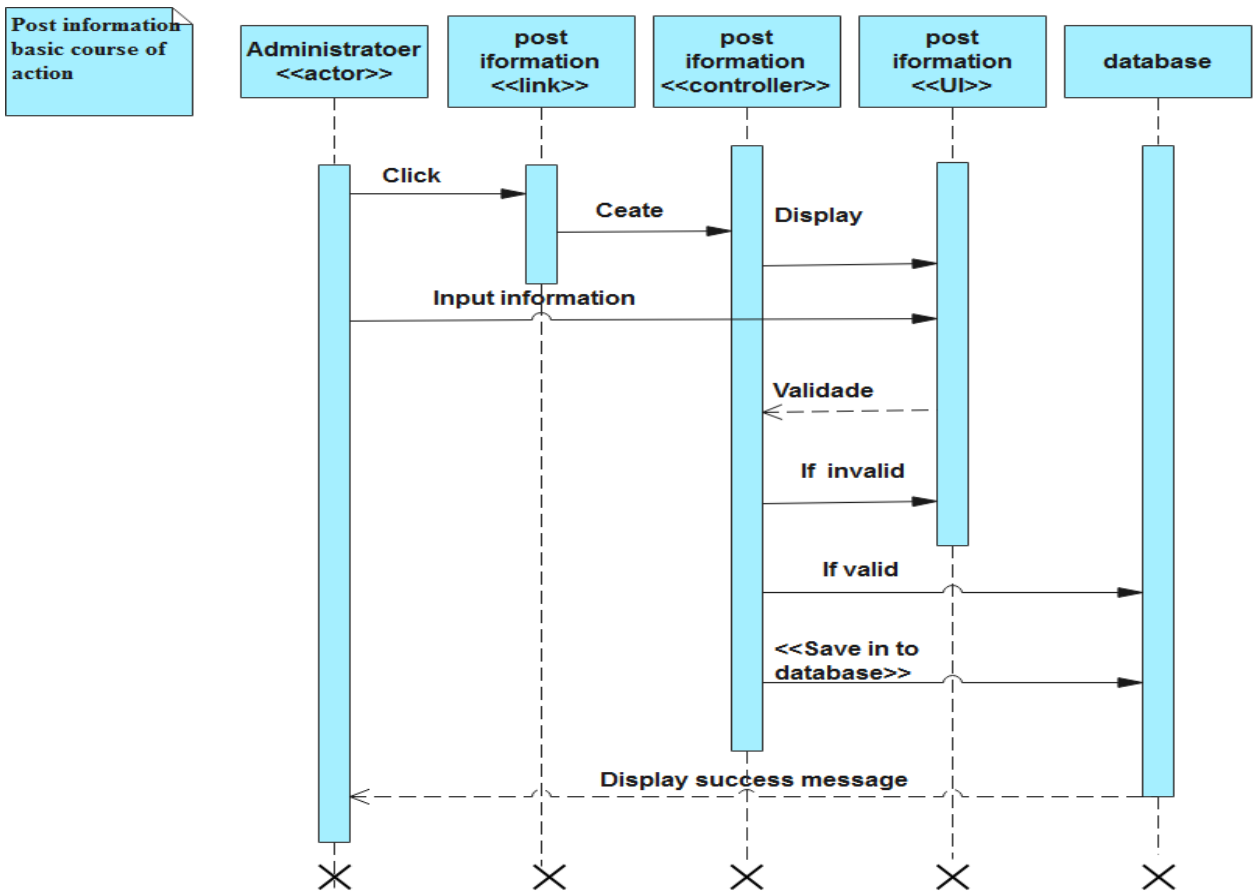


Figure 4.9 Sequence Diagram for Post Information Use case

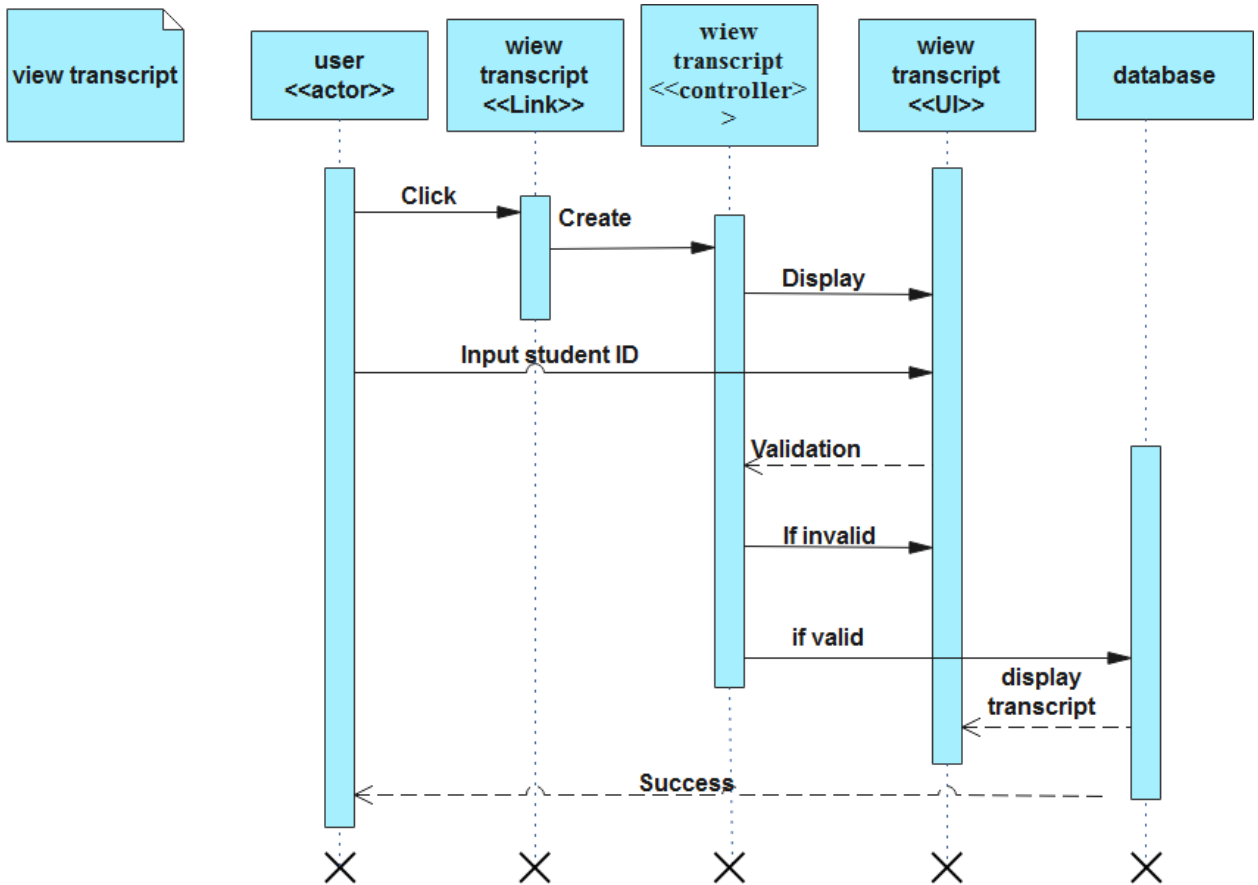


Figure 4.10 Sequence Diagram for view Transcript Use case

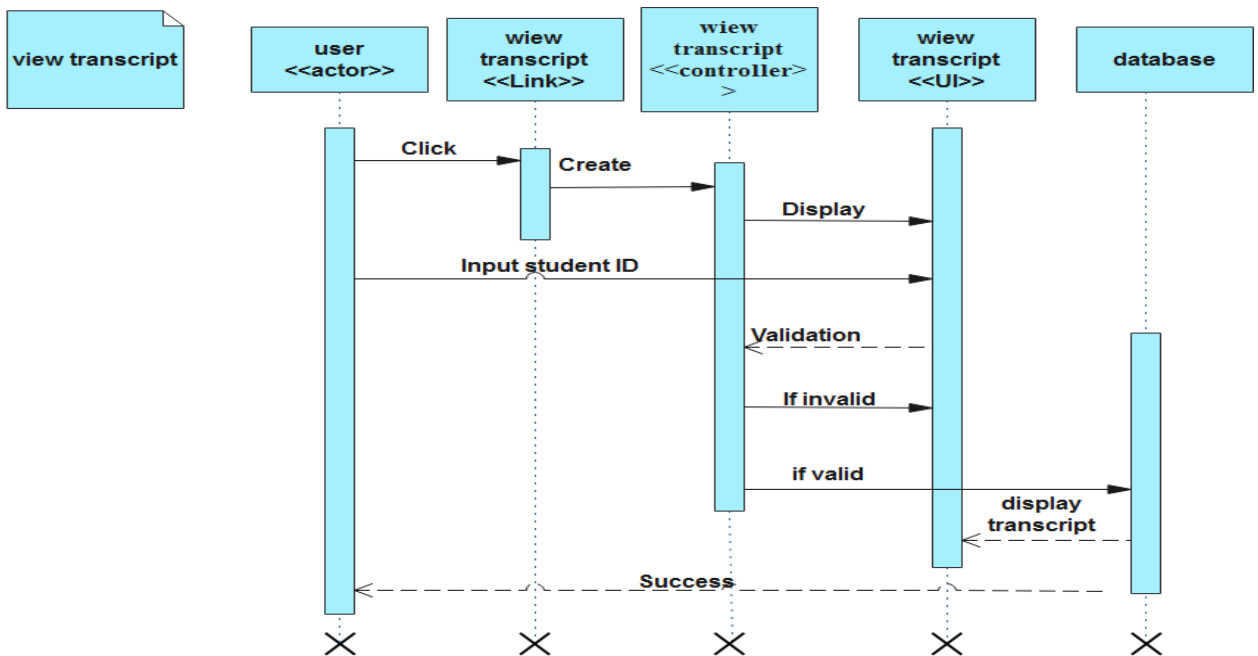


Figure 4.11 Sequence Diagram for Prepare Transcript Use case

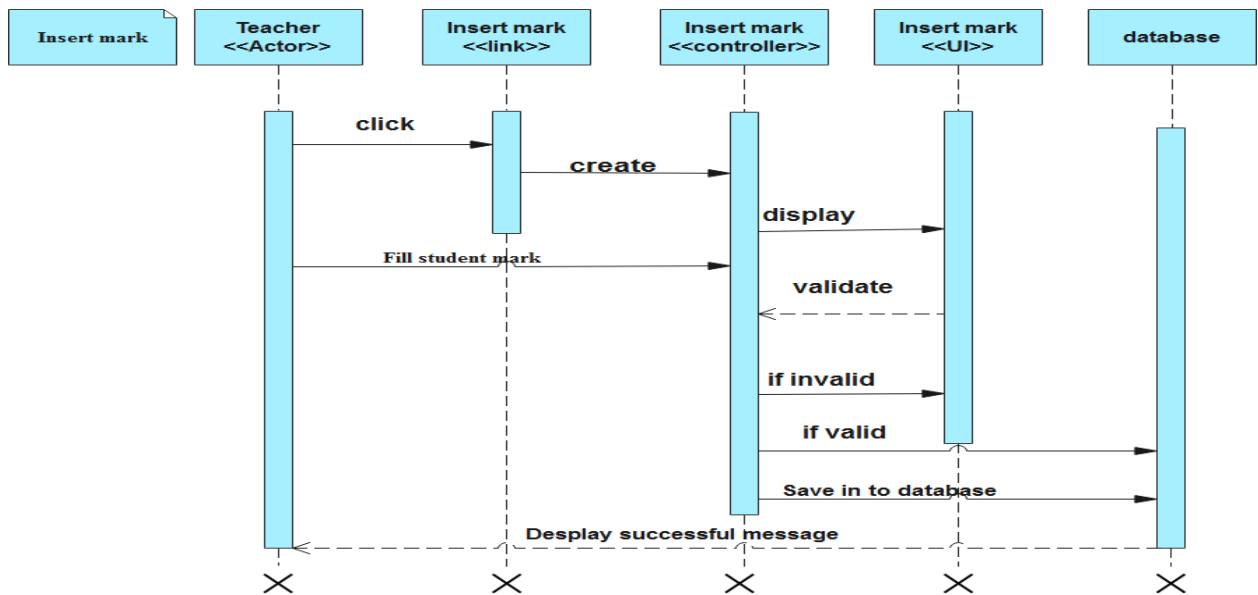


Figure 4. 12 Sequence Diagram for Insert mark Use case

4.3.2. Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. Therefore, the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deal with all types of flow control by using different elements like fork, join etc. The purposes of activity diagram described as: -

- Draw activity of the system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

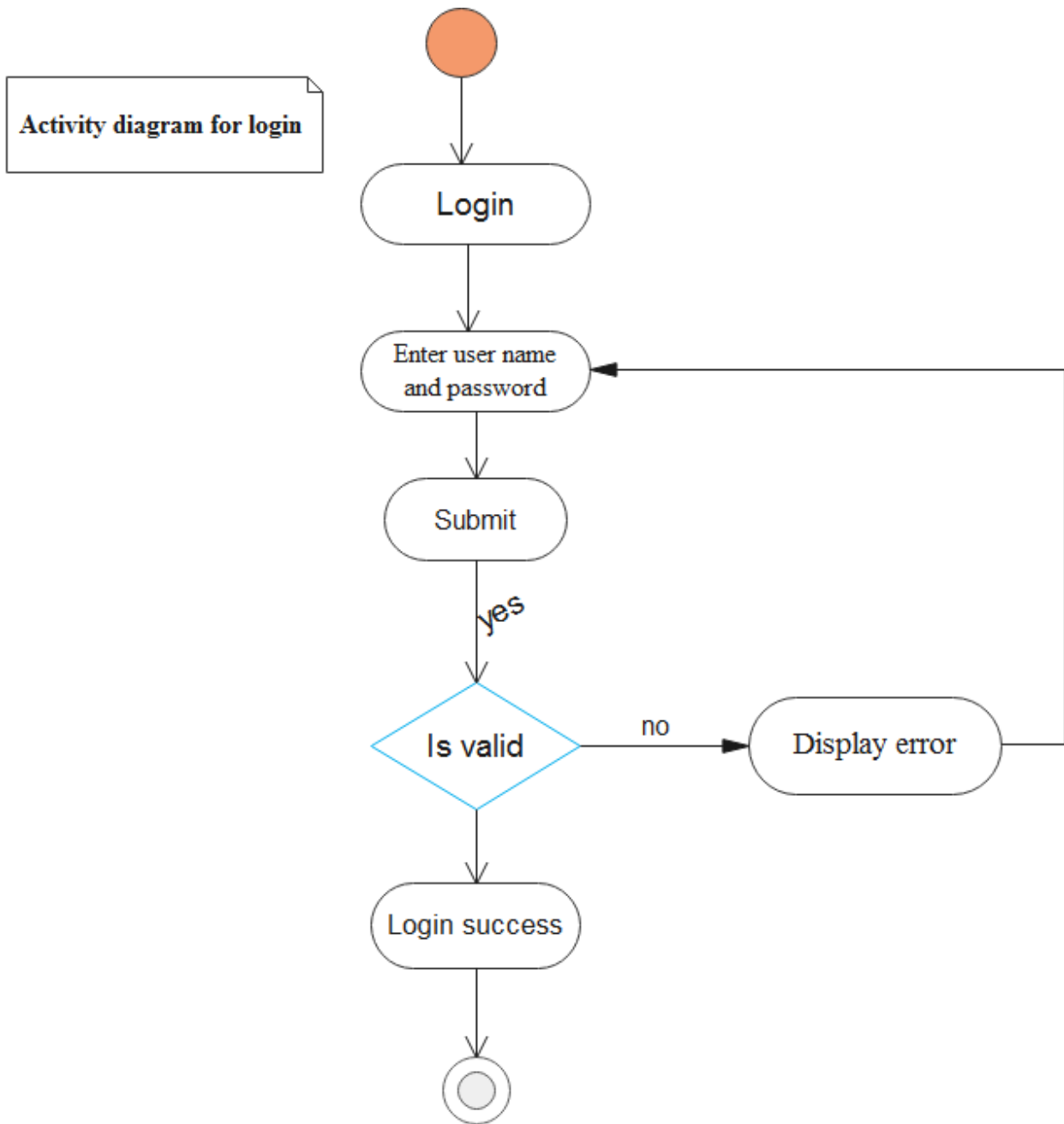


Figure 4.13 Activity diagram for Login Use case

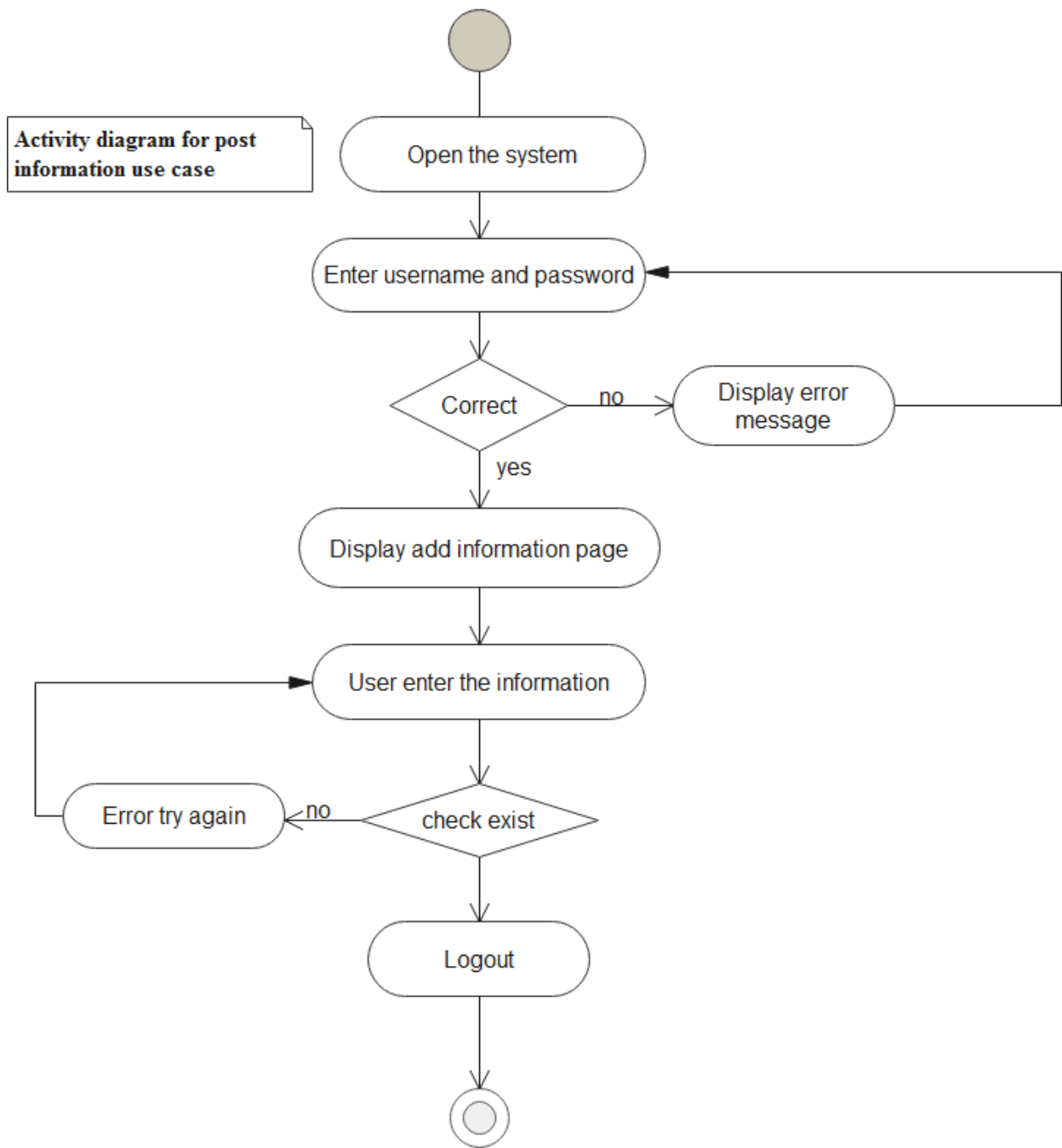


Figure 4.14 Activity diagram for Post Information Use case

Activity Diagram for registration use case

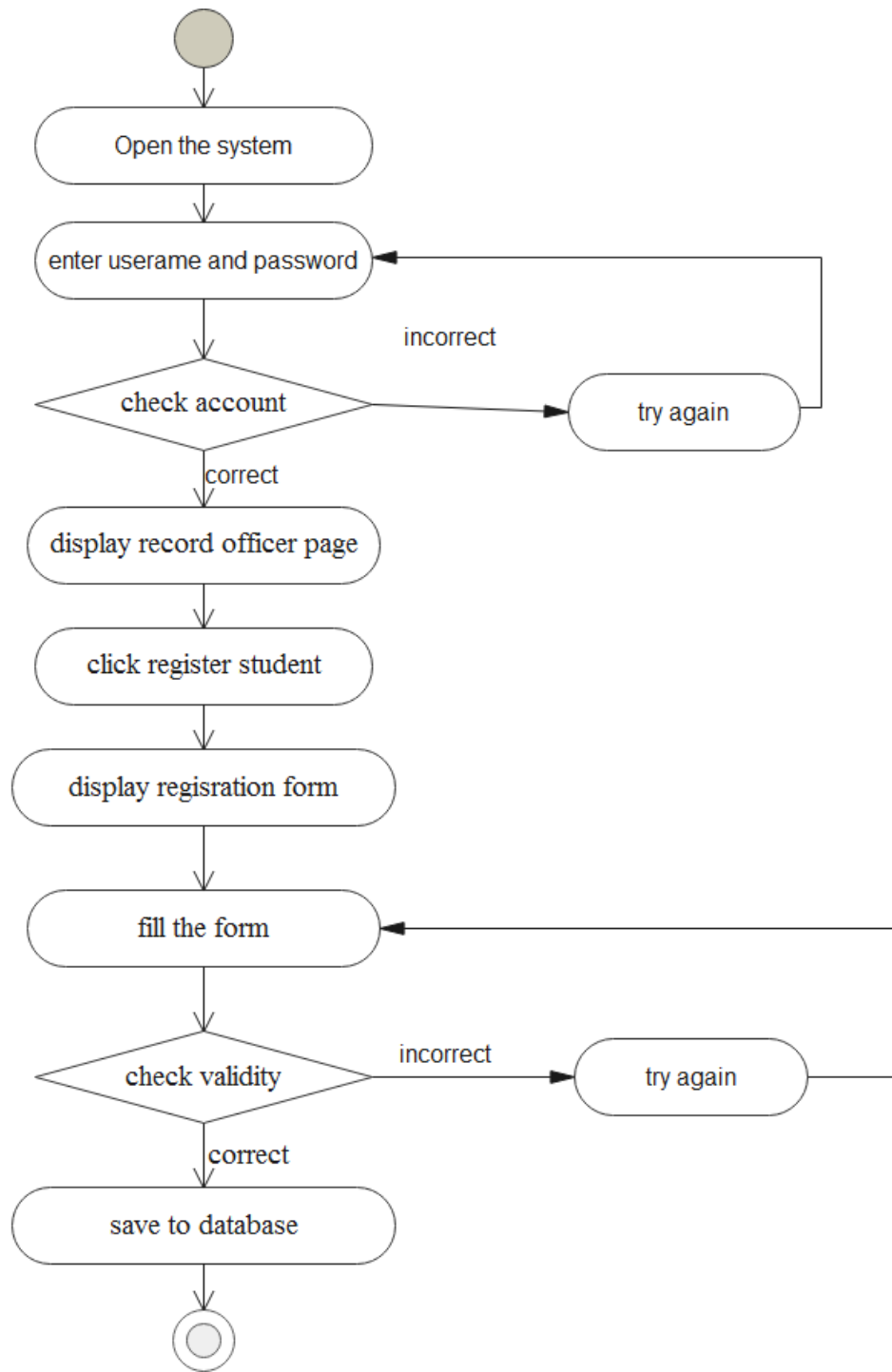


Figure 4.15 Activity diagram for registration Use case

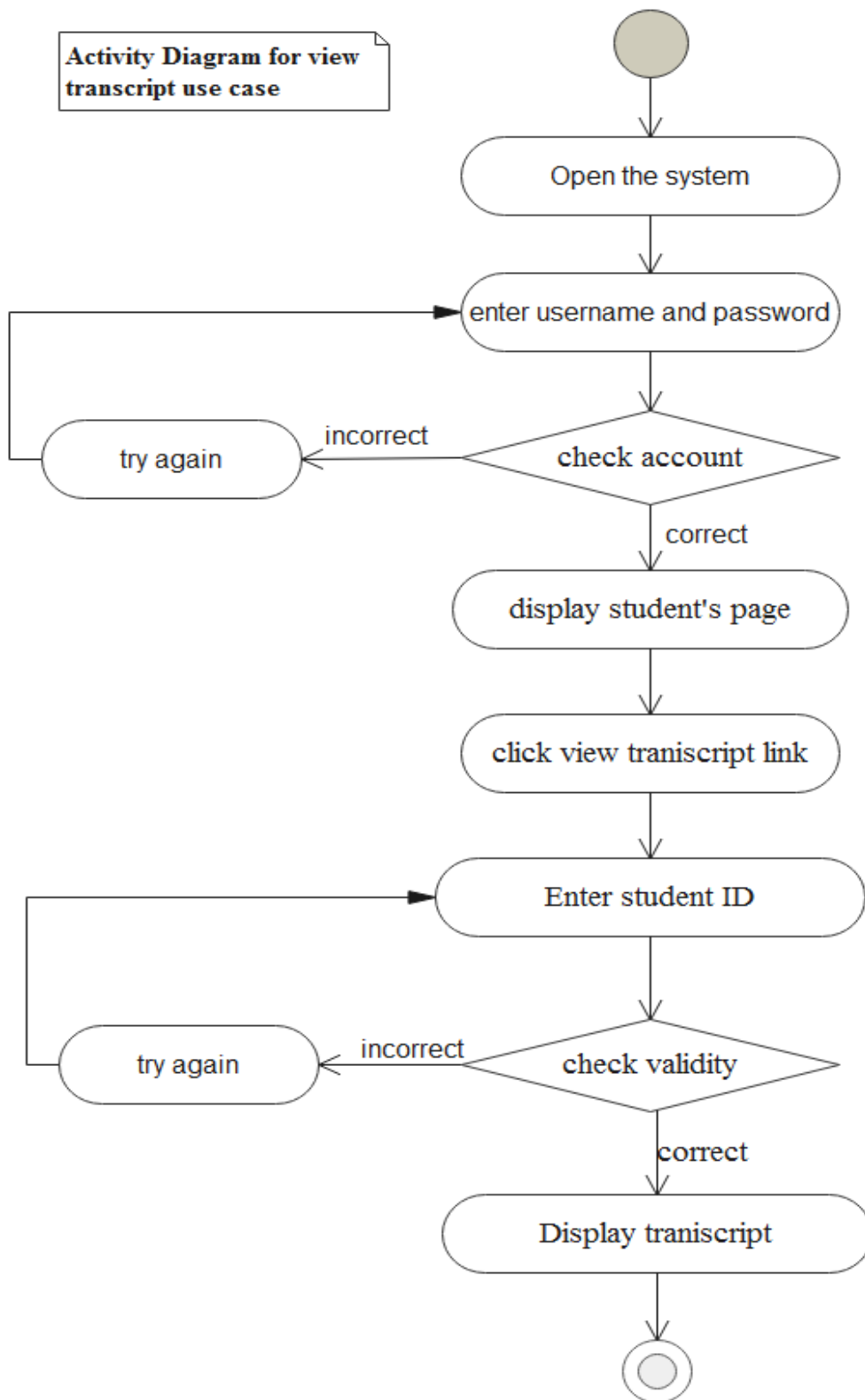


Figure 4.16 Activity diagram for View Transcript Use case

Activity Diagram for insert mark use case

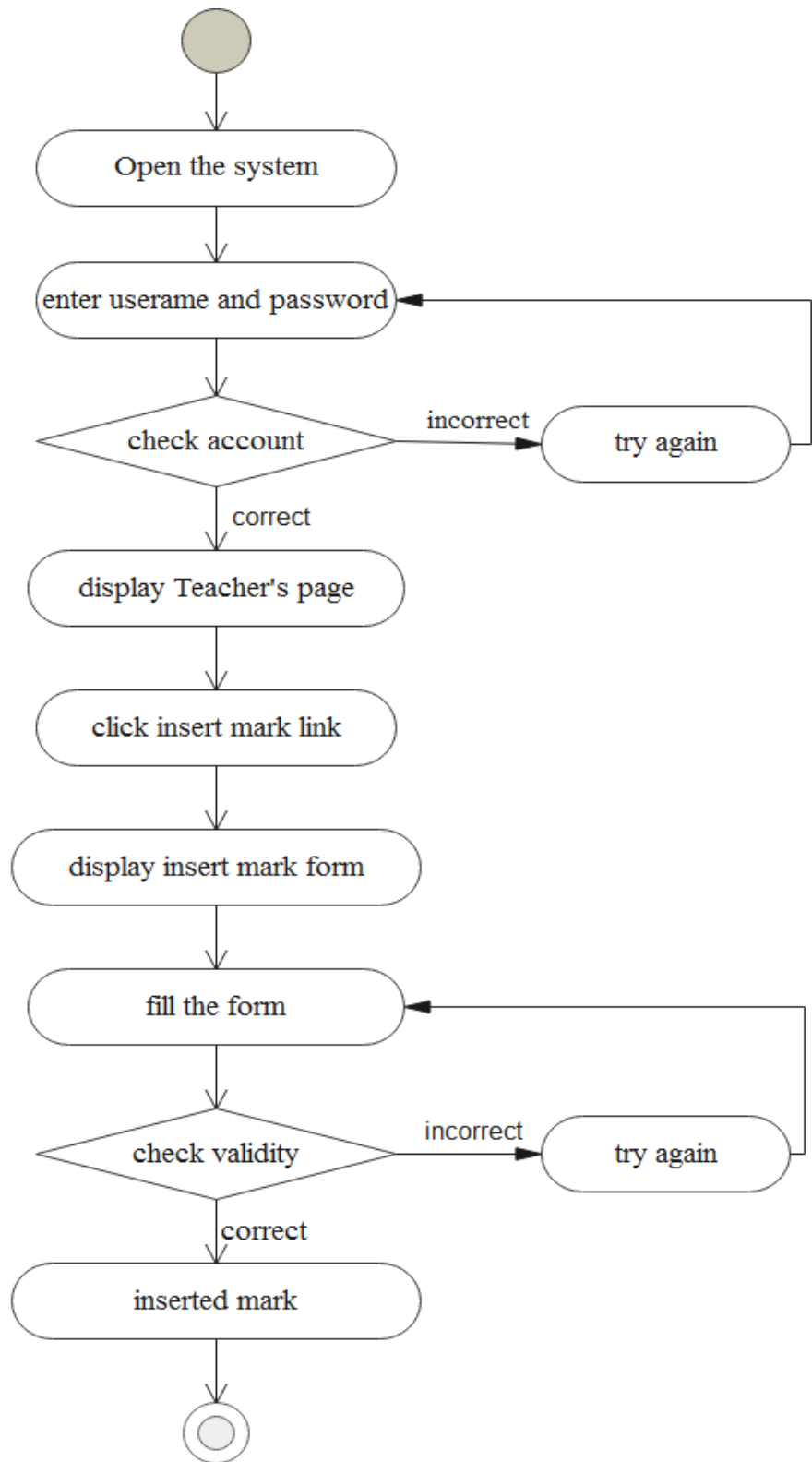


Figure 4.17 Activity diagram for Insert Mark Use case

Activity Diagram for prepare transcript use case

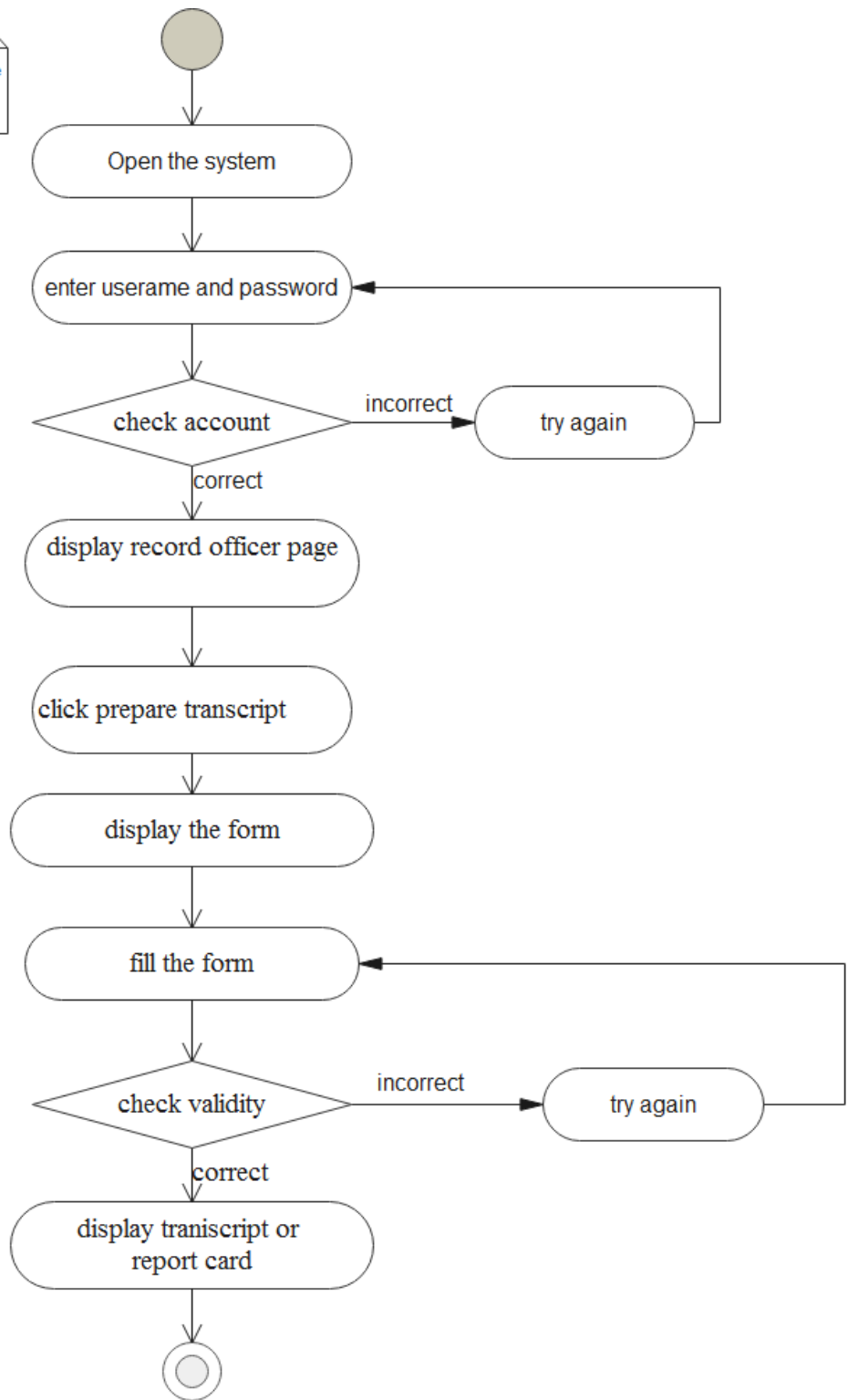


Figure 4.18 Activity diagram for Prepare Transcript Use case

4.3.3 State Chart Diagram

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It is a behavioral diagram and it represents the behavior using finite state transitions. Also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli. It provides a way to model the various states in which an object can exist.

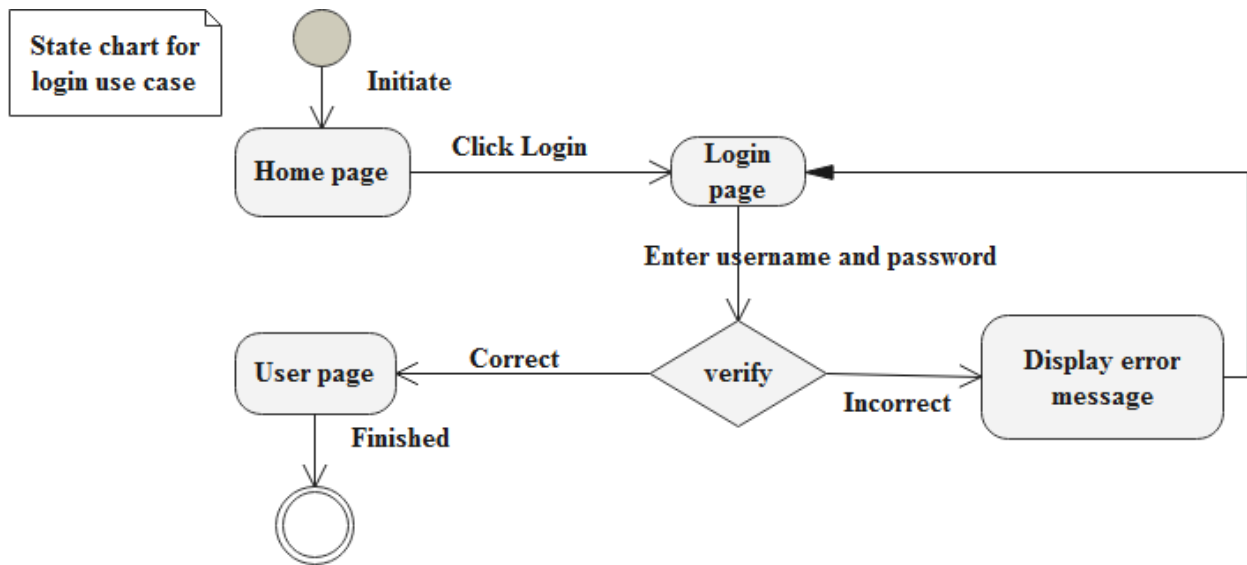


Figure 4.19 State chart for Login

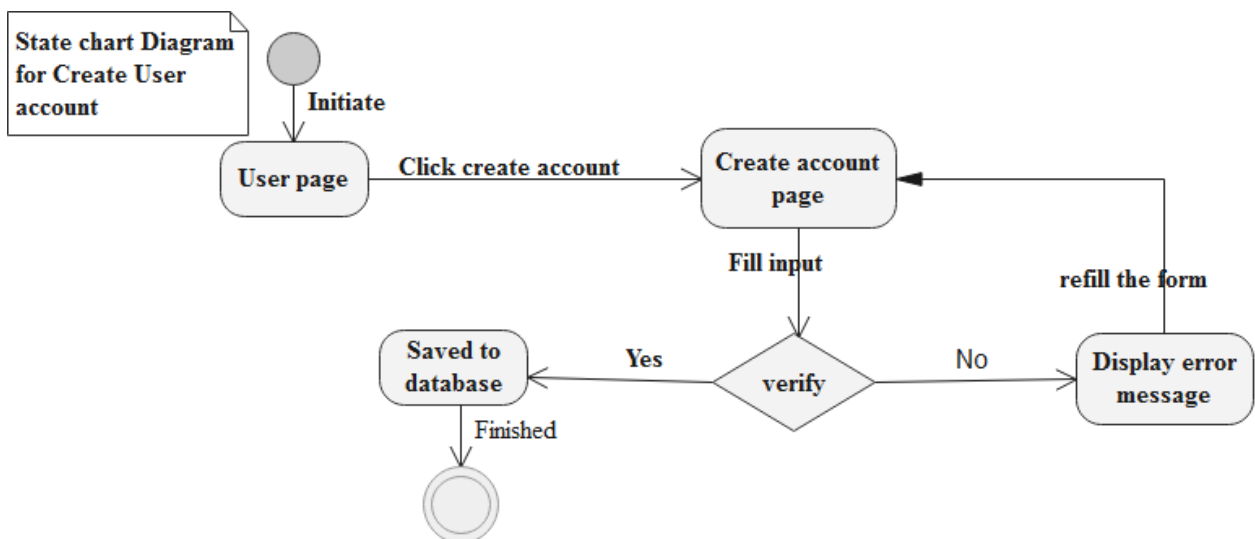


Figure 4.20 State chart for create account

State chart diagram for deactive account

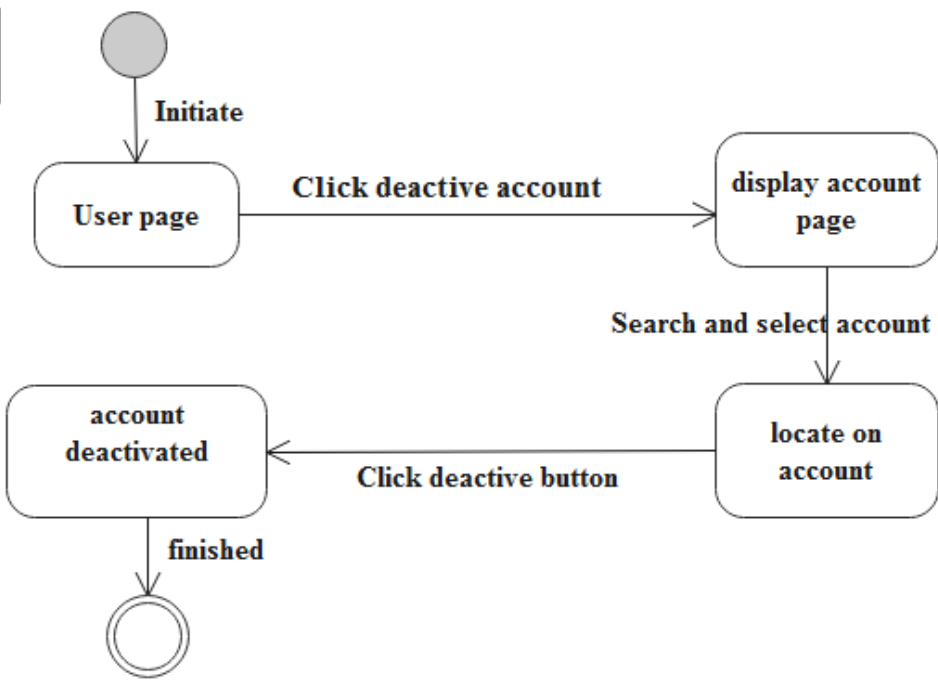


Figure 4.21 State chart for deactivate account

CHAPTER FIVE

5. SYSTEM DESIGN

In this chapter provide a brief overview of the design goals, current and proposed software architecture, Hardware/software mapping, detailed class design and database design, Persistent data management and Access control and Security. Systems design is the process of defining the architecture, components, modules, and interfaces for a system to satisfy specified requirements. Design of software involves conceiving, planning and specifying the externally observable characteristics of the software product. System design is one of the activities that are required to build and verify software. Design is the only way that we can accurately translate customer's view into finished software product or system.

5.1 Design Goal

The design goals are derived from non-functional requirements that means nonfunctional requirement is the description of the feature characteristics and attribute of our proposed system. Design goals describe the qualities of the system that the developers should consider.

User interface and human factors

The user interface of our system should have a user-familiar and attractive interface so the system should easy and understandable for every user. To increase understandability of the system, we will use bootstrap (icon), button, text field, and preferable and attractive color user interface.

Security Issues

Our system is protected against internal and external intrusions or against an authorized user by using the following security controls.

User authentication: - The system should authenticate the users when they try to access by asking user name and passwords.

The system should keep the personal information protection guideline during authentication.

User accessibility control: - User can access only their privilege concern areas of the system components.

Performance: - The system should meet the following performance criteria's

- **Response time:** The speed imposed on the system. The system should responsive maximum number of tasks with minimum times.
- **Throughput:** number of tasks accomplished in a fixed time.
- **Memory:** memory space available for speed optimizations should use efficiently.

Quality Issues

Availability: - Our system will be available for 24 hours, if intranet network available

Usability

User operability: -The system will offer simple navigation any user with basic computer knowledge so, can operate function.

Reliability: - The proposed system will minimize crash during its runtime, since more than one user could use the system simultaneously.

- Good validations for user inputs will be done.
- Avoid incorrect storage of records.

Backup and Recovery

Storing data in another place for backup purpose, if the system is destroying then it is easy to get the lost data. This can be done by placing the data in another place. If the data is failed or lost, then the lost data can be easily recovered the database.

Physical Environment

The system is deployed or installed on the server computer, but for more feature we recommend that the system to deploy on the cloud that is free from any disaster.

Resource Issues

We develop the system to minimize wastage of resource like reducing human power. In the future, the system use computer, flash, server etc.

Documentation

Since we develop user manual documentation, the user that use the system cannot face any difficulty concerning how to use the system. The user guide will aid users who work on the

system to have a detailed and clear understanding of the system steps and functionalities. The developed system has full documentation if some failure occurred could maintain easily.

5.2 Proposed System Architecture

In our system, we will use three-tier architecture these are presentation tier, application tier and database tier.

Presentation Tier

The Presentation Tier displays information related to the services or operations within our system. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network.

Application Tier

The Application Tier (middle tier) controls the business logic and processes based on Service calls from the Presentation Tier (User Interface). The Application Tier is separate from the presentation tier and, as its own layer, controls an application's functionality by performing detailed processing.

Data Access Tier

This tier consists of database servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers. Giving data its own tier also improves scalability and performance.

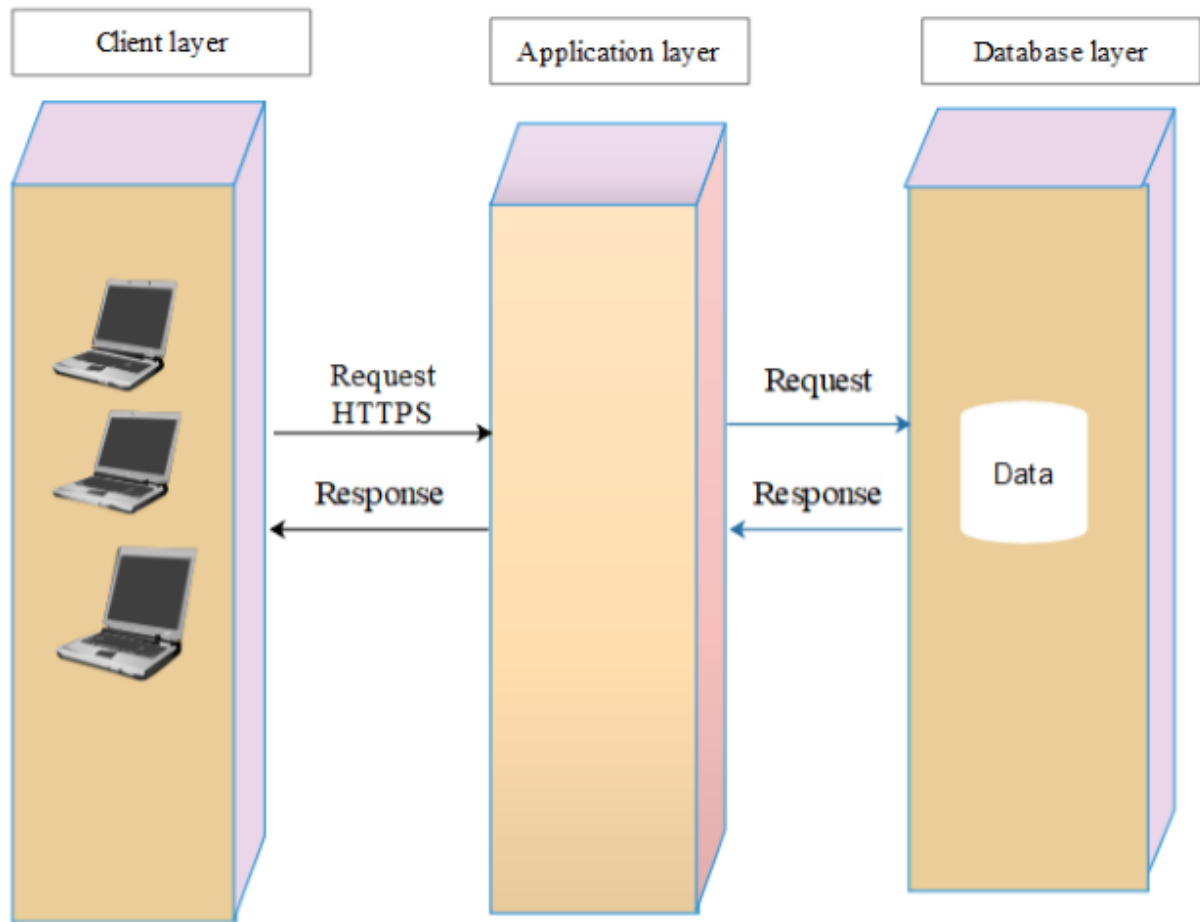


Figure 5.22 Proposed system architecture

5.2.1 Subsystem Decomposition and Description

Decomposing the system enable us to minimize the complexity of the system by decomposing in to sub systems. Subsystems identified from our system are the following

Manage user account subsystem

- Create account
- Update account
- Active account
- Deactivate account
- Delete account

Manage registration subsystem

- Add student

- Update student
- Delete student
- View student list

Manage insert mark subsystem

- Insert mark
- Update mark
- Add mark
- View mark
- Delete mark

Manage record card subsystem

- Prepare report card
- View report card
- Update report card
- Delete report card

Manage transcript subsystem

- Prepare transcript
- Update transcript
- View transcript
- Delete transcript

Feedback subsystem

- Send feedback
- View feedback

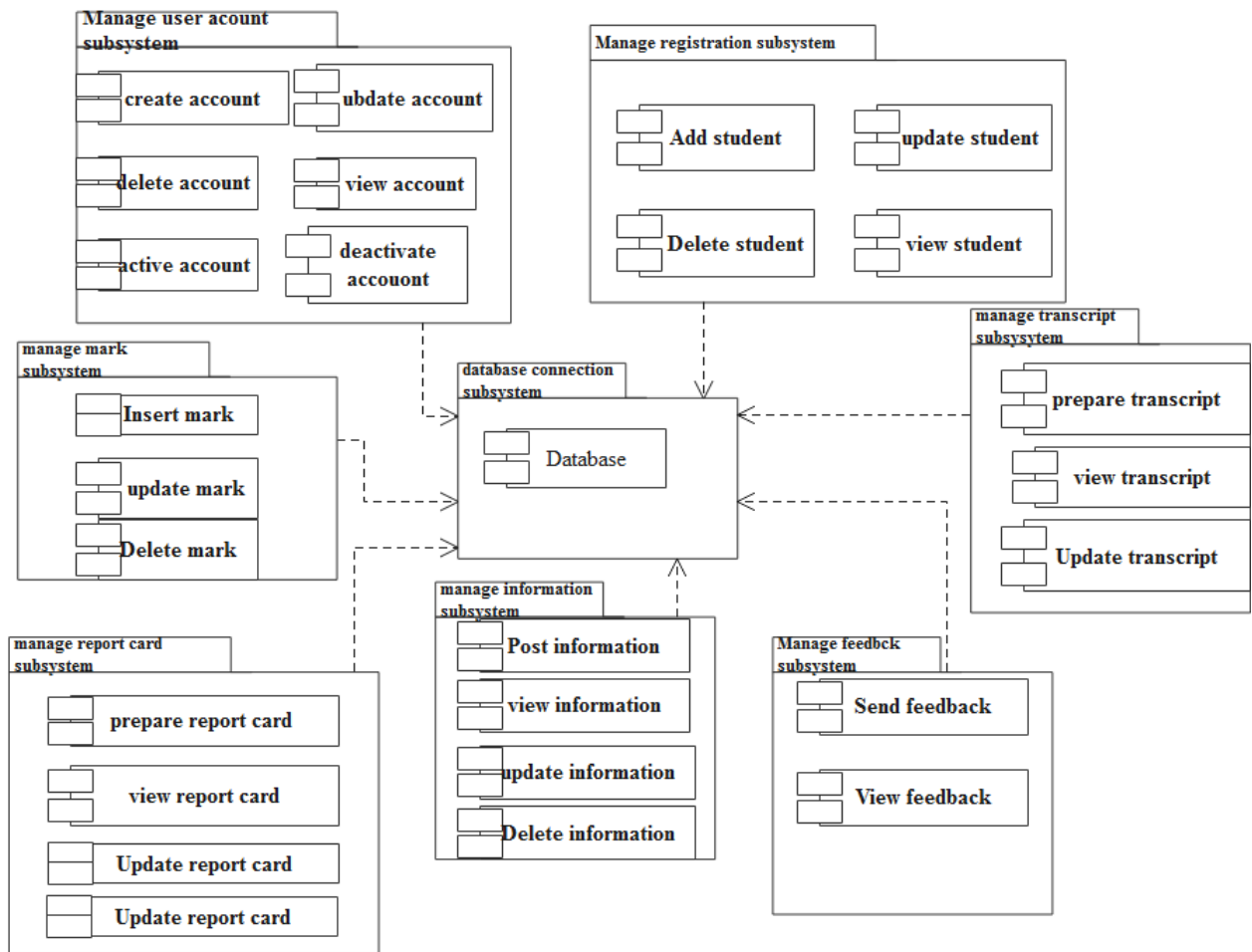


Figure 5.23 Subsystem Diagram

5.2.2 Hardware and software Mapping

Hardware or software mapping describes how subsystems are assigned to hardware and customized components. We use UML deployment diagram to explain the hardware/software mapping of our proposed system.

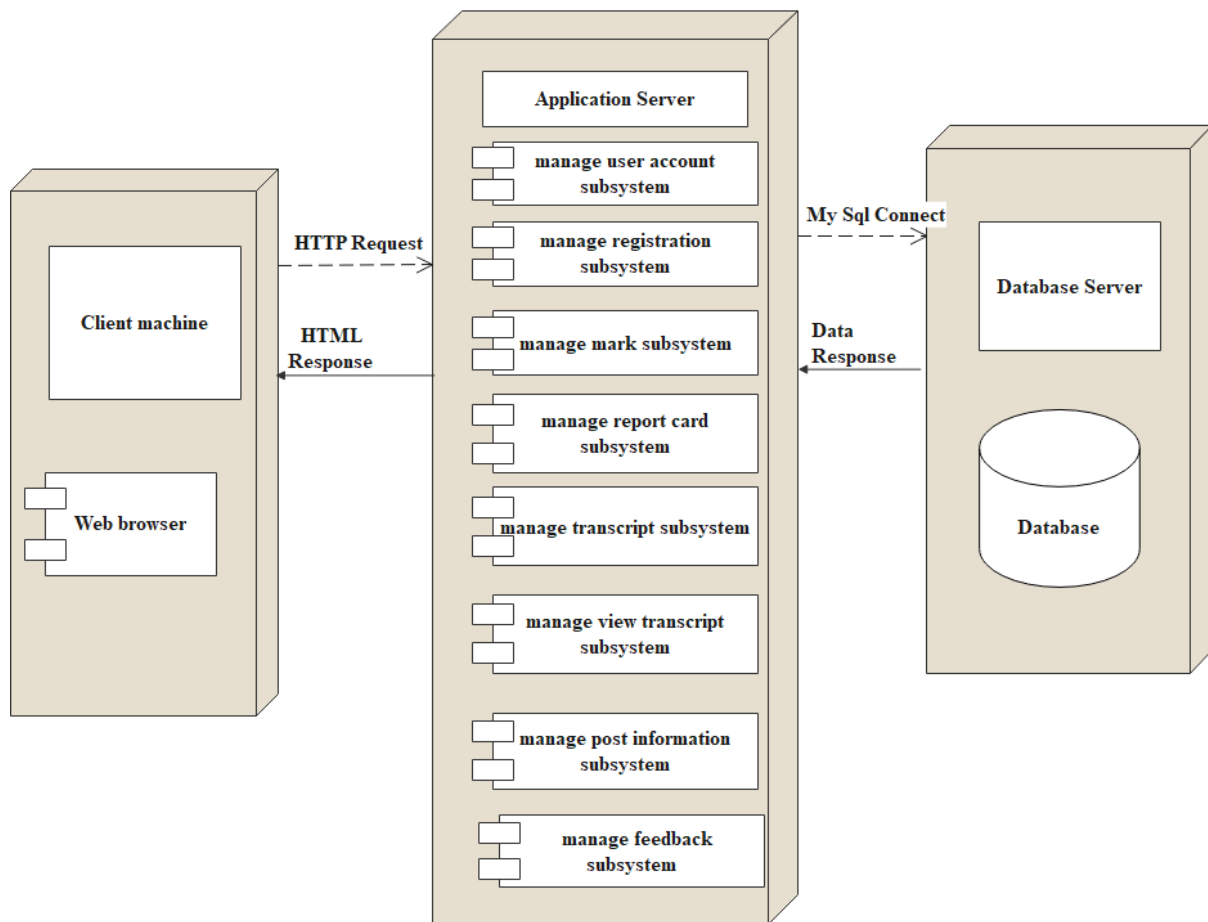


Figure 5.24 Deployment Diagram

5.2.3 Detailed Class Diagram

Detailed class diagrams 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 the classes. The class diagram with its corresponding objects will be present in the following diagram.

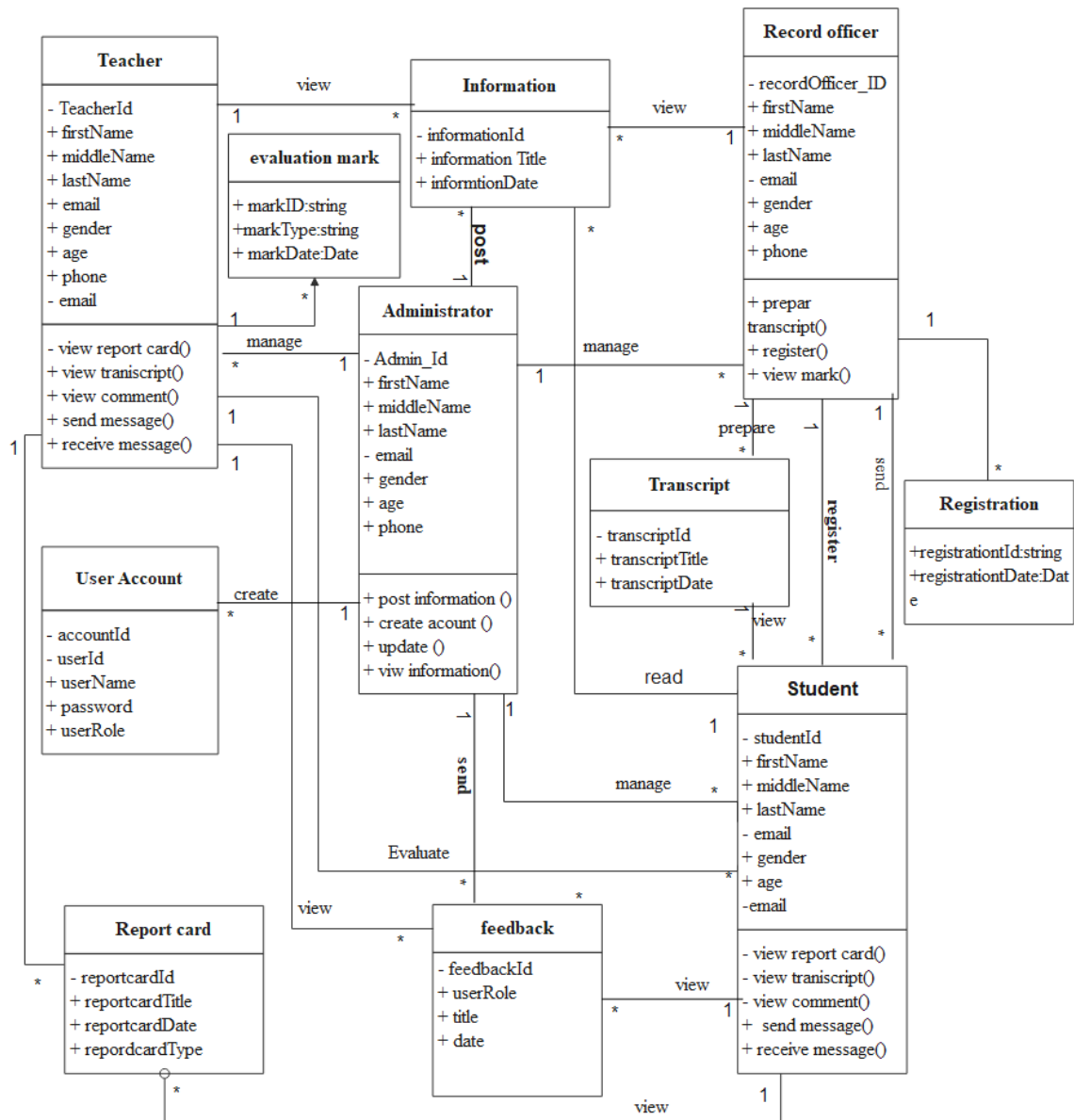


Figure 5.25 Detailed class diagram

5.2.4 Persistent Data Management

Persistent data management describes the persistent data stored by the system and the data management infrastructure required for it. Persistence of our object can be achieved by relational database since it used as machine to make object persistent. It describes the persistent data aspect of software system. Our system includes the basic table that handles the data of system implemented using SQL server.

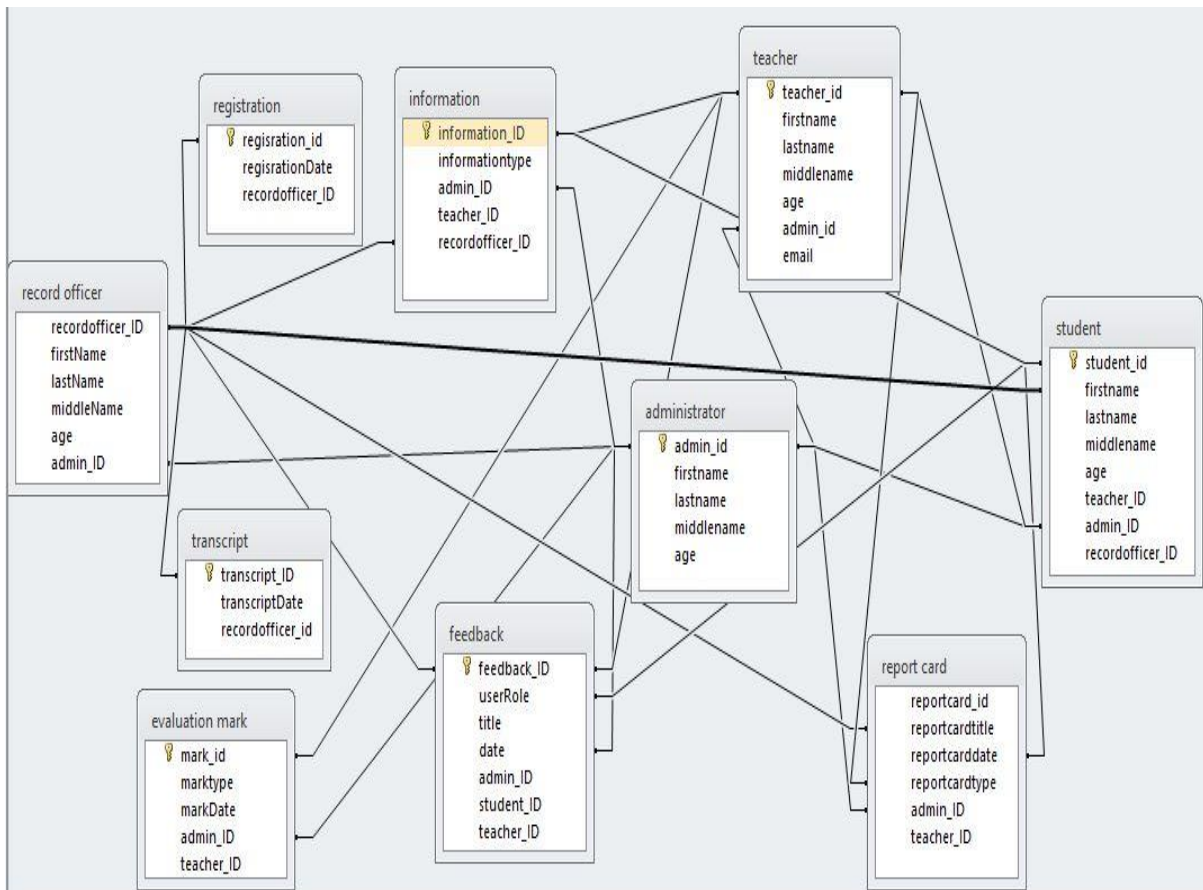


Figure 5.26 Persistence data management

5.2.5 Access control and Security

Access control is a process by which users are granted access and certain privileges to systems, resources or information.

Table 5.14 Access control privileges in our System

Actors	Privilege
Administrator	<ul style="list-style-type: none"> ❖ Login ❖ Manage user account ❖ Send report ❖ Send feedback
Teacher	<ul style="list-style-type: none"> ❖ Login ❖ Insert student mark ❖ Update student mark

	<ul style="list-style-type: none"> ❖ Delete student mark ❖ Send feedback
Record officer	<ul style="list-style-type: none"> ❖ Login ❖ Register student ❖ Prepare transcript ❖ Send feedback
Student	<ul style="list-style-type: none"> ❖ Login ❖ View transcript ❖ View feedback and registered

5.3 Package Design

Package diagram shows the decomposition of subsystems into packages and overview of each package dependencies with other packages.

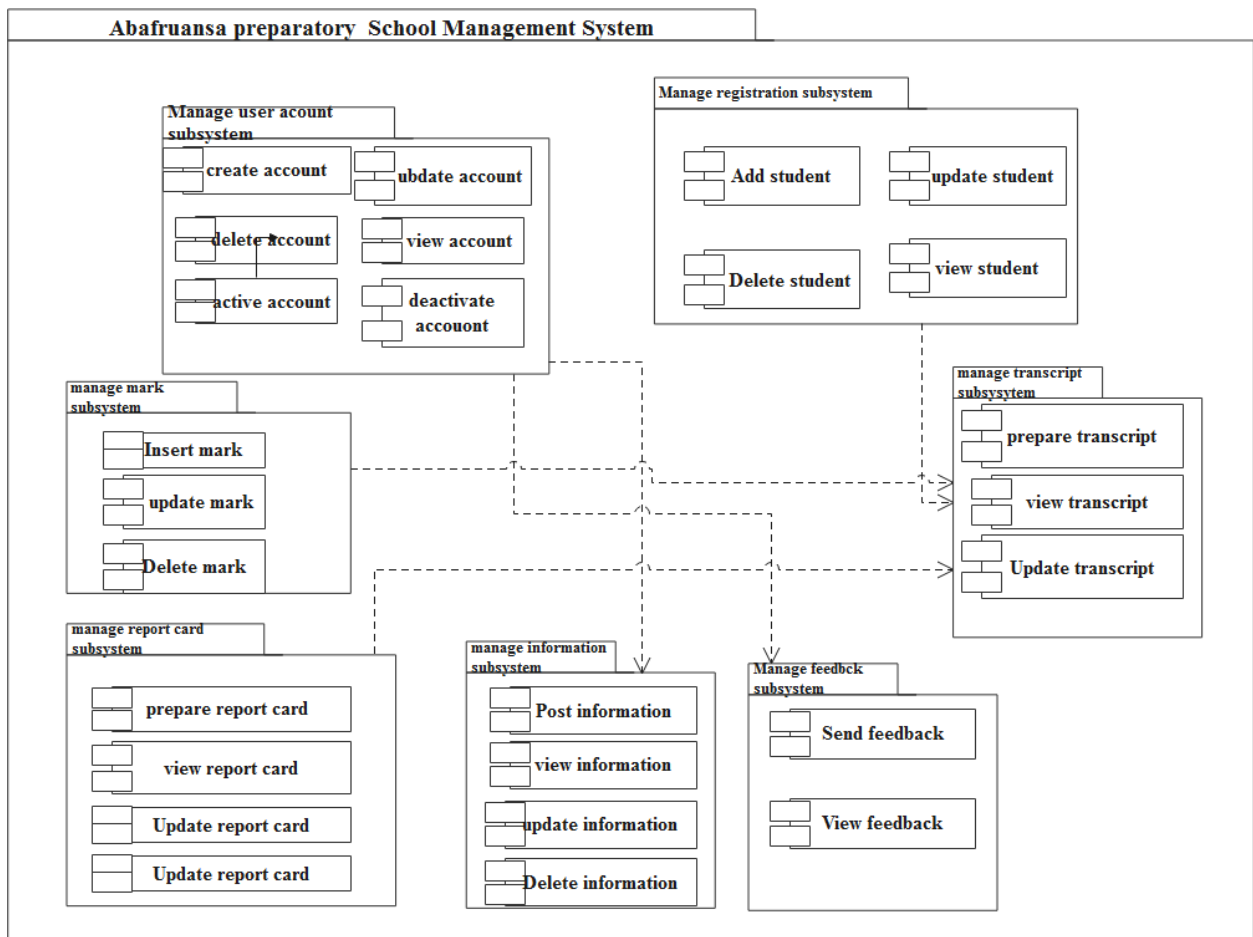


Figure 5.27 Package diagram

5.4. Algorithm Design

➤ Algorithm design for login

Enter Username and Password (U, P)

If (U is equal to the database Username & P is equal to the database Password)

 Then login successful

Else

 Login failed

End If.

➤ Algorithm design for login

Click on create account link

Form is displayed

Fill (data)

IF (Data is valid)

 Display successful message.

Else

 Display invalid input message

End IF.

5.5 User Interface Design

The bellow image Shows of sample user interface screen or graphical user interface of our project.



Figure 5.28 Home page user interface



ABA FRANSUA PREPARATORY SCHOOL
STUDENT INFORMATION MANAGEMENT SYSTEM

HOME INFORMATION CLUBS CONTACT US ABOUT US DEVELOPERS

Wednesday, 25 August 2021, 3:14 AM

CALANDER

«	2021						»
«	August						»
						<u>1</u>	
<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	
<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	
<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>	
<u>23</u>	<u>24</u>	<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	
<u>30</u>	<u>31</u>						

Administrator Login Page

User Name:

Password:

Activate Win
Go to Settings t

Figure 5.29 Login user interface

CHAPTER SIX

6. IMPLEMENTATION AND TESTING

Implementation is one of parts of the project development phase where project inputs are converted to project outputs. In implementation phase, main task is coding. In implementation; we tried to put into practice what was proposed in the project document i.e., transforming the project proposal into the actual project.

6.1. Implementation of Database

We have chosen MySQL database because MySQL database is fastest database with advanced performance and more capabilities to simplify operation of data base in a short time. MySQL does have a few advantages for web developers compared to some other systems.

- **Price.** The price of MySQL commercial licenses is less expensive than same configuration of Oracle. Each table can have a different storage engine.
- **It's easy to use.** Setting up and working with MySQL databases is relatively straightforward.
- **It works well with PHP.** As of version 5.3, PHP has a native MySQL driver that is tightly coupled with the PHP engine, making it a good choice for PHP coders.

Sample code

Database: `school`

Table structure for table `administrator`

```
CREATE TABLE IF NOT EXISTS `administrator` (  
  `firstname` varchar(30) NOT NULL,  
  `lastname` varchar(30) NOT NULL,  
  `email` int(30) NOT NULL AUTO_INCREMENT,  
  `pword` varchar(30) NOT NULL,  
  `ambot` varchar(30) NOT NULL,  
  `number` varchar(30) NOT NULL,
```

`city` varchar(30) NOT NULL,

PRIMARY KEY (`email`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

Table structure for table `teacher`

CREATE TABLE IF NOT EXISTS `teacher` (

`recordno` int(30) NOT NULL AUTO_INCREMENT,

`email` varchar(30) NOT NULL,

`firstname` varchar(30) NOT NULL,

`lastname` varchar(30) NOT NULL,

`passconf` varchar(30) NOT NULL,

`phone` int(30) NOT NULL,

`username` varchar(15) NOT NULL ,

`password` varchar(15) NOT NULL,

PRIMARY KEY (`recordno`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;

Table structure for table `adminlogin`

CREATE TABLE IF NOT EXISTS `adminlogin` (

`aid` int(11) NOT NULL AUTO_INCREMENT,

`fname` varchar(30) NOT NULL,

PRIMARY KEY (`aid`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

Table structure for table `login`

CREATE TABLE IF NOT EXISTS `login` (

`username` varchar(15) DEFAULT NULL,

`password` varchar(15) DEFAULT NULL

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
Dumping data for table `login`
```

```
INSERT INTO `login` (`username`, `password`) VALUES
```

```
(", ");
```

Table structure for table `markform2`

```
CREATE TABLE IF NOT EXISTS `markform1` (
```

```
  `stuid` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `sname` varchar(40) NOT NULL,
```

```
  `Semester` varchar(20) NOT NULL DEFAULT 'T',
```

```
  `gsection` varchar(20) NOT NULL DEFAULT '11',
```

```
  `tid` int(15) NOT NULL,
```

```
  `sex` varchar(20) NOT NULL,
```

```
  `amharic` int(11) NOT NULL,
```

```
  `english` int(11) NOT NULL,
```

```
  `maths` int(11) NOT NULL,
```

```
  `physics` int(11) NOT NULL,
```

```
  `biology` int(11) NOT NULL,
```

```
  `chemistry` int(11) NOT NULL,
```

```
  `geography` int(11) NOT NULL,
```

```
  `history` int(11) NOT NULL,
```

```
  `civics` int(11) NOT NULL,
```

```
  `it` int(11) NOT NULL,
```

```
  `business` int(11) NOT NULL,
```

```
  `economics` int(11) NOT NULL,
```

```
  `hpe` int(11) NOT NULL,
```

PRIMARY KEY (`stuid`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6667 ;

6.1.1 Implementation of Class Diagram

In class implementation the following activities we have perform:

- We Define attributes with the appropriate data type and access visibilities (private, protected, public).
- We define all methods with appropriate return type, parameters and the corresponding data types and access visibility.

Methods Implemented in Our system

ManageAccount();

SendComment();

viewMark();

#manageFeedback ();

#viewTranscript ();

+viewComment ();

#viewMark();

Register();

ViewReportcard();

postInformation();

viewInformation();

Variables implemented in our system:

+Int id

+VarChar first_Name

+VarChar middle_Name

+VarChar lase_Name

+VarChar gender

+Int age

+VarChar phone

+EmailField email

+VarChar user_Role +VarChar address

TextField Information_Title

6.1.2 Configuration of Application server

We use XAMPP application server because XAMPP is simple and lightweight. Apache distribution is extremely easy to create a local web server for testing and deployment purposes. Everything you needed is to set up a web server-server application (Apache), database (MySQL) and scripting language (PHP). XAMPP works on different operating systems and have the following advantages:

- Its free and user-friendly software available for Windows, Mac, and Linux platforms.
- It's an open-source lightweight software which is very easy to set up, develop, test, and deploy.
- Handles many administrative tasks and saves time by managing configuration changes.

6.1.3 Configuration of Application Security

We validate our inputs by using client and server sides. Validate of the inputs sides allow the user to fill correct information in our system we validate all the input to enter the correct format. Example the name must be in the form of string not include other like special character number our system refuse this inputs. In addition, age must be in the form of number.

6.1.4 Implementation of User Interface

The user interface of our system should have a user-familiar and attractive interface so the system should easy and understandable for every user.



Figure 6. 30 Home page

Administrator Login Page

User Name:

Password:

Figure 6.31 Administrator Login page

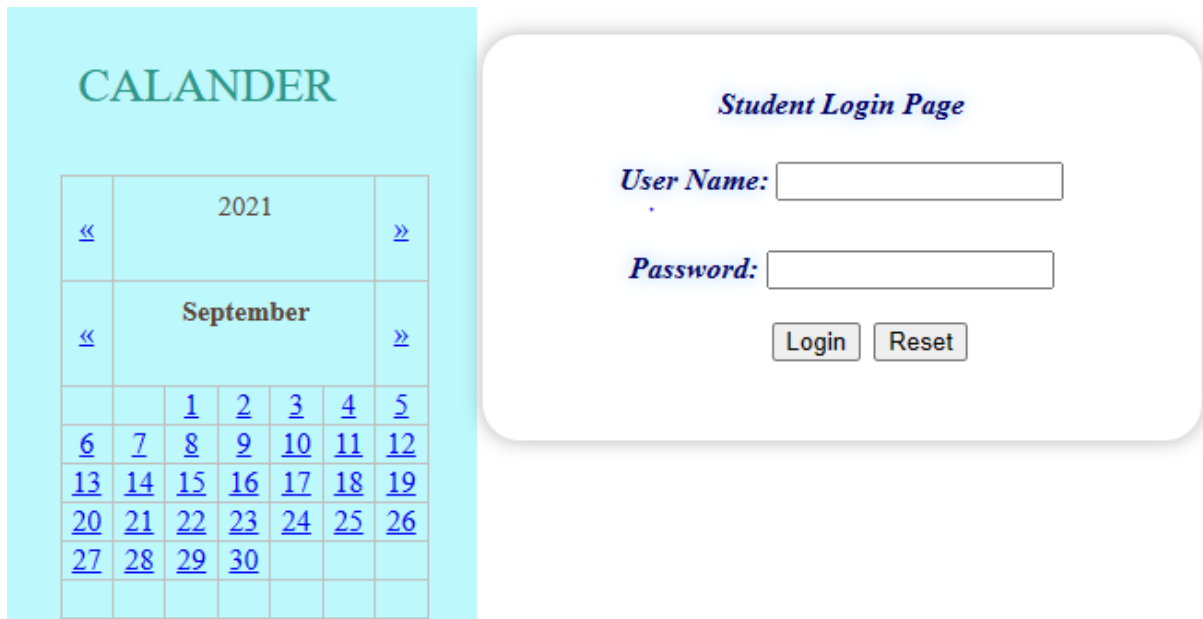


Figure 6.32 Student Login page

6.2 Testing

Software Testing is considered as the one of the important and necessary phases in SDLC because of the following reasons given below:

- Testing always helps to verify that complete software requirements are implemented correctly or not, means it is implemented according to the defined requirements or not.
- Testing definitely helps in identifying defects/bugs and ensuring they are recognized/addressed before software deployment stage.
- Testing also demonstrates that software/application appears to be working according to specifications and the performance requirements that have been defined is also met.

6.2.1 Testing tools and Environments

Testing tools are important for the success of testing phase and naturally the success of product. In our unit testing phase XAMPP, Sublime Text3, and Google Chrome are used. **XAMPP**: - is an open-source tool used to handle the administration of MySQL database. It allows user to view, modify, add and delete tables and their records in database. This tool will be used in order to check the correctness of the database and database related part of the project.

Sublime Text3:- is used to edit PHP, HTML, JavaScript and CSS code.

6.2.2 Unit Testing

It is done at the source or code level for language-specific programming errors such as bad syntax, logic errors or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the program's correctness. It is a way of testing each of the system functionality independently. Accordingly, the team has tested each one of the systems activities and the rest accompanying activities independently using different user input, different login mechanisms and any technique of fault finding so that an incorrect functioning of the activities is become corrected at the right time.

When we check unit test of Student registration, the following

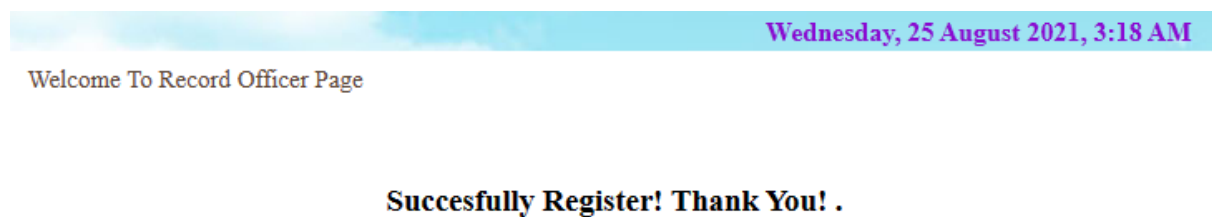


Figure 6.33 registration with correct information

When do not insert correct Average, it display the following

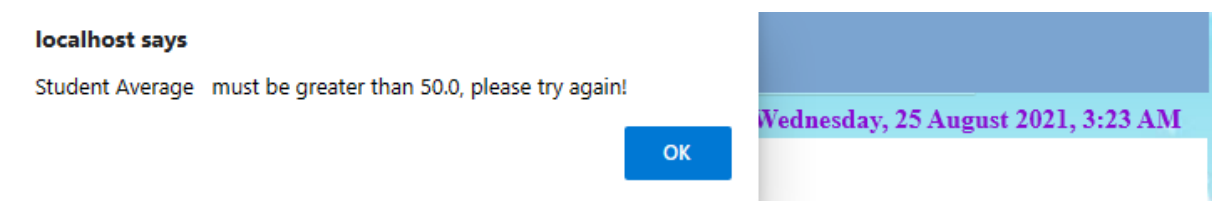


Figure 6.34 registration without correct information

6.2.3 System Testing

In this level of testing process, we have examined how the whole subsystems work together to achieve the desired goal (user's requirements of the system). The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, under this testing is mainly concerned with areas such as performance, security, validation, and load. But we will more focus only on function validation and performance.

Sample test:

- Evaluate the functionality of the subsystem after a combination of individual subsystem whether it works correctly or not.
- Check the coherence and coupling of each subsystem.
- Check the overall functionality that achieves the user's requirement.

6.2.4 Integration Testing

In this level of testing, we have examined how the different procedures work together to achieve the goal of the subsystem. The type of integration testing that we have follow bottom up. So, we integrate each component from single function to the main function incrementally.

Sample Tests:

- Check the interaction between individual functionality which performs the specific tasks.
- Evaluate the functionality of the subsystem after combination all individual functionality.
- Identify the Independence of each subsystem with another subsystem.

6.2.5 Acceptance Testing

It is the process whereby actual users test a completed information system, the end result of which is the users' acceptance of it once they are satisfied with it. Acceptance refers to the fact that users typically sign off on the system and "accept" it once they are satisfied with it. Testing the system in the environment where it will be used. The purpose of acceptance testing is for users to determine whether the system meets their requirements. In acceptance test the team test the system by different user that satisfies the requirements. The user tested by inserting real data. Our system is not accepted.

CHAPTER SEVEN

7. CONCLUSION AND RECOMMENDATION

7.1. Conclusion

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

The analysis helps the team to well understand the major functional areas and processes of the system. Through this method we evaluate the existing system that is manual system weakness. After that, we performed requirements elicitation to discover user and system requirements. This phase consisted of drawing the functional as well as non-functional requirements of the system. Then we have undertaken a major phase in system development process: object oriented Analysis. Here, we developed to model the new system we proposed using UML diagrams: Use case, sequence, and class diagrams Also, we designed the new system user interface prototyping and implementation.

7.2. Recommendation

According to the scope of our project we develop web-based system. Due to shortage of time and some other condition like resources and Covid-19 we do not include some features to the system. We can't do all the tasks that are needed in the system so to enhance the performance and functionality of the system our team believes that this system should be fully operationally by adding some functionality that are not included in the proposed system. We recommended also the next developer can include the following tasks:-

- Developing online exam
- When registration choosing stream (Natural or social).
- Generate mark

REFERENCE

- [1] B. H. W. ., D. T. Alan Dennis, System Analysis & Design, Carolina: Wiley, 2015.
- [2] P. Jalote, A Concise Introduction to, I. Mackie, Ed., India, 2008.
- [3] W. S. Saba, Object-oriented Analysis and Design, 1999.
- [4] M. B. Wendy Boggs, Mastering UML with Rational Rose 2002, 2002.
- [5] J. J. I. & B. G. Rumbaugh, The Unified Modeling Language., 2003.

Appendix

Sample code

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
</head>
<?php
$con = mysqli_connect("localhost","root","");
if (!$con)
{
die('Could not connect: ' . mysqli_error());
}
//if (mysqli_query("CREATE DATABASE school",$con))
//{
//echo "Database created";
//}
//else
//{
//echo "Error creating database: " . mysqli_error();
//}
$db=mysqli_select_db($con,"school");
//$sql = "CREATE TABLE login(username varchar(15),password varchar(15))";
```

```

//mysql_query($sql,$con);
if(!$db){
die("error:".mysql_error());
}
$i=2;
$query="select * from login";
$result=mysql_query($con,$query);
while($row=mysql_fetch_array($result)){
if($_POST['username']==$row['username'] && $_POST['password']==$row['password'])
{
$i=$i+1;
$_session_register("username");
$_SESSION['username']=$row['username'];
}}
$sql="INSERTINTOlogin(username,
password)VALUES('$_POST[username]', '$_POST[password]')";
if (!mysql_query($con,$sql))
{
die('Error: ' . mysql_error());
}
else
echo "1 record added";
mysql_close($con);
if($i>=1)
header("location:ahome.php");

```

```
else
header("location:parent_page.php");
mysqli_close($con);
?>
<body>
</body>
</html>
```