

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

PROJECT TITLE: ABAFRANSUA PREPARATORY SCHOOL MANAGEMENT SYSTEM

Student Name	Student ID
1. Abebe Tariku	CIR/004/10
2. Habtamu Ayalew	CIR/031/10
3. Sintayehu Mitike	CIR/061/10

Advisor name: Mr.Ayele.G

Aug 23, 2021

Wolkite University, Wolkite, Ethiopia

DECLARATION

This is to declare that this project work which is done under the supervision of Mr. Ayele G. and having the title Abafransua preparatory school management system is the sole contribution of: **Abebe Tariku, Habtamu Ayalew and Sintayehu Mitike**. No part of the project has been reproduced illegally (copy and paste) which can be considered as plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: _____

Group Member:

Name	Signature
Abebe Tariku	_____
Habtamu Ayalew	_____
Sintayehu Mitike	_____

APPROVAL FORM

This is to confirm that the project report entitled Abafransua preparatory school management system submitted to Wolkite University, College of Computing and Informatics Department of Computer Science by: **Abebe Tariku, Habtamu Ayalew and Sintayehu Mitike** Is approved for submission.

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

ACKNOWLEDGMENT

First, we would like to thank my lord for his help throughout starting this project from the first chapter up to know and all our success. Next, we would like to extend our thanks to our advisor Mr. Ayele for his heartfelt advice and help to shape up our project through review and criticism thoroughly. We would also like to thank Abafransua general and preparatory school director Ms.Mamat Abidu and record officer Mr. Bizat Grima for their willingness to help us by giving any information about the existing system how it works and their drawback in addition to suggestion. Finally, we would like to express our deep gratitude to our instructors and our trust and closest friends for constant support and goodwill.

Table contents

DECLARATION.....I

APPROVAL FORM..... II

ACKNOWLEDGMENT..... III

ACRONYM, ABBREVIATION AND DEFINITION.....XIII

ABSTRACT..... XIV

CHAPTER ONE..... 1

1. INTRODUCTION..... 1

1.1. Background of organization..... 1

1.2. Statement of problem..... 2

1.3. Objectives of the Project..... 3

1.3.1. General objective..... 3

1.3.2. Specific objective..... 3

1.4. Feasibility analysis of the project..... 3

1.4.1. Technical Feasibility..... 3

1.4.2. Operational feasibility..... 4

1.4.3. Economic feasibility..... 4

1.5. Scope and limitation of the project..... 5

1.5.1. Scope of project..... 5

1.5.2. Limitation of our project..... 5

1.6. Significance of the Project..... 5

1.7. Beneficiary of the Project..... 6

1.8. Methodology of the Project..... 6

1.8.1. Data Collection Tools..... 6

1.8.2. Development Tools and Technologies..... 6

1.8.3. Front end....7
1.8.4. Back end....7
1.8.5. System Analysis and Design....7
1.8.6. System Development Model....8
1.8.7. System Testing Methodology....8
1.8.8. Deployment Environment....8
CHAPTER TWO....9
2. DESCRIPTION OF THE EXISTING SYSTEM....9
2.1. Introduction of Existing System....9
2.2. Users of Existing System....9
2.3. Major Functions of the Existing System....10
2.4. Drawbacks of the Existing System....10
2.4.1 Performance....10
2.4.2 Security and Controls....10
2.4.3 Input and output accuracies....10
2.4.4 Economics....11
2.4.5 Data storage problem....11
2.5. Business Rules of the Existing System....11
CHAPTER THREE....12
3. PROPOSED SYSTEM....12
3.1. Functional requirements....12
3.2. Nonfunctional requirement....13
3.2.1. Security....13
3.2.2. Response time....14
3.2.3. Usability....14

3.2.4. Error Handling....14
3.2.5. User Interface and Human Factor.... 14
3.2.6. Efficiency.... 14
3.2.7. Resource issue.... 14
3.2.8. Hard ware consideration....15
3.2.9. Physical environment.... 15
3.2.10. Backup and recovery....15
3.2.11. Performance consideration....15
3.2.12. Quality issue....15
3.2.13. Documentation.... 15
CHAPTER FOUR....17
4. SYSTEM ANALYSIS....17
4.1. System model....17
4.1.1 Use case diagram....17
4.1.2 Use case description....20
4.1.3. Use case scenario....33
4.2. Object model....34
4.2.1. Class diagram.... 34
4.2.2. Data Dictionary.... 36
4.3. Dynamic model....39
4.3.1. Sequence diagram....39
4.3.2. Activity diagram....48
4.3.3. State chart diagram....53
CHAPTER FIVE.... 55
5. SYSTEM DESIGN....55

5.1 Introduction....55
5.2. Goals of design....55
5.3. Proposed System Architecture....57
5.3.1. Hardware/Software Mapping....58
5.3.2. Subsystem Decomposition and Description....60
5.3.3 Detailed Class Diagram....62
5.3.4. Persistent Data Management....64
5.3.5. Access Control and Security....65
5.4. Package....66
5.5. Algorithm design....67
5.6. User interface design....69
CHAPTER 6....71
6. IMPLEMENTATION AND TESTING....71
6.1. Implementation of the Database....71
6.2. Implementation of the Class Diagram....71
6.3. Configuration of the Application Server....74
6.4. Configuration of Application Security....74
6.5. Implementation of User Interface....74
6.6. Testing....74
6.6.1. Testing Tools and Environment....74
6.6.2. Unit Testing....74
6.6.3. Integration Testing....75
6.6.4. System Testing....75
6.6.5. Acceptance Testing....75
CHAPTER SEVEN....76

7. CONCLUSION AND RECOMMENDATION..... 76

7.1. CONCLUSION..... 76

7.2. Recommendation..... 77

REFERENCES..... 78

APPENDICES..... 79

List of Figure

Figure 4. 1: use case diagram..... 19

Figure 4. 2: class diagram.....35

Figure 4. 3: login sequence diagram.....40

Figure 4. 4: take exam sequence diagram.....41

Figure 4. 5: student registration of sequence diagram.....42

Figure 4. 6: prepare exam sequence diagram..... 43

Figure 4. 7: prepare card of sequence diagram.....44

Figure 4. 8: post information of sequence diagram..... 45

Figure 4. 9: add teacher of sequence diagram..... 46

Figure 4. 10: view grade of sequence diagram.....47

Figure 4. 11: activity diagram login form.....48

Figure 4. 12: activity diagram for view transcript/card..... 49

Figure 4. 13: activity diagram for send feedback and comment..... 50

Figure 4. 14: activity diagram for student register..... 51

Figure 4. 15: activity diagram for post information..... 52

Figure 4. 16: state chart for login..... 53

Figure 4. 17: state chart for register.....54

Figure 4. 18: state chart for post information..... 54

Figure 5. 1: overall architecture of the system..... 58

Figure 5. 2: deployment Diagram.....59

Figure 5. 3: component diagram.....61

Figure 5. 4: detail class diagram.....63

Figure 5. 5: Persistence data management diagram..... 64

Figure 5. 6: package diagram..... 66

Figure 5. 7: login page..... 69

Figure 5. 8: register form..... 70

Figure 5. 9: time table..... 93

Figure 5. 10: transcript..... 94

List of table

Table x. 1: acronym, definition and abbreviation.....XIII

Table 1.1: Number of employers and students both grade11 and 12..... 2

Table 4. 1: use case description manage user account..... 20

Table 4. 2: use case for description of login.....21

Table 4. 3: use case for description of post information..... 22

Table 4. 4: use case for description of add teachers..... 23

Table 4. 5: use case for description of arrange student..... 24

Table 4. 6: use case for description of prepare transcript.....25

Table 4. 7: use case for description of view comment..... 26

Table 4. 8: use case for description of prepare exam..... 27

Table 4. 9: use case for description of view attendance..... 28

Table 4. 10: use case for description of sign attendance..... 29

Table 4. 11: use case for description of take exam.....30

Table 4. 12: use case for description of prepare card..... 31

Table 4. 13: use case for description of view result..... 32

Table 4. 14: use case for description of check payment.....33

Table 4. 15: data dictionary of login.....36

Table 4. 16: data dictionary of post information..... 36

Table 4. 17: data dictionary of prepare card.....36

Table 4. 18: data dictionary of class schedule..... 37

Table 4. 19: data dictionary of add teacher..... 37

Table 4. 20: data dictionary of register student..... 37

Table 4. 21: data dictionary of payment of student..... 37

Table 4. 22: data dictionary of attendance of student.....38

Table 4. 23: data dictionary of manage of account.....38

Table 4. 24: data dictionary of parent.....38

Table 4. 25: data dictionary of take exam..... 39

ACRONYM, ABBREVIATION AND DEFINITION

No	Acronym& Abbreviation	Definition
1	CSS	Cascade Style Sheet
2	GUI	Graphical use interface
3	ID	Identification card
4	MD5	Message digest algorithm
5	MYSQL	Microsoft Structural Query Language
6	PHP	Hypertext pre processor
7	SQL	Structured query language
8	UI	User interface
9	UML	Unified modeling language
10	XAMPP	Cross platform, Apache, MYSQL,PHP and Perl

Table x. 1: acronym, definition and abbreviation

ABSTRACT

The current Abafransua preparatory school management system is a manual system. The current system exhibits manual process different activates in a manual way, due to manual way of managing the system there are problems like, data losing, economical problem, security problem and other problem by considering the current problems in order to find out the current system problem we have use different mechanism like document analysis, we have interview schools teacher and directors by gathering the functional requirements from clients we have going to do the proposed system. Proposed system is developed with the aim of solving these for Abafransua preparatory school. We observe how the current system operates and how the manual system done based on this the proposed and also systems improves all this process and makes all the process automated. For analysis model we use case diagram, sequence and activity diagram. For system design we include deployment diagram, UI design, and component diagram, for the project that is going to be developed in general we going to web based system. In order to implement this computerized system, the proposed system uses HTML, CSS and java script as front end and PHP programming language and MYSQL as back-end.

CHAPTER ONE

1. INTRODUCTION

Education is a key factor to change or to decide the development of the country or the development of different technology or the discovery of new things, so in order to be effective teaching learning method there must be an effective and a well-organized system needed. Our country is one that gives great attention to develop quality of education is needed and maintains the quality of the service and amended a new policy and strategy to develop quality of education on the other hand, still problems are there in different schools. For example when we had gone Abafransua preparatory school has many problems are happen in the school and there were a huge problems in other that appear in different preparatory schools by this case we decided to select Abafransua preparatory school. Our project is normally designed to solve problems that appear in Abafransua preparatory school and our title is that Abafransua preparatory school management system used to simplify problems like losing of persistent data, insecurity, difficult to do different database query and student managing system is so difficult and others are the problem of the current system by observing and interview this problem we have to have initiate to do the proposed system and we have going to do the system.

1.1. Background of organization

Wolkite city is found in south nation nationality and people of Ethiopia found in south eastern from the capital city Addis Ababa and wolkite is 158 km far from the capital and Abafransua preparatory school found in Gurage zone and it located south eastern parts of Wolkite city around 11 kilometers far apart from its capital wolkite city. Abafransua preparatory school is founded in Gubre worda which is found one kilometer far from the center of the town. The school was formed on 1994 E.C. It consist both high school and preparatory school, but we are going to deal with preparatory one and we will do only preparatory management system. Vision of school is that students should be creative and knowledgeable in all dimension like morally and others and its mission of the school is that to be competitive from other schools and to develop schools in infrastructure and others. In the school there are a number of students that attend their normal education listed below.

Table 1.1: Number of employers and students both grade11 and 12

Member	Male	Female	Total
Student	553	335	888
Teacher	46	12	58
Record officer	1	1	2
Director	1	1	2

1.2. Statement of problem

Abafransua school employers do their activity normally in manual way of recording students information processing and manual analysis of information. The manual preparatory student management system is one that take more time and very tedious activity and unqualified, costly and not satisfactory. Due to this reason we expressed in the next the existing manual processing has the following tangible drawback or problems.

- Searching student file is difficult because of search all file until they get the necessary information of someone files. One has to look through all file to get a single student file. It is very boring, takes a lot of time and may be found at the last after they take very long time and also it is a very tedious for employees.
- File control mechanism is complicated and not secured.
- Lose of data; the data in the school can be lost through different reasons because it normally stored in manual way or paper work.
- Teachers cannot easily prepare student card
- Record officer cannot easily check transcript card
- Teachers cannot easily arrange student within a class
- Teachers cannot easily calculate student's average, rank and total mark because now they have done their activity in a calculator.
- Have not persistence database store student information for a long time.
- Difficult to do managing student wasting resource: loosing papers, color of printer and power supply are some of wasting resources.

1.3. Objectives of the Project

The aim of our project is that in order solve problems that found in Abafransua preparatory school and to gate tangible and measurable out come by using the given time.

1.3.1. General objective

The main objective of our project is to develop student management System for Abafransua preparatory school.

1.3.2. Specific objective

To overcome or achieve the general objectives of the project, the proposed system consists of the following specific objectives:-

- Study the existing system and find out the problem.
- In order to control data losing by using security system
- In order to simplify working condition of employers
- List out of the problem of the existing system
- Identification of system requirements and problems of the existing system.
- Data collection of the current system in order to get the problem of exiting system
- Design of the proposed system for the overall view of the structure
- Implementation of the proposed system
- Lastly testing the proposed system for weather system functionality fulfilled or not

1.4. Feasibility analysis of the project

Feasibility study is used to investigate the proposed system in multiple dimensions. It deals in different aspect of either the system is visible or not. There a number of analysis like we will discuss we have included technical, operational and economic feasibility

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. The proposed system can be easily maintained and repaired without requiring high professional experts or technical assistants, because the system was developed by familiar programming language (environment).There is record officer and information technology (IT) teachers who are practically trained by other application found in the Abafransua preparatory school or they know the other computer application. I.e. IT

teachers and record officers that already learn about the computer before they teach, students start to learn grade 9 up to now about the behavior of computer, So that the project can be said technically feasible.

1.4.2. Operational feasibility

Normally the users of the existing system are not happy because of many problems are found in the existing system like difficult to do different database system and data losing others are the problems of existing system in case of the above mentioned problems the client are need to be modified the current manual preparatory management system. This system bring better achievement for the operations performed by the school by providing efficient registration and storage of student information, easy updating ,deletion, modification etc. This intern increases the efficiency of work in the school and it also decrease the time to do their task properly and the client are happy to use the system in a sufficient condition. Do their activity effectively. So that one can say that the system is operationally feasible.

1.4.3. Economic feasibility

Normally the current system takes huge or take exceeded amount of budget because of it needs a hard materials and needs many employers to do the task. The project we developing will have the following benefits.

Tangible Benefits

The systems provided tangible benefit like:-.

- Minimizes registration cost
- Minimize demand of paper.
- Easily access student information throughout the school.

Intangible Benefits

- Minimizes the work load of the employee.
- Increased flexibility
- Increase speed of activity
- Improves the confidence of the system users.

1.5. Scope and limitation of the project

1.5.1. Scope of project

Our project concerned Abafransua preparatory school and it will applicable or its working area is in that school that is due to the rules and intention of different schools and requirements of each schools are different due to this case we will to do Abafransua school to fulfill the needs of the system. Our project we have included the main and necessary requirement by fulfilling the question of the client and our system perform from registration during registration payment also included to grading and students card generation and also examination and finally attendance generation and scheduling also others will perform in our system.

1.5.2. Limitation of our project

Normally we have not included very little functionality of our system because of different problems. The preparatory student management system only include preparatory grade 11 and 12. We will not include employee salary pay rolling, detail information about employee not include face recognition, digital signature and examination cheating and not included the discussion question only choose question are upload.

1.6. Significance of the Project

After this project had finished and properly used it gives great benefits to record officer, teacher, director and students of the school. Some of the significant that the team has identified are listed out as follows:

- Reduce manual process and administrative cost to maintain existing system.
- Reduce time, cost and redundancy of information or data.
- Improving efficiency, control and security of existing system.
- Fasten and facilitate the strategy of processing student mark
- Improve employee working condition and simplify employers activity
- Control data losing
- Easily store data
- Increase performance of the system
- Easily update, delete, search and other query system.

1.7. Beneficiary of the Project

- Teachers
- Record officers
- Students
- Directors

1.8. Methodology of the Project

There are many techniques of data collection. Among them the team used the following methods for collecting data. We have gone and collect data from director of Abafransua preparatory school and we have get a great feedback from director and record officer of the school.

1.8.1. Data Collection Tools

Normally we have used different mechanism in order to get a sufficient information about the existing system. They have already answered the question what we have questioned them. All in all we have used different ways from that listed below.

Interviewing: by asking the school administrator specially record officer and director. We choose this type because we can get detailed information about the current system.

Observation: By visiting, the school staffs while they are working. For our project we will observe 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 and we have also get the mission vision of and also Abafransua school when formed.

Document analysis: we have seen different documents that the school administrative being used for the current system. We will have going to analysis different system

1.8.2. Development Tools and Technologies

Tools used to design the system requirement documentation and implement.

Hardware

- HP Computer: 3.67GHzprocessor; 4.00 GB RAM, local disk 500 GB
- Printer
- Flash disk: 8GB San Disk
- Copy machine
- Others like papers, pen…etc.

Software

- Microsoft Office 2007/2010:
- Mozilla Firefox:
- Microsoft Visio:
- XAMPP control and WAMP Server
- Note pad++
- Edraw max
- MYSQL
- Cell phone camera
- Visual studio code
- Sublime and other software which are used to do our project

We have used different features of development tools and in front end and back end in order to do a well functional project.

1.8.3. Front end

In front end we have used such programming tools in front end we have used the following tools:

- HTML
- CSS
- JavaScript and others

1.8.4. Back end

We have used in back end:

- PHP
- XAMMP server
- MySQL database

1.8.5. System Analysis and Design

Normally we have going to use different methods to design and analysis our system we will design the system using different diagram like class diagram, component diagram, deployment, persistent diagram, and others. For analysis method we use case model, sequence diagram, activity diagram, user interface prototype and others.

1.8.6. System Development Model

Normally we will use iterative model the reason why we use this model type is that in order to add features on every stage.

1.8.7. System Testing Methodology

Unit testing: testing mechanism that used to test our system each and individual component.

Integrated testing: use to test whether the system or components are meets or integrated.

System testing: used to test whether the overall system fully functional or it include the overall requirement or not.

1.8.8. Deployment Environment

After we will develop the system will deploy in Abafransua preparatory school in order to manage students and student information.

CHAPTER TWO

2. DESCRIPTION OF THE EXISTING SYSTEM

2.1. Introduction of Existing System

Preparatory school follows after the completion of primary and secondary school or high school. At the beginning of each academic year which starts in September (Ethiopian New Year), the students get registered and assigned rooms. Each class (section) of students is assigned to a fixed room. Home room teachers are assigned to each class of students. There are two semesters per year. The first semester final examination is usually administered during January, the second semester final examination is administered during the end of June and consequently the results of each class of students is collected, organized, ranked by the corresponding home room teacher and reported to each student. The homeroom teacher also records attendance of each student on each school day which is later organized by the attendance officer. A student who has been absent for more than twenty days is not allowed to take a semester final examination and will be forced to withdraw. Transcripts are generated by the record officer. A student may request transcript when he/she wants to transfer to other school or when he/she has completed from the school and needs to join higher education or for some other purpose.

2.2. Users of Existing System

Users of the current system or existing system are listed below.

- **Director:** person who can post school information in manually or orally, arrangement of students in their class and scheduling, view comment and attendance.
- **Teacher:** a person who evaluate and who can process student mark, prepare report card, prepare exam, register students and give comment to student orally.
- **Student:** A person who can register to the school, see his/her result, receive transcript and card, payment during registration and take exam.
- **Record officer:** a person who keeps record of students and prepare transcript and card.
- **Home room teacher:** a person who view student attendance
- **Parent:** a person who can meet on school and view their student.

2.3. Major Functions of the Existing System

Even if we say the current system is manual system the existing system perform the following major activities.

- Registration of students
- Distribute Report card for student
- Maintain records of student's mark in each course.
- Preparing report card of students
- Officers accept marks from the teacher.
- Arranging of students in their class
- Examination generation
- Distribute transcript and report card
- Exam scheduling
- Class scheduling
- Calculating total rank average and others in other way grading system
- Attendance generation for both student and teacher.

2.4. Drawbacks of the Existing System

2.4.1 Performance

The system is normally manual and the performance of the existing system does not provide fast response time because it is difficult to access data from the stored document. And also, it is slow /time and energy consuming.

2.4.2 Security and Controls

Every record of the existing management System is stored in the manual way, so, it is difficult to control and secure due to data and record are stored in a manual, since it doesn't have any authentication and authorization system. Data and files are easily lose due to lack of authorization and authentication.

2.4.3 Input and output accuracies

In the existing of Abafransua preparatory school Student management System the input of information is redundant, inaccurate, not well organized and inflexible and also these inputs may

leads to create confusion and work load on employees and produce inaccurate output since the information is written by pen and on paper.

2.4.4 Economics

Due to the operation that is done by the hand most of the activities are causes to high consumption of resources like papers, man power, time etc. This makes the existing system costs are too high.

2.4.5 Data storage problem

Due to store in a manual way in order to found out the data there were a huge problems to access it or find the data.

- Data are not easily accessible due to place in different location in the shelf.
- Difficult to change and edit.
- Data redundancy that leads to inconsistency
- Time consuming
- Consumes more man power
- Needs more manual calculation

2.5. Business Rules of the Existing System

The Business Rules that are involved in the current student information system:-

- Any student that full fills the registration; they must pass the school exam.
- The students must be pay 300 Birr and above 350 Birr as a costs share during they are register.
- Any student who is not present 3 days and above in a month may not sit on examination. But their parent come to the school and gives reason by what case he is present.
- If student absent for more than one week consequently, they will must bring a letter from the responsible organization.
- Each student must wear school uniform when they come to the school.
- Any students are not learning without their class.
- Any student who cheat exam and signed by exam teachers, then the students get 0 in that course.

CHAPTER THREE

3. PROPOSED SYSTEM

Our project Abafransua preparatory school management system designed to solve problems in that school. Used to solve problems like, losing of data, database that used to store student information, making grade in our system and security problems. To design and analysis we have use object oriented software like unified modeling language like, state chart ,sequence diagram, activity diagram, deployment diagram ,component diagram, class diagram and use case diagram. For implementation we have also use different programming language like, cascading sheet, hypertext markup language java script for front end and for back end we have used PHP, MySQL server or XAMPP server. The proposed system is very easy to operate and easily handle all the data and the work done by the existing systems. The main objective of the proposed system is:-

- It minimizes the task load of the school
- It will minimize wastage of hard copy materials like paper and record book.
- It is time efficient to do activities
- It decrease the work load of employees

It will illuminate some of the physical activities of the office such students can see their transcript and report card without having physical contact with the school.

3.1. Functional requirements

The functional requirement for the system describes the functionally or services that the system is expected to provide. Functional requirement normally the way to express what are done our system and what is the role of the system what actions or activates must done in our system. What are the needs of clients and normally done the requirement gathering. Functional requirement for a software system may be expressed in a number of different student management systems.

The functional requirement we have done are listed below:

- Enable students to see comments feed backed from teacher.
- Allow student and parents to view student grading card securely.

- Enable record officer search the student file from the database
- Allow record to registration student.
- Record officer to prepare transcript.
- The director enables to post new information.
- Attendance generation
- Teachers generate examination
- Generate class scheduling
- Generate exam scheduling
- Login system
- Payment system
- Marking system (calculating grades or student mark).
- Parents give comment to teachers
- Parent view grade their student (their son's or daughter grade).

3.2. Nonfunctional requirement

Non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of the system, rather than specific behaviors of the system which are specified under functional requirements of the system it shows the system is attractive and well function and other. The following are considered as the non-functional requirements for the project we are going to develop.

3.2.1. Security

Any information system must be secure to fulfill the three main components of system security which are confidentiality, integrity and availability, therefore to make the system that we are going to develop secure, we are going to employ authentication mechanism which is giving the user username and password so that no one apart from the authenticated user can get access to the system and our system have authentication and authorization system in order to make our system secure and not attacked by any or different crackers and in order to control data losing and users have password to access privilege to our system and have flexible password in order to control and we have use MD5 encryption algorithm.

3.2.2. Response time

The system should give the right response for the user as fast as possible, since the system we are develop web based system it will not be sophisticated to use and it will not take much time to handle responses. Our system will give the response through a short time and it is so fast to give feedback.

3.2.3. Usability

After the proposed system is implemented it will be usable by most of schools since it will be easy and friendly to use it is believed to be usable and it will use for a long period of time.

3.2.4. Error Handling

When a user interacts with the system errors may occur. To control this kind of inaccuracies system will generate different user friendly messages. To do this, most of the system execution buttons will be controlled according to the sequence which the user is expected to follow, or this can be done by generating different system responses to the input of the user. By using different programming tools we able to solve the error like fatal error and other different errors by using programming tools like JavaScript.

3.2.5. User Interface and Human Factor

The system we are going to develop will have a user friendly graphical user interface (GUI) which allows users to interact with the system easily. The users can easily contact to our system because of users have some know how about computer knowledge and it will easily contact the system. Users of the system are teachers, students, directors and record officer because of the listed user the user is expected to have knowledge. Generally it includes the necessary features for each user friendly and interactive interface the system should have to provide interactive and easily manageable user interface for the user.

3.2.6. Efficiency

In existing system, there is no efficiency because all activities performed manually in the paper. It can reduce the efficiency of the system. As we observed in existing system to perform some action, it takes more time and many resources and it results economic problem but our system solve the efficiency of the system.

3.2.7. Resource issue

In our project it needs a desktop computer and a skill person who can easily interact to the system and get a great feedback all in all resources needed to computer and others. The system consumes high processor speed and memory for both client and server side this enables the system to provide in efficient and effective manner in addition of the mentioned above the system will be best provision and become consistent.

3.2.8. Hard ware consideration

Needs a hardware and like ram, desktop computer, hard disk and the system or our project will deployed by the existing computer and the system is portable it will run by any type of computer and it support any type of run browser.

3.2.9. Physical environment

The system will installed in the server side script in but for more feature we recommended that the system deployed that is free from natural and human disaster.

3.2.10. Backup and recovery

Storing data in another place for backup purpose. If the system will lose some data it is easy to get the loosed data. it can be easy to be placing data its data. If the failed or lost it can be recovered from database easily.

3.2.11. Performance consideration

Performance since the system is going to accessed by different user with different needs, it should be capable handling and process query. Generally the system should be able to handle many users and easily responsive. The system should be responsive fast in order to access the required information. It able to get or run on a short time.

3.2.12. Quality issue

Availability the system should available at any time and reliability the system should control any type of crash during run time since users can use simultaneously. Validation for input user that must put the exact value to the provided system be done in our system and it avoid incorrect storage record.

3.2.13. Documentation

Documentation helps or guide a team to make knowledge management and used for users to guide how the system operate. There for it is a necessary requirement and it helps project

maintenance purpose. The documentation or final project contains proposal, report and final documentation. The documentation is well defined and written a well-organized way and it also reflect what have done in the implementation system. It also helps how it functional and a process of what activity done the system. Show the process how users use the system also shows how errors and other problems be solved and how system will give feedback to client or users.

CHAPTER FOUR

4. SYSTEM ANALYSIS

4.1. System model

Object oriented Analysis combines data and the processes that act on the data into things called objects. Object oriented system development approach is Integrates easily with object-oriented programming languages. Easy to maintain and expand because new objects can be cloned using inherited properties.

Object oriented system development approach gives easier way to break down problems into Simple and small components so that it reduces the vague appearance of the big problem. The major activities covered in this chapter are constructing a use case model, documenting the use case course of events, constructing sequence and activity diagrams and user interface prototype about the proposed system in general we have use different object object-oriented diagrams like system model use case diagram and object model class diagram and data directory and finally dynamic model like state chart sequence and activity diagram in order to represent our system.

4.1.1 Use case diagram

Use case modeled normally composed of a use case diagram and accompanying documentation describing use cases actors and users that participate in deploying the system and who have done which activity will done in our system. Also a use case representation the system intended to functions and its environment. The functionality are specified by the use case and actors who participate on the system and on its environment.

Use case is the identification and representation of a sequence of that users or actors takes the particular target to the system. It has represented by elliptical with descriptive name and its description.

An actor is a person, system that participate or have a role in one or more interaction in to the system relationship between actor and class are expressed in use case diagram. A relationship exists whenever an actor is involved in the interaction described by use case. The purpose of an actor analysis is to identify all of the actors that interact to the system. An actor have a role that interact the system and do its functionality and the necessary things that the system do. Generally we have four actors that participate in the system they are:-

- Director

- Teachers
- Record officer
- Students
- Parents
- Administrator
- Home room teacher
- Cashers

We have put in the way of their activity they have the following role in the system Use Case represents interaction between the user and the system. The following use cases have been identified from the system specification.

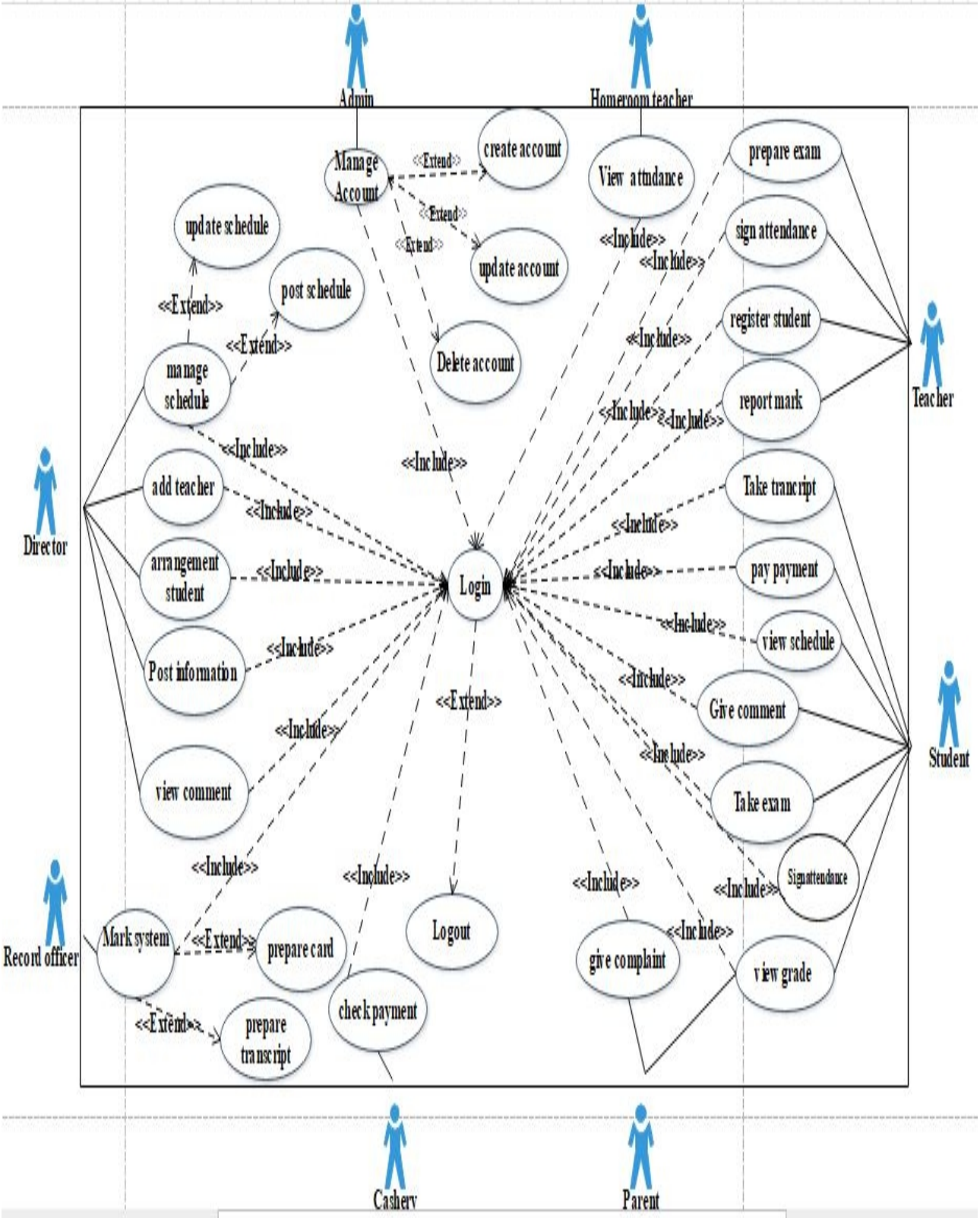


Figure 4. 1: use case diagram

4.1.2 Use case description

Table 4. 1: use case description manage user account

Use case name	Manage user account
Use case id	01
Use case description	Enables system admin to manage users to use the system
Actors	Administrator
Precondition	The administrator have a username and password before login to the system.
Post condition	The users will create account to the system and do their activity with their own interface.
Flow events	Step1, The users want to login. Step2, The system displays login page for users. Step3, The user enters its username and password to the system. Step4, The system check the validity of user name & password Step5, The systems display the appropriate page. Step6, The use case end.
Alternative flow events/course of action	If the users enters incorrect user name and password. 1: The system display error message. 2: The use case continuous at step3.

Table 4. 2: use case for description of login

Use case name	Login
Use case id	02
Use case description	Enables to log to system and use system for different purpose.
Actors	Director, Record officer, home room teacher, parents, administrator, Teacher and Students
Precondition	The administrator, teacher, student and record officer have a username and password before login to the system.
Post condition	The users will login to the system and do their activity with their own interface.
Flow events	Step1, The users want to login. Step2, The system displays login page for users. Step3, The user enters its username and password to the system. Step4, The system check the validity of user name & password Step5, The systems display the appropriate page. Step6, The use case end.
Alternative flow events/course of action	If the users enters incorrect user name and password. 1.1: The system display error message. 1.2: The use case continuous at step3.

Table 4. 3: use case for description of post information

Use case name	Post information
Use case id	03
Use case description	Enables director to add necessary information to the system and to post information to members of the system.
Actors	Director
Pre-condition	Should have a legal account (username and password).
Post condition	Director successfully post information.
Flow events	Step1, The Director wants to post information to the system. Step2, The Director first login to the system by their own username and password. Step3, The system check that person is legal or not to post information. Step4, The system displays post information page for admin if he/she is legal. Step5, The Director post information to the system the targeted member. Step6, The Director logout from the system Step7, The use case end.
Alternative flow events/course of action	If Director enter wrong user name and password. 2.1: The system display please enters a correct user name and password. 2.2: The use case continuous to step2

Table 4. 4: use case for description of add teachers

Use case name	Add teachers
Use case id	04
Use case description	Enables the director to add teachers
Actors	Director
Pre-condition	Should have a legal account (username and password).
Post condition	Director successfully add teacher.
Flow events	Step1, The Director wants to add teacher to the system. Step2, The Director first login to the system by their own username and password. Step3, The system check that person is legal or not to add teacher. Step4, The system displays add teacher page for admin if he/she is legal. Step5, The Director add teacher to the system. Step6, The Director logout from the system Step7, The use case end.
Alternative flow events/course of action	If Director enter wrong user name and password. 1: The system display please enters a correct user name and password. 2: The use case continuous to step2

Table 4. 5: use case for description of arrange student

Use case name	Arrange student
Use case id	05
Use case description	Enables directors to arrange students in their room due to their limit
Actors	Director
Pre-condition	Should have a legal account (username and password).
Post condition	Director successfully arrange student.
Flow events	Step1, The Director wants to arrange student to the system. Step2, The Director first login to the system by their own username and password. Step3, The system check that person is legal or not to arrange student. Step4, The system displays arrange student page for admin if he/she is legal. Step5, The Director arrange student based on their name alphabet. Step6, The Director logout from the system Step7, The use case end.
Alternative flow events/course of action	If Director enter wrong user name and password. 1: The system display please enters a correct user name and password. 2: The use case continuous to step2

Table 4. 6: use case for description of prepare transcript

Use case name	Prepare transcript
Use case id	06
Use case description	Enables record officer to prepare student transcript
Actors	Record officer
Pre-condition	Record officer Should have a legal account (username and password).
Post condition	Record officer successfully prepare transcript
Flow events	Step1, The record officer wants to prepare transcript. Step2, The record officer first login to the system by their own username and password. Step3, The system check that person is legal or not to prepare transcript. Step4, The system displays prepare transcript page for user if he/she is legal. Step5, The record officer by calculating sum for both grade 11 and 12 and average for both and finally rank different query. Step6, The record officer logout from the system Step7, The use case end.
Alternative flow events/course of action	If record officer enter wrong user name and password. 5.1: The system display please enters a correct user name and password. 5.2: The use case continuous to step2

Table 4. 7: use case for description of view comment

Use case name	View comment
Use case id	07
Use case description	Enables Director, student and teacher to view comments commented from others
Actors	Director, student and teacher
Pre-condition	Should have a legal account (username and password).
Post condition	Director, student and teachers should see their comment successfully.
Flow events	Step1, The Director, teacher and student wants to view comments to the system. Step2, The Director, student and teacher first login to the system by their own username and password. Step3, The system check that person is legal or not to view comment. Step4, The system displays view comment page for admin if he/she is legal. Step5, The Director, teacher and student view comment by inserting essential information to the system. Step6, The Director, student and teacher logout from the system Step7, The use case end.
Alternative flow events/course of action	If Director, teacher and student enter wrong user name and password. 6.1: The system display please enters a correct user name and password. 6.2: The use case continuous to step2

Table 4. 8: use case for description of prepare exam

Use case name	Prepare exam
Use case id	08
Use case description	Enables the teacher to prepare exam.
Actors	Teachers
Pre-condition	Should have a legal account (username and password).
Post condition	Teacher successfully prepare exam.
Flow events	Step1, The teacher wants to prepare exam to the student. Step2, The teacher first login to the system by their own username and password. Step3, The system check that person is legal or not to prepare exam student. Step4, The system displays prepare exam page for admin if he/she is legal. Step5, The teacher prepare exam by inserting essential information to the system. Step6, The teacher logout from the system Step7, The use case end.
Alternative flow events/course of action	If teacher enter wrong user name and password. 1: The system display please enters a correct user name and password. 2: The use case continuous to step2

Table 4. 9: use case for description of view attendance

Use case name	View attendance
Use case id	09
Use case description	Add necessary information to the system or exam to the system.
Actors	Home room teacher
Pre-condition	Should have a legal account (username and password).
Post condition	Home room teacher successfully view attendance page.
Flow events	Step1, The home room teacher wants to view student attendance. Step2, The home room teacher first login to the system by their own username and password. Step3, The system check that person is legal or not to view student attendance. Step4, The system displays view student attendance page for admin if he/she is legal. Step5, home room teacher will see or view student attendance on the system. Step6, home room teacher finally logout from the system after view student attendance. Step7, The use case end.
Alternative flow events/course of action	If room teacher enter wrong user name and password. 1: The system display please enters a correct user name and password. 2: The use case continuous to step2

Table 4. 10: use case for description of sign attendance

Use case name	Sign attendance
Use case id	10
Use case description	Add necessary information to the system or attendance to the system.
Actors	Teachers and students
Pre-condition	Should have a legal account (username and password).
Post condition	Teacher and student successfully sign attendance.
Flow events	Step1, The teacher and students wants to sign attendance. Step2, The teacher and student first login to the system by their own username and password. Step3, The system check that person is legal or not to sign attendance. Step4, The system displays sign attendance page for user if he/she is legal. Step5, The teacher and student sign attendance by inserting essential information to the system. Step6, The teacher and student logout from the system Step7, The use case end.
Alternative flow events/course of action	If teacher and student enter wrong user name and password. 1: The system display please enters a correct user name and password. 2: The use case continuous to step2

Table 4. 11: use case for description of take exam

Use case name	Take exam
Use case id	11
Use case description	Enables student to take exam
Actors	Student
Pre-condition	Should have a legal account (username and password).
Post condition	Student has successfully take exam
Flow events	Step1, The students wants to take exam. Step2, The student login to the system by their own username and password. Step3, The system check that person is legal or not to prepare card. Step4, The system displays examination page for user if he/she is legal. Step5, The students reads the exam questions and select which he or she realized to the correct answer and click the submit button. Step6, The record officer logout from the system Step7, The use case end.
Alternative flow events/course of action	If Student enter wrong user name and password. 1: The system display please enters a correct user name and password. 2: The use case continuous to step2

Table 4. 12: use case for description of prepare card

Use case name	Prepare card
Use case id	12
Use case description	Enables to prepare student card
Actors	Record officer
Pre-condition	Should have a legal account (username and password).
Post condition	Record officer successfully prepare card.
Flow events	Step1, The record officer wants to prepare card. Step2, The record officer first login to the system by their own username and password. Step3, The system check that person is legal or not to prepare card. Step4, The system displays prepare card page for user if he/she is legal. Step5, The record officer by inserting essential information to the system and doing different calculation like sum, average and rank. Step6, The record officer logout from the system Step7, The use case end.
Alternative flow events/course of action	If record officer enter wrong user name and password. 1: The system display please enters a correct user name and password. 2: The use case continuous to step2

Table 4. 13: use case for description of view result

Use case name	View result
Use case id	13
Use case description	Enables to parents to see their sons or daughters result and students also view their grade result
Actors	Parents and students
Pre-condition	Both student and parent should have a legal account (username and password).
Post condition	Both parents and student successfully view grade
Flow events	Step1, both students and parents want to view result or view grade. Step2, both students and parents should login to the system by their own username and password. Step3, The system check that person is legal or not to prepare card based on their given password. Step4, The system displays view result page for user after they passed login part. Step5, the students and parents see or view their result. Step6, The parent and student logout from the system Step7, The use case end.
Alternative flow events/course of action	If both students and parents enter wrong user name and password. 1: The system display please enters a correct user name and password. 2: The use case continuous to step2

Table 4. 14: use case for description of check payment

Use case name	Check payment
Use case id	14
Use case description	Enables chaser to check payment
Actors	Cashers
Pre-condition	Should have a legal account (username and password).
Post condition	Cashers properly check weather students pay payment or not.
Flow events	Step1, The cashers wants to check payment. Step2, The casher first login to the system by their own username and password. Step3, The system check that person is legal or not to prepare card. Step4, The system displays payment page for user if he/she is legal. Step5, the casher check whether the student pay payment or not Step6, cashers logout from the system Step7, The use case end.
Alternative flow events/course of action	If the cashers enter wrong user name and password. 1: The system display please enters a correct user name and password. 2: The use case continuous to step2

4.1.3. Use case scenario

1. Scenario name:-**See comment** first teachers and directors must login in to the system through authorized user name and password. If teachers and directors wants to see comments which is forward from others click comment link page and then comment page then displayed and finally see comments successfully.
2. Scenario name:- **Create account** first administrator login in to the system and the system display admin page click account link and the system display the account form and finally administrator fill correct information and the system display successfully create account.
3. Scenario name:-**Register student** if director wants to registered first director must full fill the passing criteria and then log to the system and the registration page displayed and registered by teachers finally the system display successfully registered.

4. Scenario name:-**prepare exam** if teacher wants prepare exam to the student on the system. First teacher login in to the system and click the teachers page link and the teachers click then the displays the prepare exam and the teacher prepare the exam and finally successfully prepare exam.

4.2. Object model

In this section we have discussed overview of object model, especially class diagram. Class diagram is a static diagram. It represents the static view of an application it describes the attributes and the operations of class and imposed by the system. The class diagrams are widely used to modeling object oriented system because they are UML diagram, which can be mapped directly with object oriented languages.

4.2.1. Class diagram

A class diagram shows a collection of class, interfaces, association, and collaboration and constraints. Used to a describe structures of the system in terms of class. Class diagram used to describe the system in terms of objects, attributes, functions, class, operations and association with their relationship. In object oriented real world concepts are modeled in objects. Conceptual modeling hereby allows to model this concept which latter involve in to full-fledged class model. A class is a set of objects that share a common structure and a common behavior (the same attribute, operation, relationship and semantics).

A class is an abstraction of real world. When this items in real world they are instance of the class and are referred to as an object. Class are represented by rectangles with three sections. The top section represented or indicate a class and the middle section indicates attributes which stores information about items. The bottom sections shows or express methods of a class done on the objects. The class diagram below shows the class of our Abafransua preparatory school management system with their relationship (including inheritance and association) and their operation and attributes their class in the system Abafransua preparatory school management system there are different classes and relationships and attributes and operation or functions.

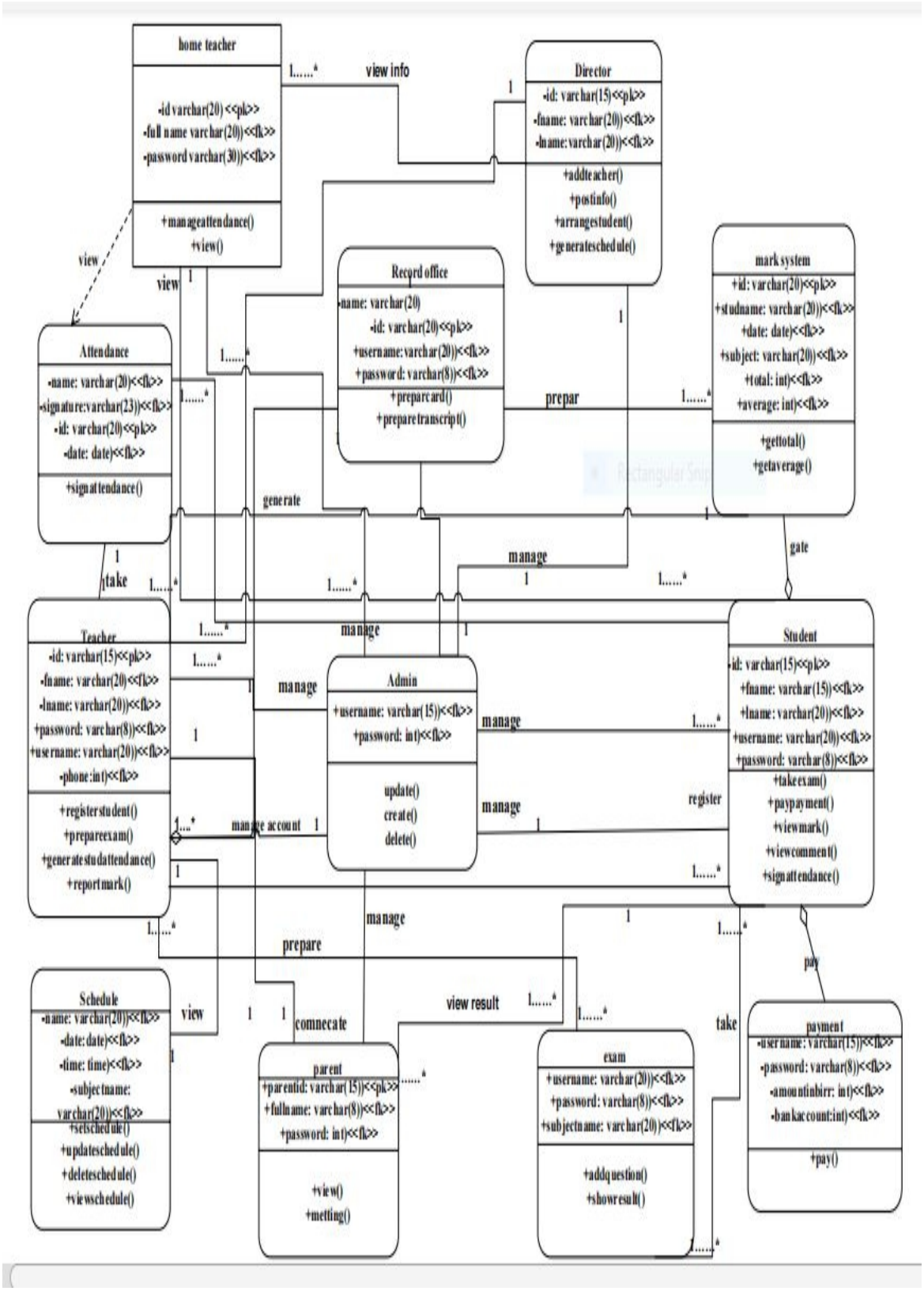


Figure 4. 2: class diagram

4.2.2. Data Dictionary

Data dictionary normally a table that contains tables of attributes data type filed size filed name and its constraints that are necessarily put in a database.

Table 4. 15: data dictionary of login

key constraint	Field name	Caption	Data type	Field size
Primary key	Login id	Login id	Varchar	50
Not null	User name	User name	Varchar	50
Not null	Password	Password	Varchar	50

Table 4. 16: data dictionary of post information

key constraint	Field name	Caption	Data type	Field size
Primary key	Post Id	Post Id	Varchar	50
Not null	User name	User name	Varchar	50
Not null	Phone no	Phone no	Int	50
Not null	Message	Message	Varchar	100

Table 4. 17: data dictionary of prepare card

Key constraint	Field name	Caption	Data type	Field size
Primary key	Student id	Student id	Varchar	50
Not null	Student name	Student name	Varchar	50
Not null	Semester	Semester	Varchar	10
Not null	Grade	Grade	Int	10
Not null	Year	Year	Year	10
Not null	Subject	Subject	Varchar	20
Not null	Average	Average	Int	10
Not null	Rank	Rank	Int	5
Not null	Total	Total	Int	5

Table 4. 18: data dictionary of class schedule

Key constraint	Field name	Caption	Data type	Field size
Not null	Course name	Course name	Varchar	30
Not null	Room	Room	Varchar	10
Not null	Start time	Start time	Time stap	10
Not null	Section	Section	Varchar	10

Table 4. 19: data dictionary of add teacher

Key constraint	Field name	Caption	Data type	Field size
Primary key	Teacher id	Teacher id	Varchar	30
Not null	Teacher name	Teacher name	Varchar	30
Not null	Age	Age	Int	5
Not null	Sex	Sex	Varchar	10
Not null	Phone no	Phone no	Int	20
Not null	Status	Status	Varchar	30

Table 4. 20: data dictionary of register student

Key constraint	Field name	Caption	Data type	Field size
Primary key	Student id	Student id	Varchar	30
Not null	Full name	Full name	Varchar	70
Not null	Age	Age	Int	7
Not null	Sex	Sex	Varchar	10
Not null	Promote	Promote	Varchar	30
null	Phone no	Phone no	Int	50

Table 4. 21: data dictionary of payment of student

Key constraint	Field name	Caption	Data type	Field size
Primary key	Student id	Student id	Varchar	20
Not null	Student name	Student name	Varchar	50
Not null	Bank account	Bank account	Int	30

Not null	Date	Date	Date	40

Table 4. 22: data dictionary of attendance of student

Key constraint	Field name	Caption	Data type	Field size
Primary key	Id	Id	Varchar	40
Not null	Full name	Full name	Varchar	70
Not null	Date	Date	Date	40
Not null	Attended	Attended	Varchar	40
Not null	Section	Section	Varchar	50

Table 4. 23: data dictionary of manage of account

Key constraint	Field name	Caption	Data type	Field size
Primary key	Id	Id	Varchar	50
Not null	User name	User name	Varchar	50
Not null	Password	Password	Varchar	20
Not null	Full name	Full name	Varchar	50
Not null	Age	Age	Int	10
Not null	Sex	Sex	Varchar	10
Not null	Status	Status	Varchar	20
Null	Phone no	Phone no	Varchar	30
Not null	Email	Email	Varchar	40
Not null	address	Address	Varchar	40

Table 4. 24: data dictionary of parent

Key constraint	Field name	Caption	Data type	Field size
Primary key	Parent_id	Parent_id	Varchar	40
Not null	First_name	First_name	Varchar	50
Not null	Last_name	Last_name	Varchar	50
Not null	Address	Address	Varchar	50
Not null	Phone_no	Phone_no	Varchar	40

Not null	Sex	Sex	Varchar	20

Table 4. 25: data dictionary of take exam

Key constraint	Field name	Caption	Data type	Field size
Primary key	Exam_id	Exam_id	Varchar	30
Not null	Subject _name	Subject _name	Varchar	50
Not null	Exam name	Exam name	Varchar	30

4.3. Dynamic model

The dynamic model represents time dependent aspect of the system. It is concerned with the temporal changes the state of an object in the system.in this section we will describe the dynamic behavior of the system in terms of activity, sequence and state chart diagram.

4.3.1. Sequence diagram

Sequence diagram is used to depict graphically how objects with each other via messages in the execution of use case or operation. They illustrate how the operation are performed between objects and in what sequence. A sequence diagram is an interaction diagram that shows how process operate with one another and in what order. It is a construct message of sequence chart, this sequence diagrams object interaction arranged in time sequence. It depicts the objects and classes involved in a scenario and the sequence of message exchange between the objects needed to carry out the functionality of the system. The following diagram shows the series of actions for the respective action. And also shows the life line of an object that participate in the sequence diagram or the system in the system Abafransua preparatory school management system sequence of actions are there to perform each task.

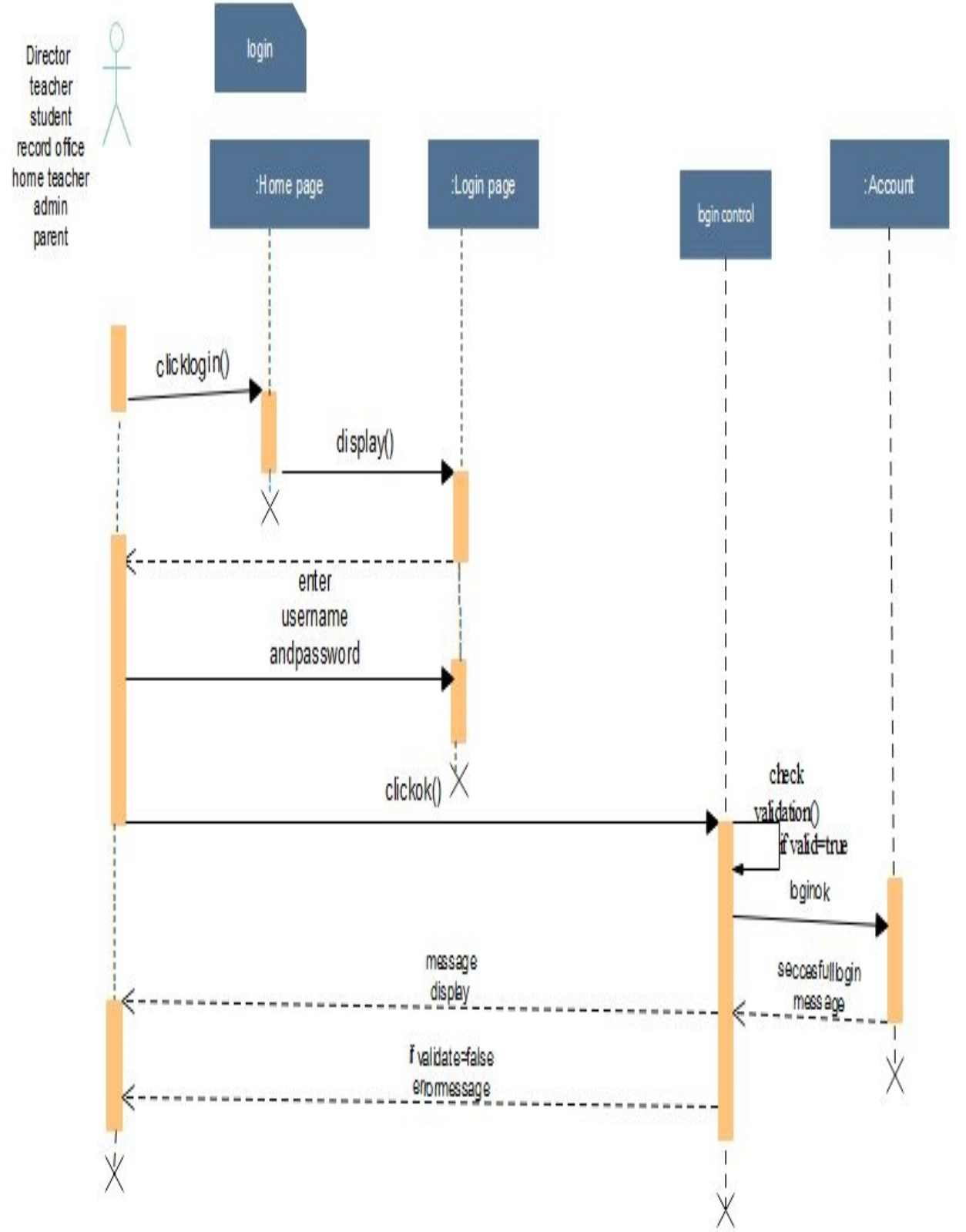


Figure 4. 3: login sequence diagram

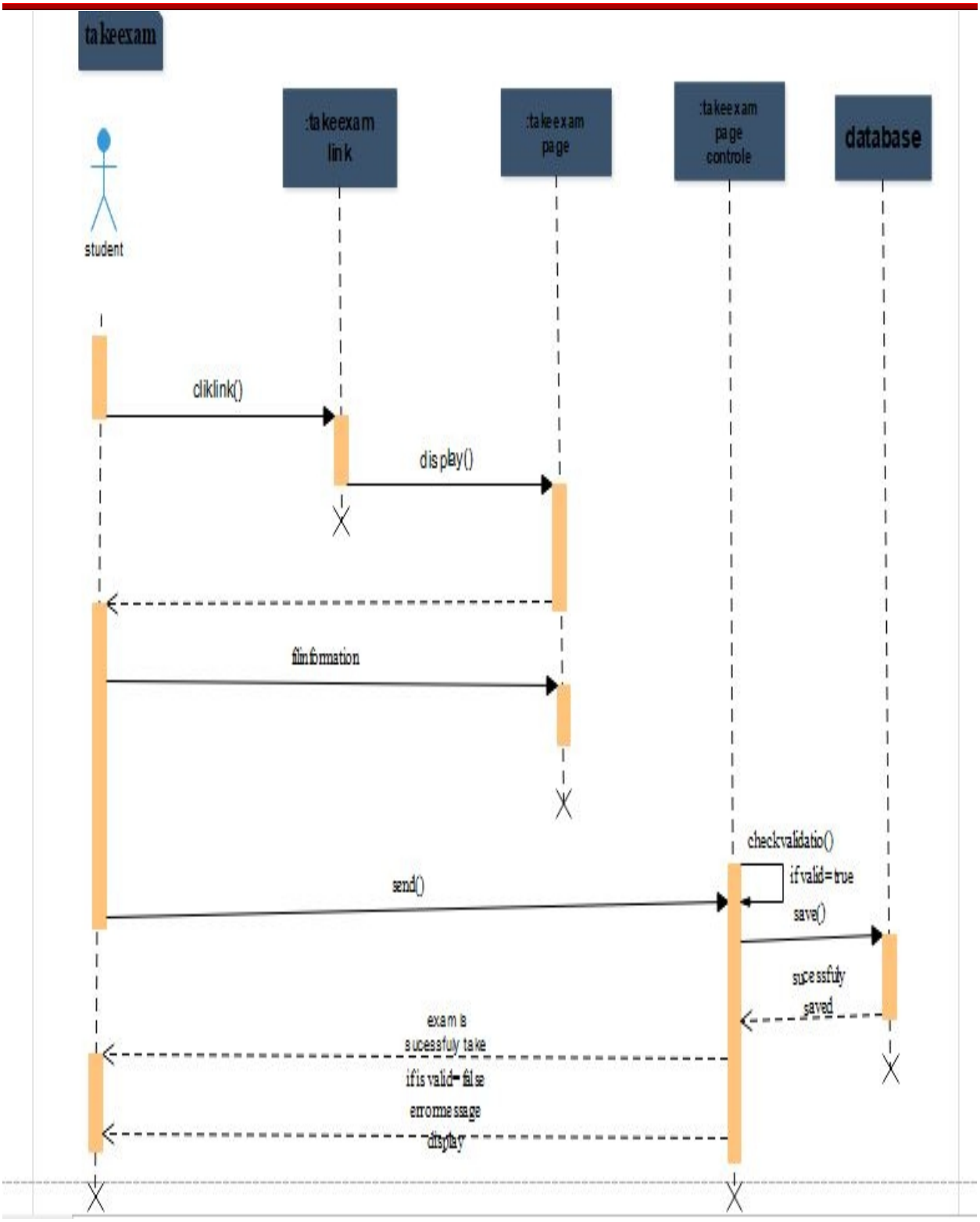


Figure 4. 4: take exam sequence diagram

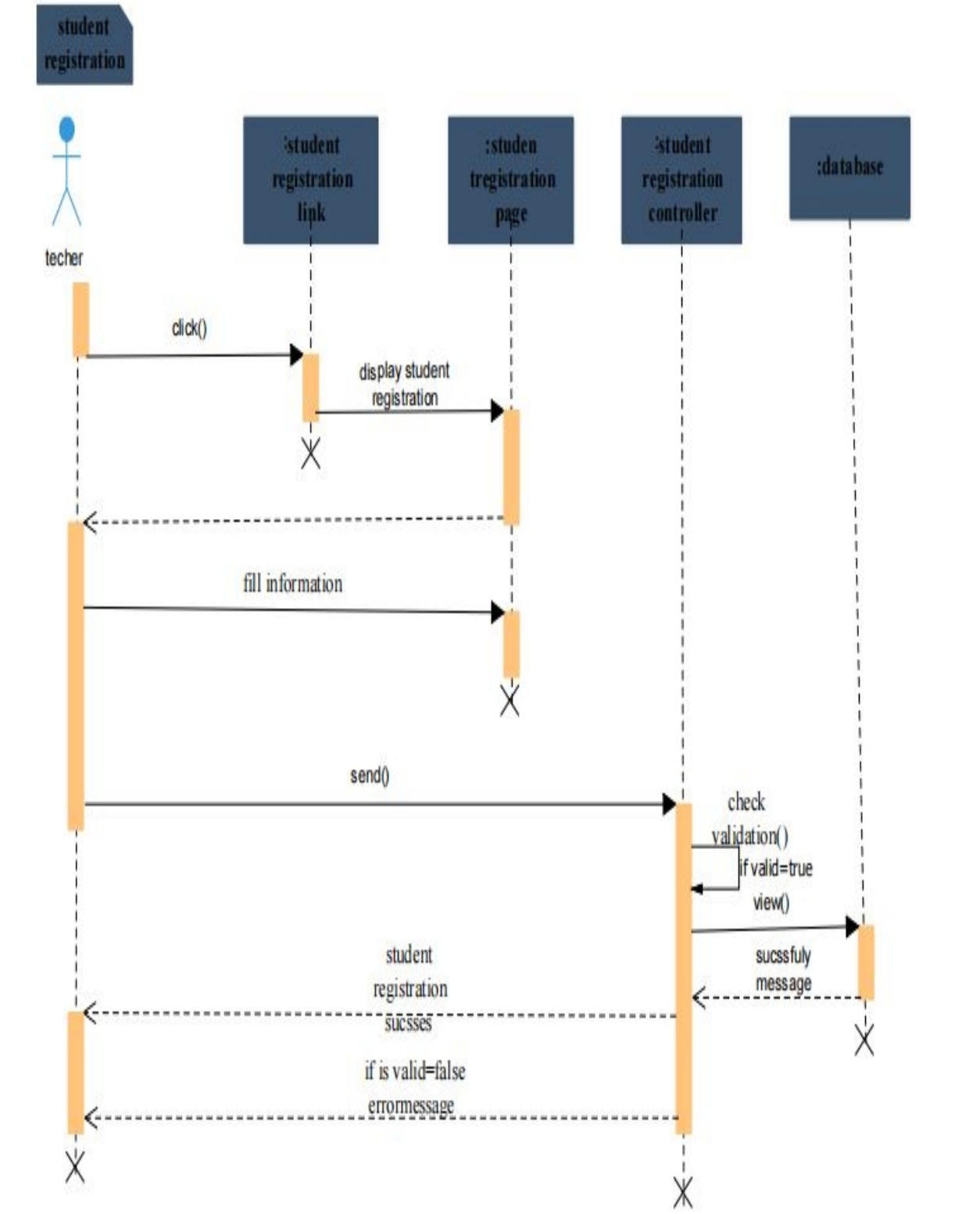


Figure 4. 5: student registration of sequence diagram

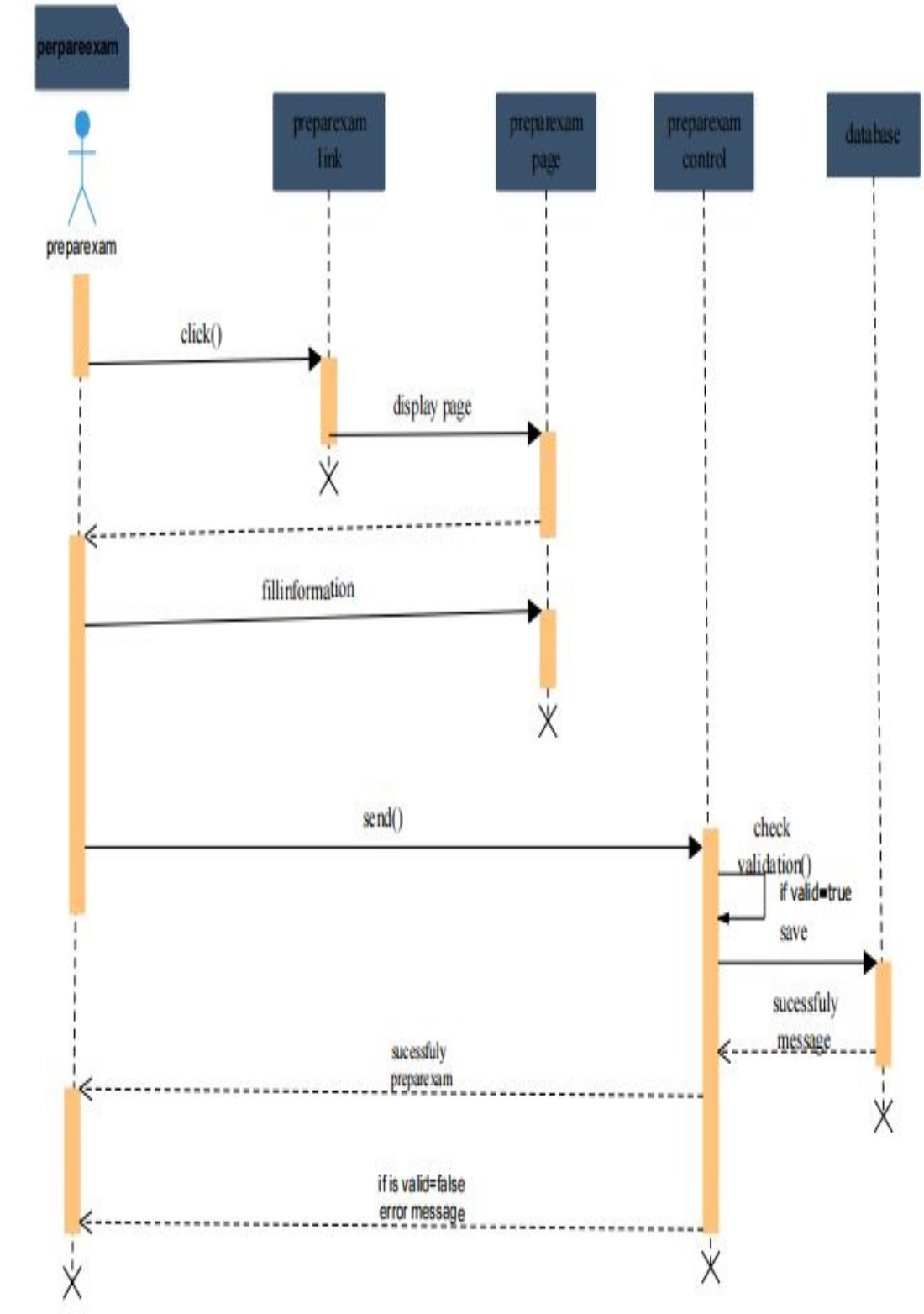


Figure 4. 6*: prepare exam sequence diagram*

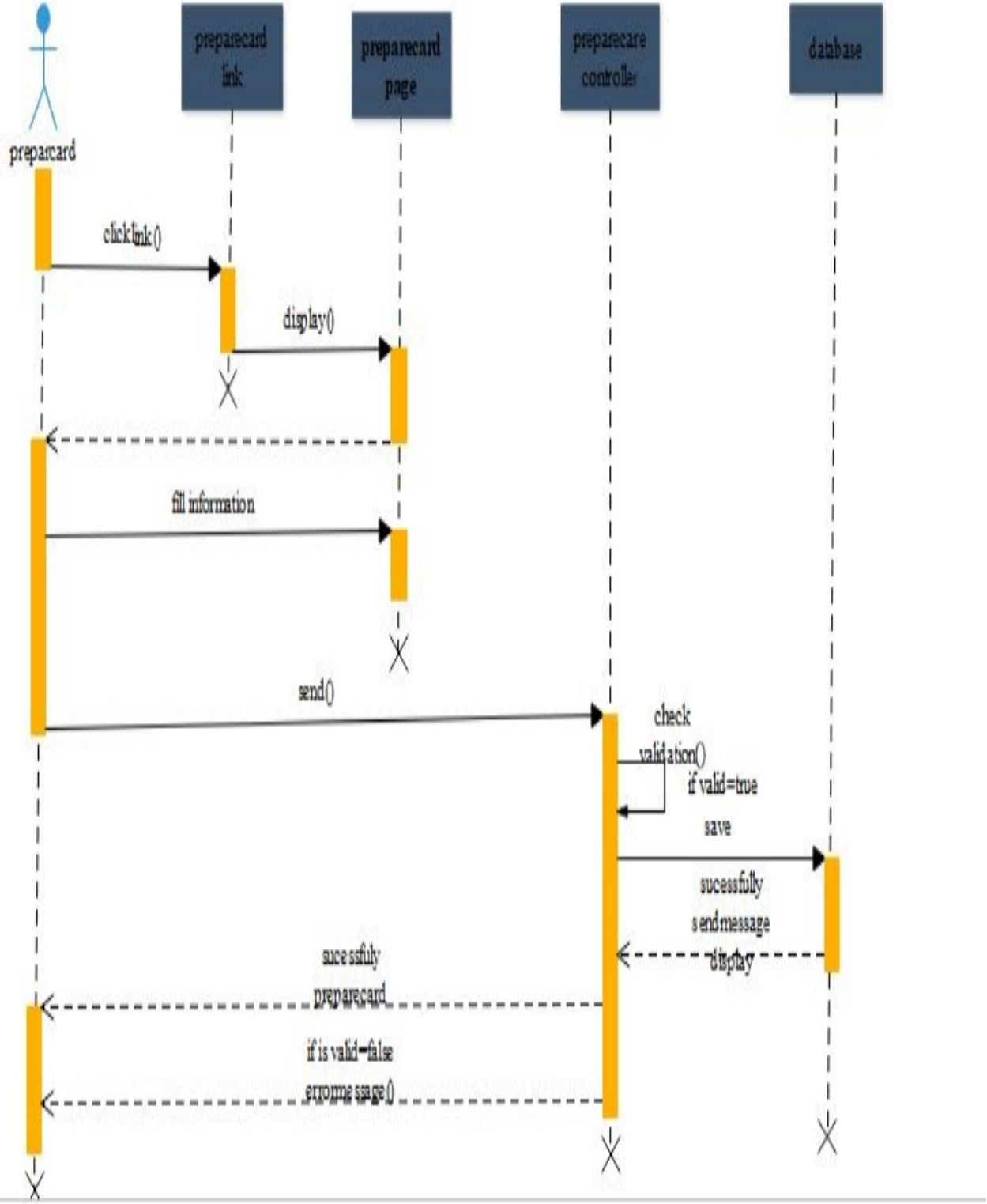


Figure 4. 7: prepare card of sequence diagram

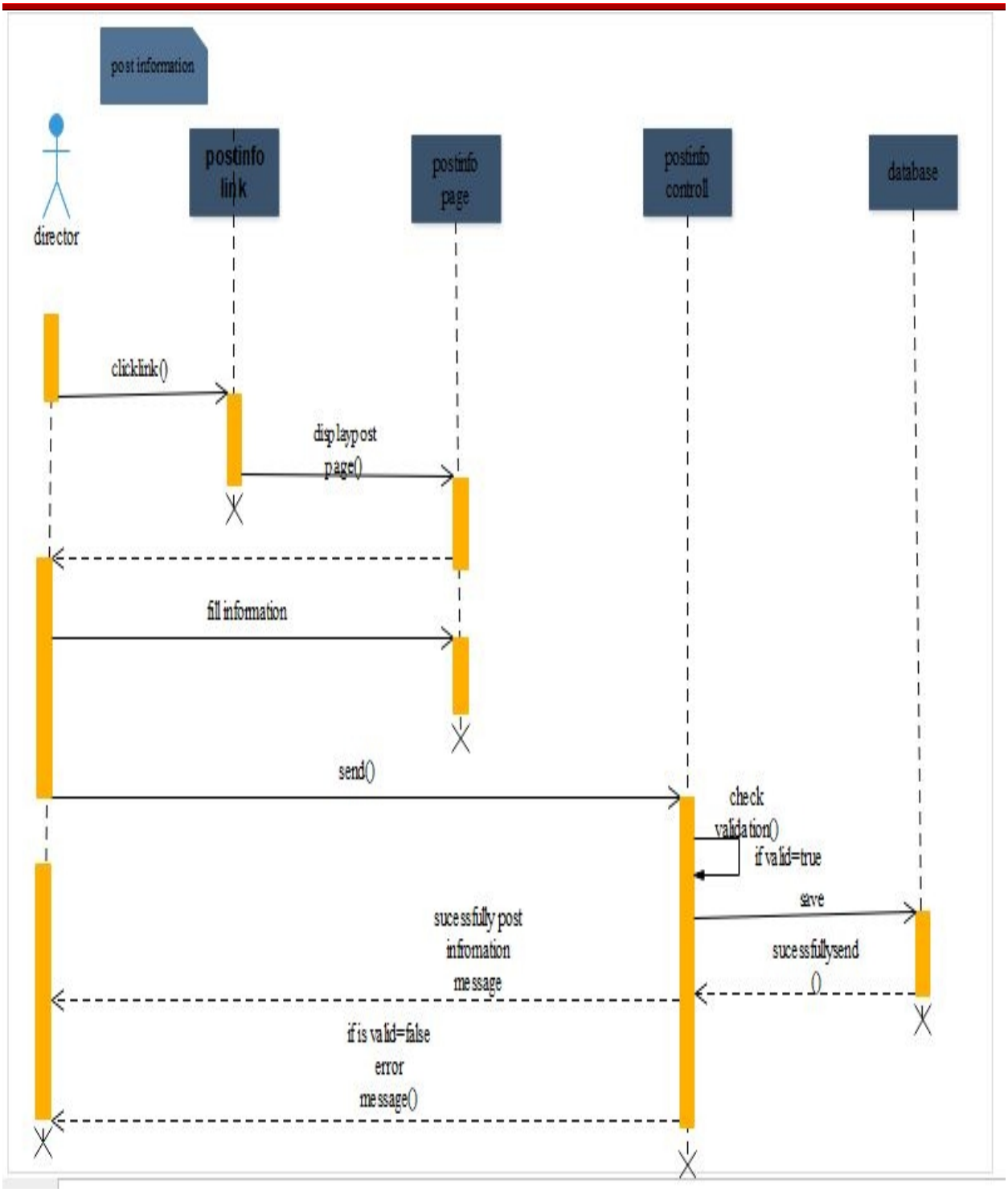


Figure 4. 8: post information of sequence diagram

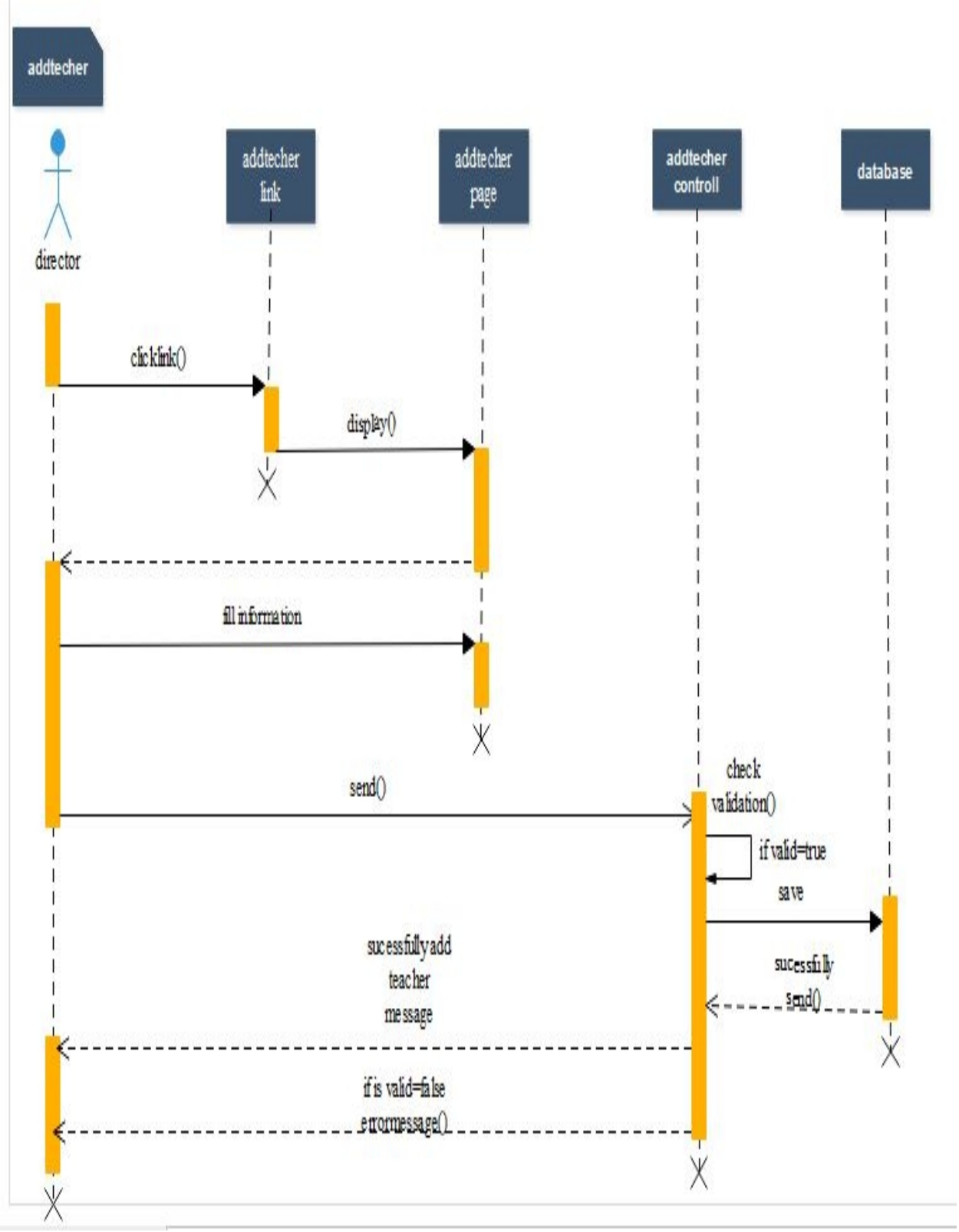


Figure 4. 9: add teacher of sequence diagram

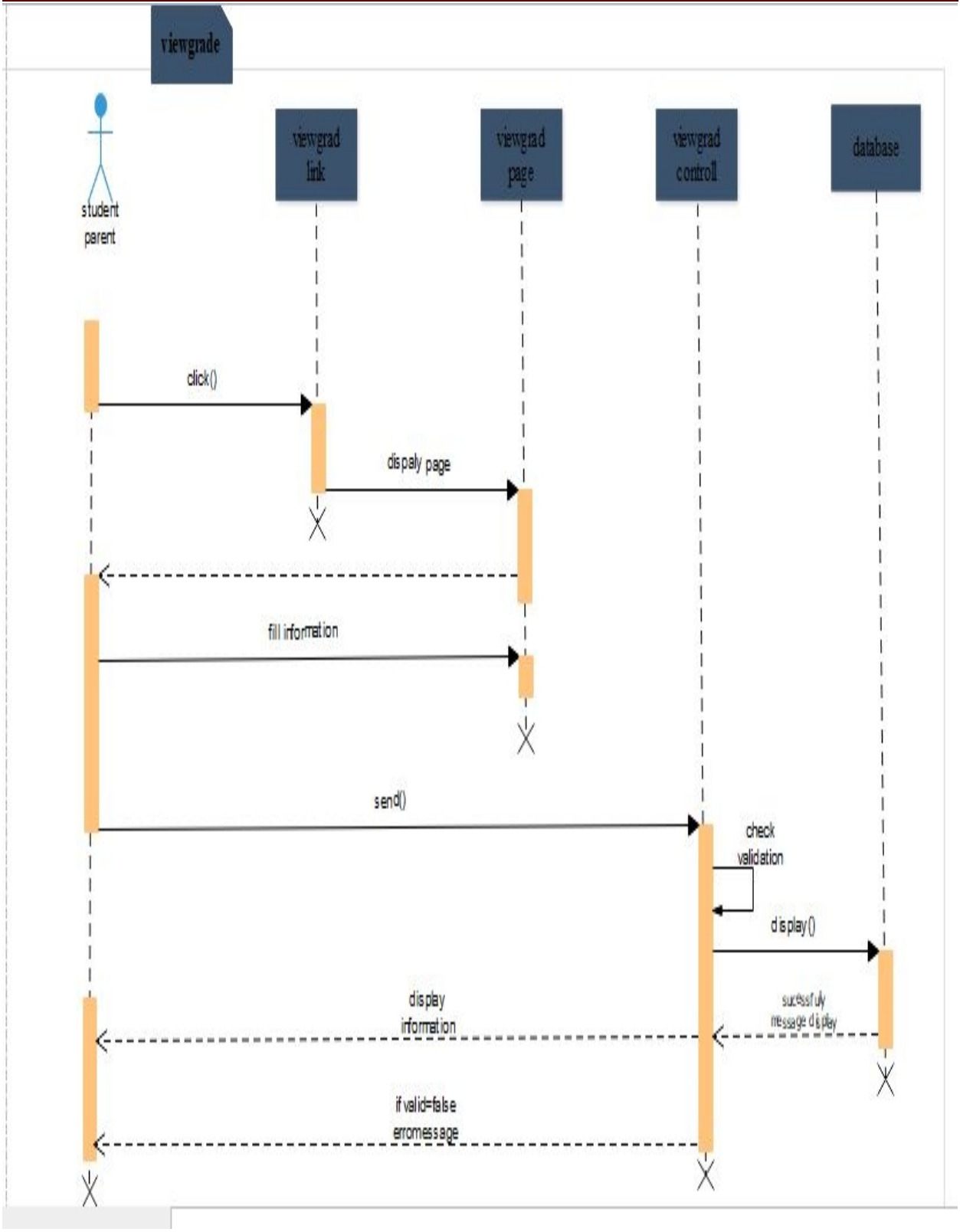


Figure 4. 10: view grade of sequence diagram

4.3.2. Activity diagram

Activity diagrams describe the workflow behavior of a system. They are typically used for business process modeling, for modeling the logic captured by a single use case or usage scenario, or for modeling the detailed logic of a business rule. UML activity diagrams could potentially model the internal logic of a complex operation. In many ways UML activity diagrams are the object-oriented equivalent of flow charts Activity diagrams do not give detail about how objects behave or how objects collaborate. Diagrams are read from top to bottom and have branches and forks to describe conditions and parallel activities. . Activity diagrams show the sequence of activities in a process, including sequential and parallel activities. Symbols are used for activities, decisions and so on. Arrows represent events that connect the activities. In the system Abafransua preparatory school management system there are flow of actives that are done on the system.

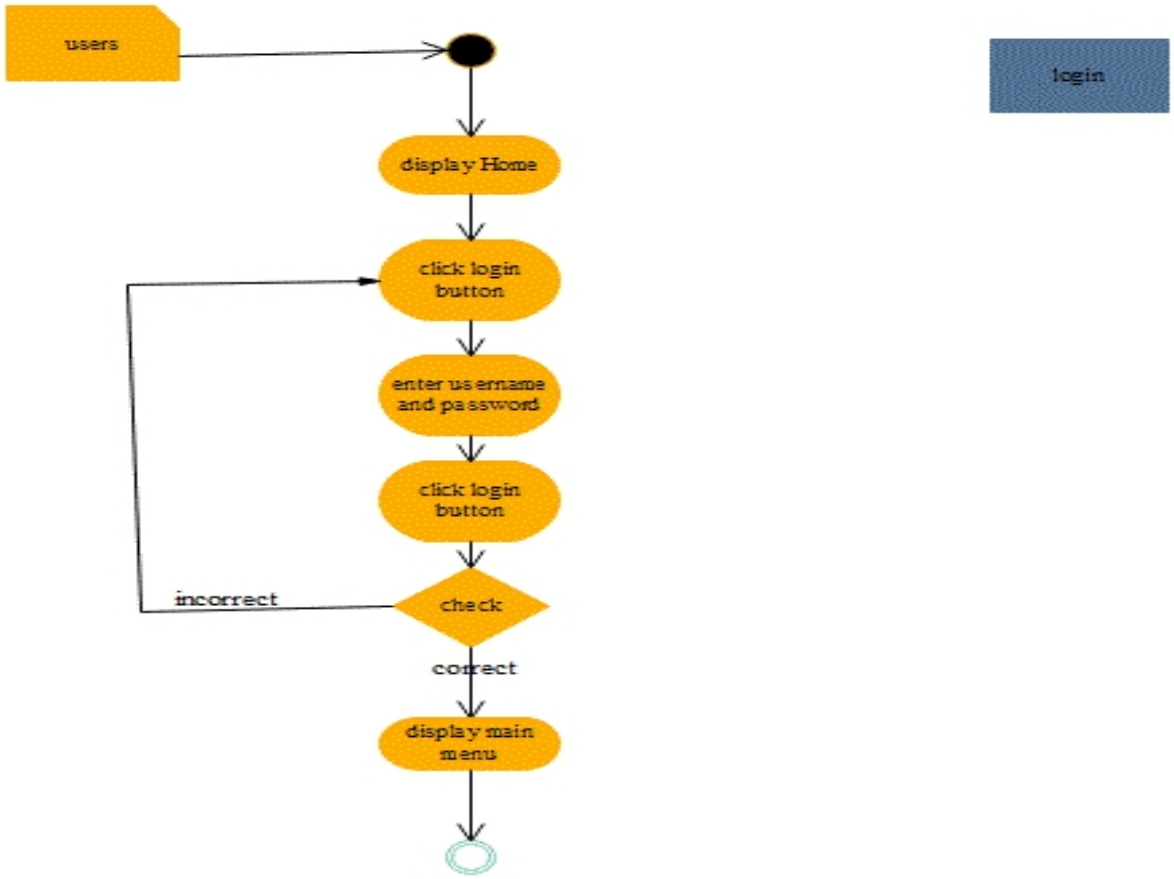


Figure 4. 11: activity diagram login form

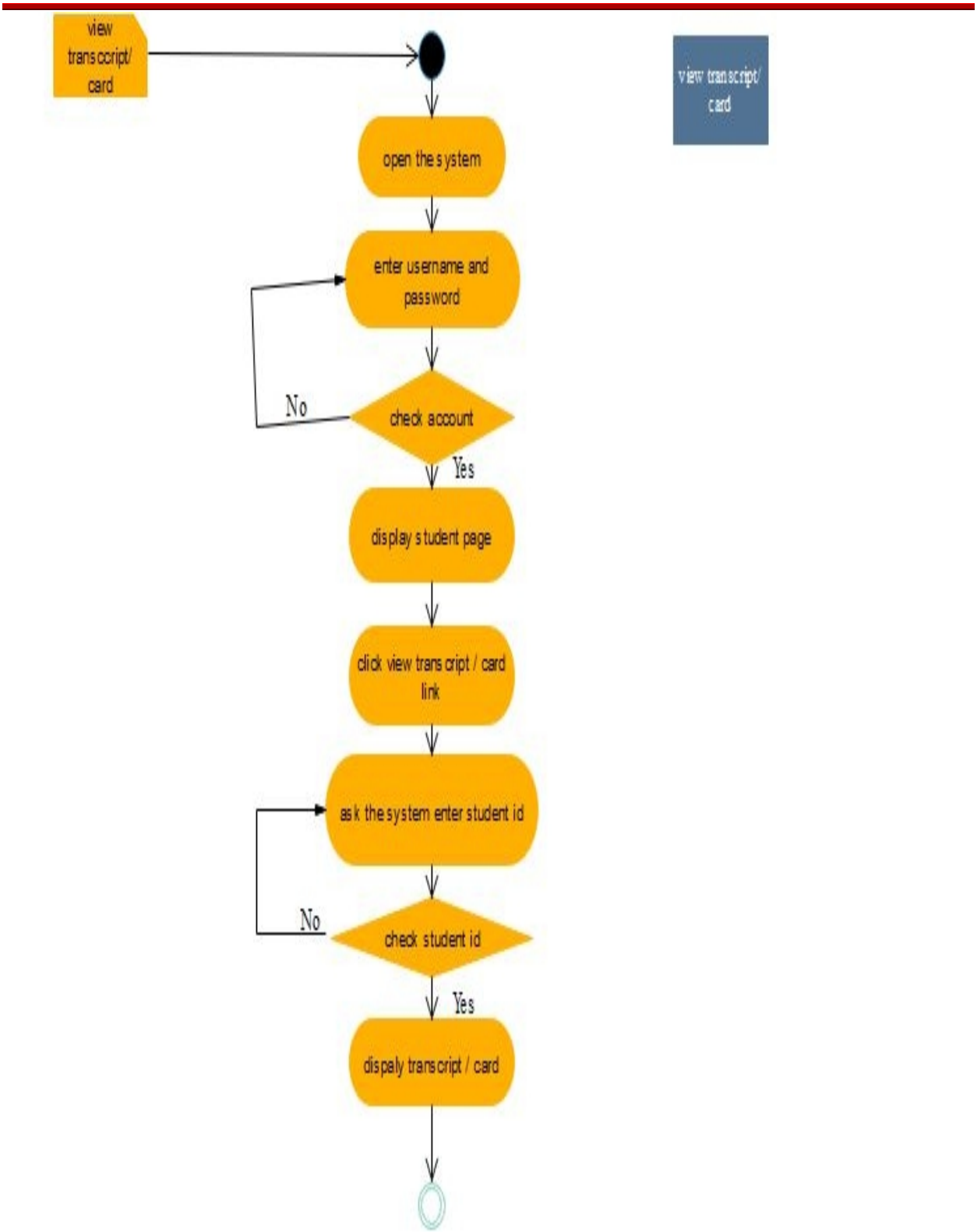


Figure 4. 12: activity diagram for view transcript/card

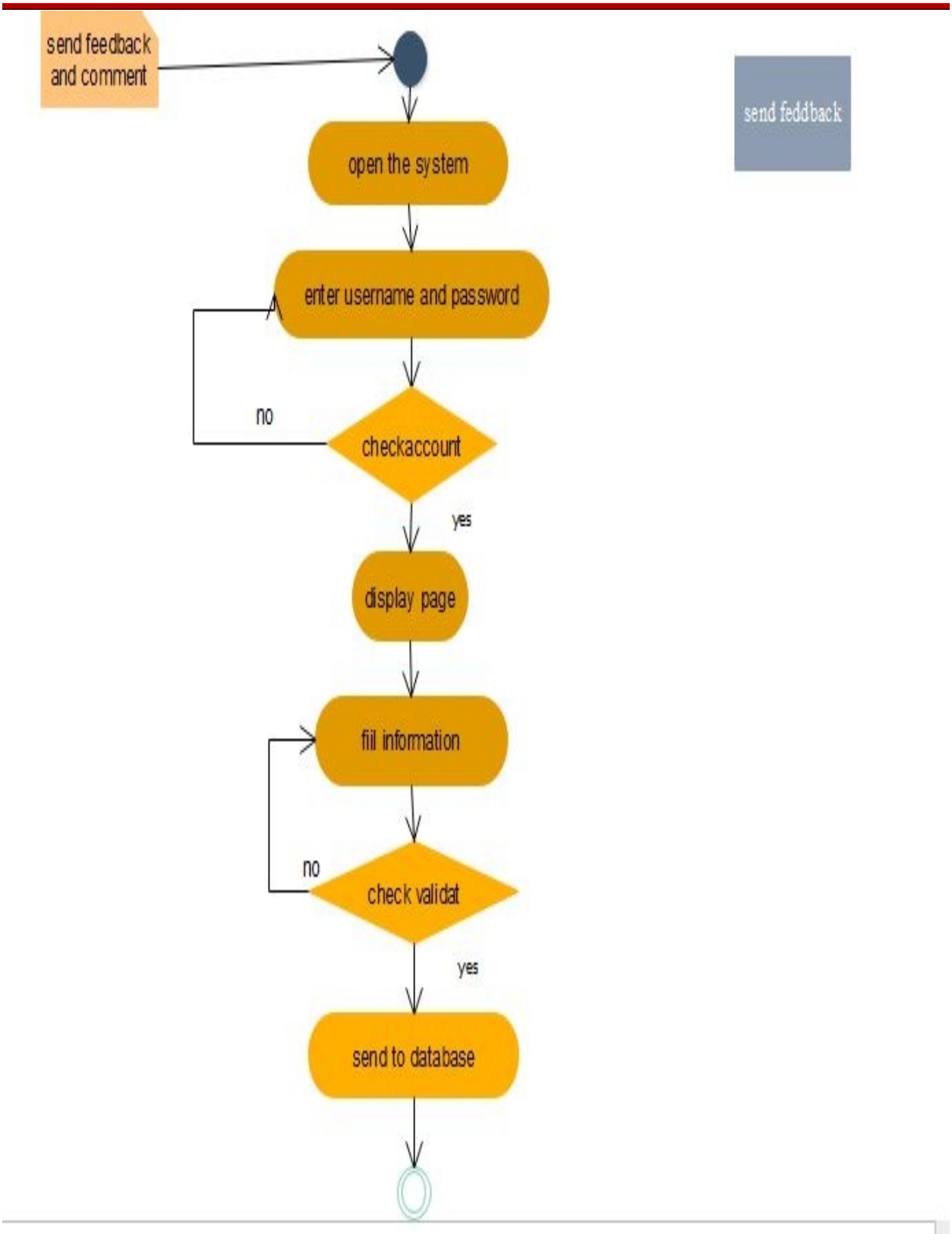


Figure 4. 13: activity diagram for send feedback and comment

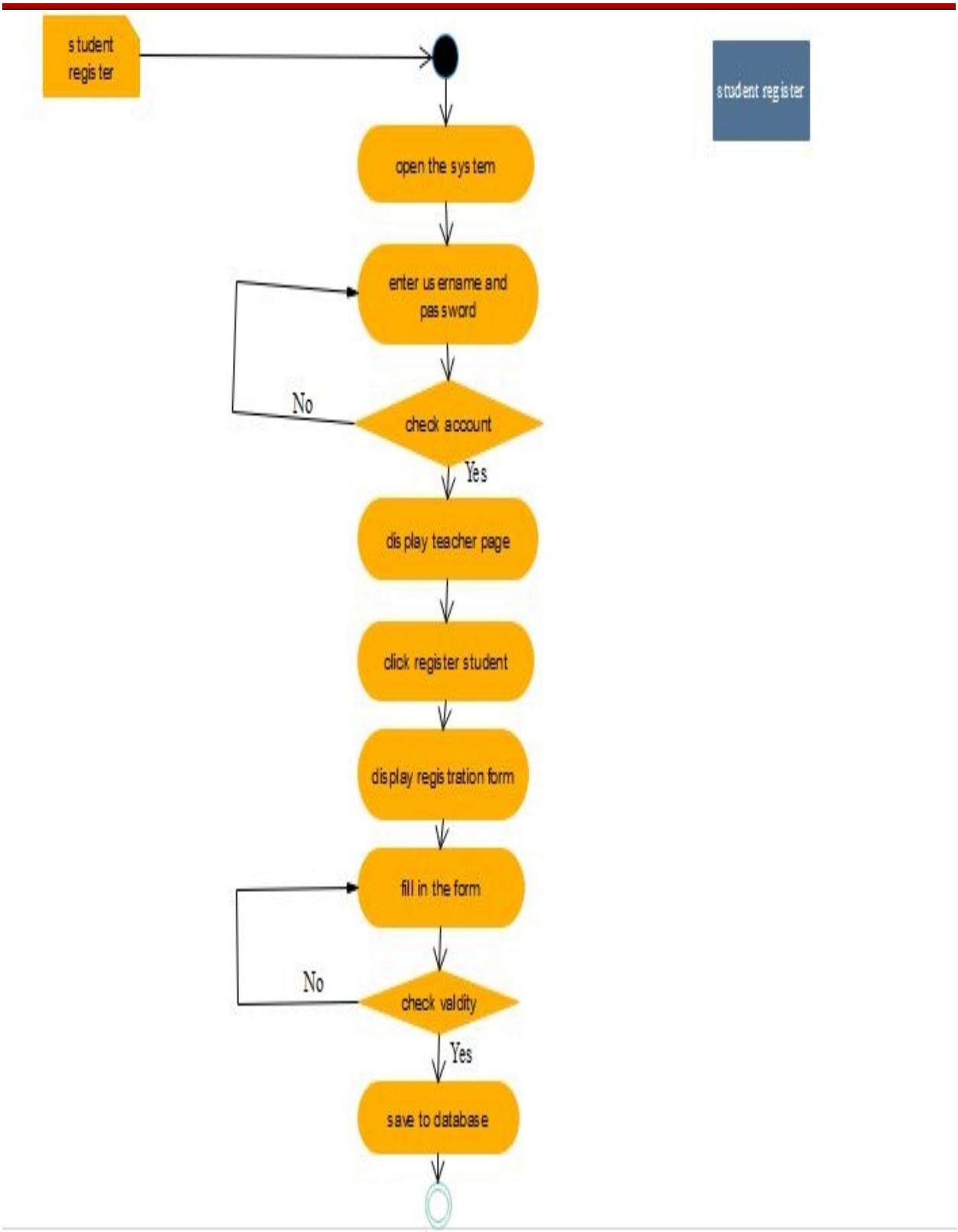


Figure 4. 14: activity diagram for student register

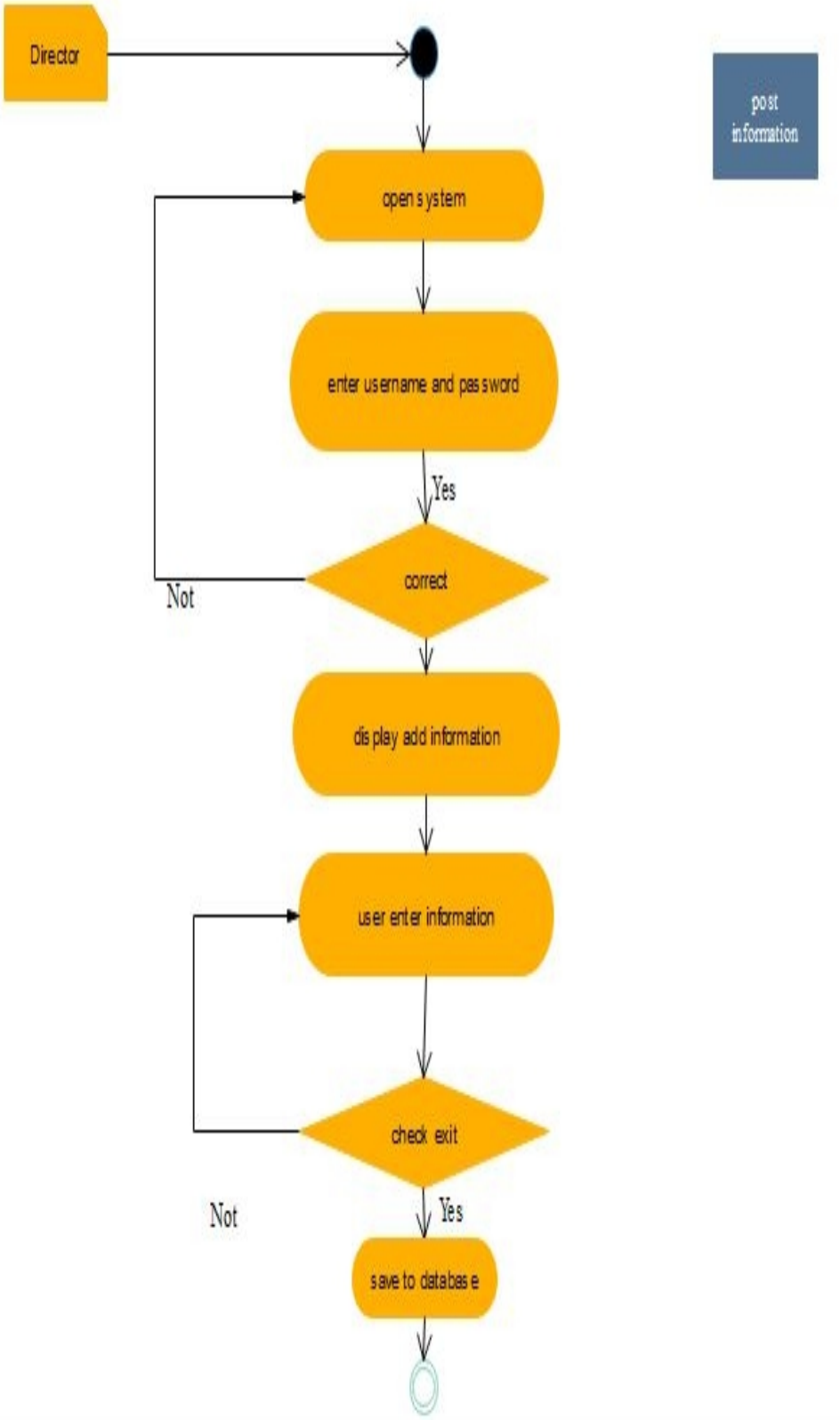


Figure 4. 15: activity diagram for post information

4.3.3. State chart diagram

A state chart diagram is normally used to model how the state of an object changes in its lifetime State chart diagrams are good at describing how the behavior of an object changes across several use case executions. A state chart is a hierarchical model of a system and introduces the concept of a composite state also called nested state. They have their own state these states are Initial state- This is represented as a filled circle. Final state- This is represented by a filled circle inside a larger circle. State- These are represented by rectangles with rounded corners. Transition- A transition is shown as an arrow between two states.

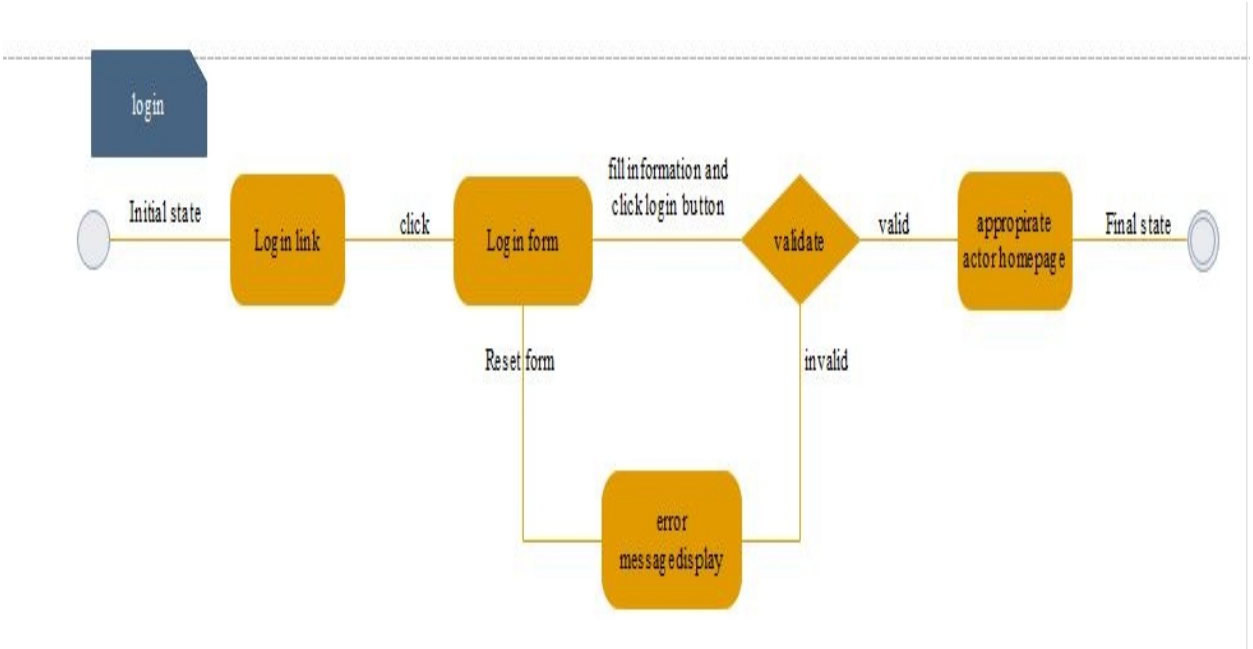


Figure 4. 16: state chart for login

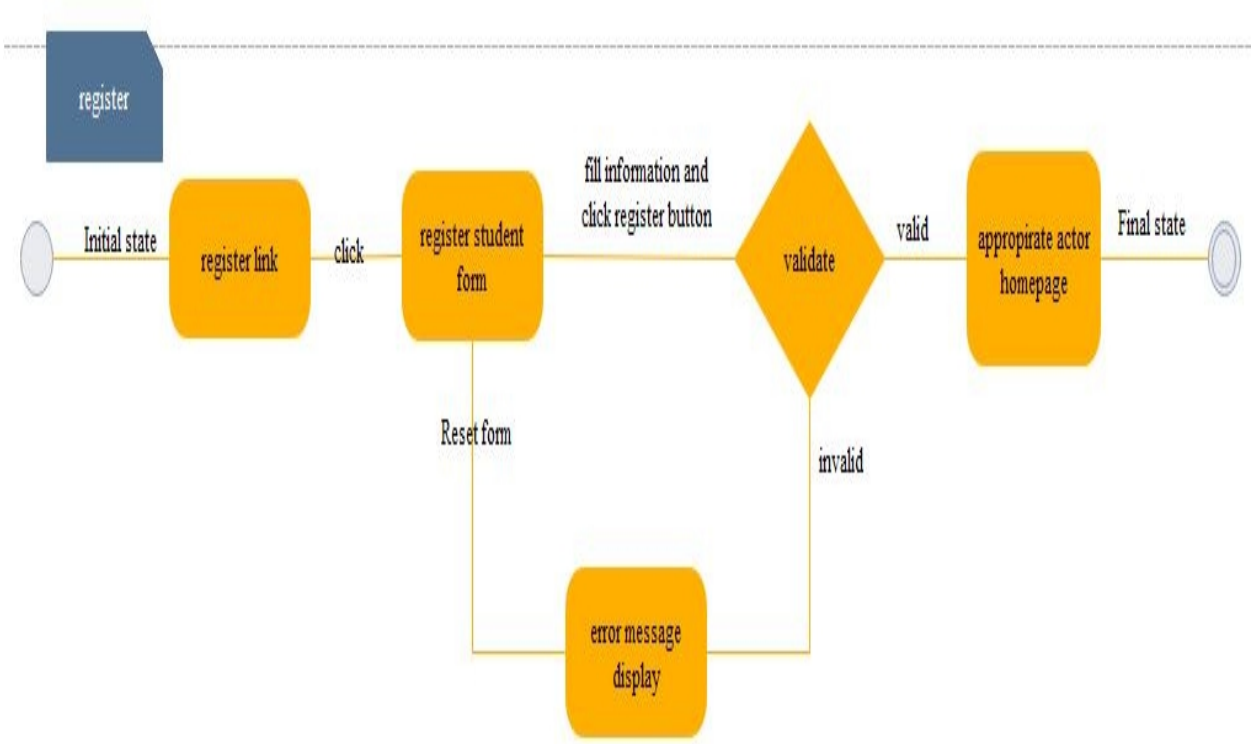


Figure 4. 17: state chart for register

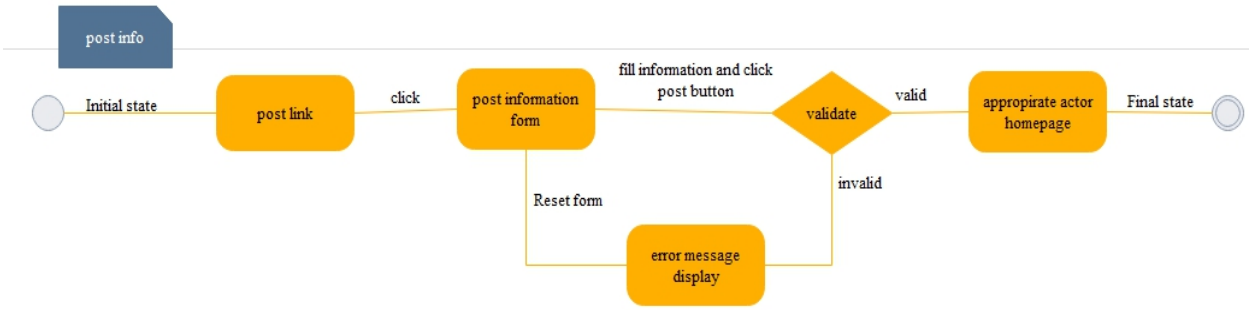


Figure 4. 18: state chart for post information

CHAPTER FIVE

5. SYSTEM DESIGN

5.1 Introduction

In this chapter we describe the important system qualities and defines the values against which options are evaluated (i.e. it describes the qualities of the system that developers should optimize). The objectives of design are to model the system with high quality. Implementing of high quality system depend on the nature of design created by the designer. If one wants to changes to the system after it has been put in to operation depends on the quality of the system design. So if the system is designed effectively, it will be easy to make changes to it. The design goals are derived from the nonfunctional requirements. The main goal of the system design is to manage complexity by dividing the system in to smaller manageable piece. Some of the goal is listed below this Design goals describe the qualities of the system that the developers should consider. Generally the main goal of system design is that in order to exclude the irrelevant one and to include the relevant one to the system or including functionality or functional requirements to the system in order to design the system we have use different diagrams like: - hard ware software mapping, current and proposed software architecture, Persistent data management and Access control and security.

5.2. Goals of design

The major advantage or goal of system design is used to include the essential or relevant and to remove or exclude irrelevant one in to the system so by including the relevant one we have going to design Abafransua preparatory school management system.

User Interface and Human Factor

The system we are going to develop will have a user friendly graphical user interface (GUI) which allows users to interact with the system easily. The users can easily contact to our system because of users have some know how about computer knowledge and it will easily contact the system. Users of the system are teachers, students, directors and record officer because of the listed user the user is expected to have knowledge. Generally it includes the necessary features

for each user friendly and interactive interface the system should have to provide interactive and easily manageable user interface for the user.

Hard ware consideration

Needs a hardware and like ram, desktop computer, hard disk and the system or our project will deployed by the existing computer and the system is portable it will run by any type of computer and it support any type of run browser.

Security

Any information system must be secure to fulfill the three main components of system security which are confidentiality, integrity and availability, therefore to make the system that we are going to develop secure, we are going to employ authentication mechanism which is giving the user username and password so that no one apart from the authenticated user can get access to the system and our system have authentication and authorization system in order to make our system secure and not attacked by any or different crackers and in order to control data losing and users have password to access privilege to our system and have flexible password in order to control hack by others and we have use MD5 secure system in order to much to the system.

Performance since the system is going to accessed by different user with different needs, it should be capable handling and process query. Generally the system should be able to handle many users and easily responsive. The system should be responsive fast in order to access the required information. It able to get or run on a short time.

Error Handling

When a user interacts with the system errors may occur. To control this kind of inaccuracies system will generate different user friendly messages. To do this, most of the system execution buttons will be controlled according to the sequence which the user is expected to follow, or this can be done by generating different system responses to the input of the user. By using different programming tools we able to solve the error like fatal error and other different errors by using programming tools like JavaScript.

Quality issue

Availability the system should available at any time and reliability the system should control any type of crash during run time since users can use simultaneously. Validation for input user that must put the exact value to the provided system be done in our system and it avoid incorrect storage record.

Backup and recovery

Storing data in another place for backup purpose. If the system will lose some data it is easy to get the loosed data.it can be easy to be placing data its data. If the failed or lost it can be recovered from database easily.

Physical environment

The system will installed in the server side script in but for more feature we recommended that the system deployed that is free from natural and human disaster.

Resource issue

In the system it needs a desktop computer and a skill person who can easily interact to the system and get a great feedback all in all resources needed to computer and others. The system consumes high processor speed and memory for both client and server side this enables the system to provide in efficient and effective manner in addition of the mentioned above the system will be best provision and become consistent.

Documentation

Documentation helps or guide a team to make knowledge management and used for users to guide how the system operate. There for it is a necessary requirement and it helps project maintenance purpose. The documentation or final project contains proposal, report and final documentation. The documentation is well defined and written a well-organized way and it also reflect what have done in the implementation system.it also helps how it functional and a process of what activity done the system. Show the process how users use the system also shows how errors and other problems be solved and how system will give feedback to client or users.

5.3. Proposed System Architecture

The proposed software will be implemented in three layers of Client/Server architecture. The layers are the following:

- **The application layer** application layer the users deploy.
- **Logic/ connector layer** the connection between applications to database server.
- **Database layer** the layer that used to store data files on the system.

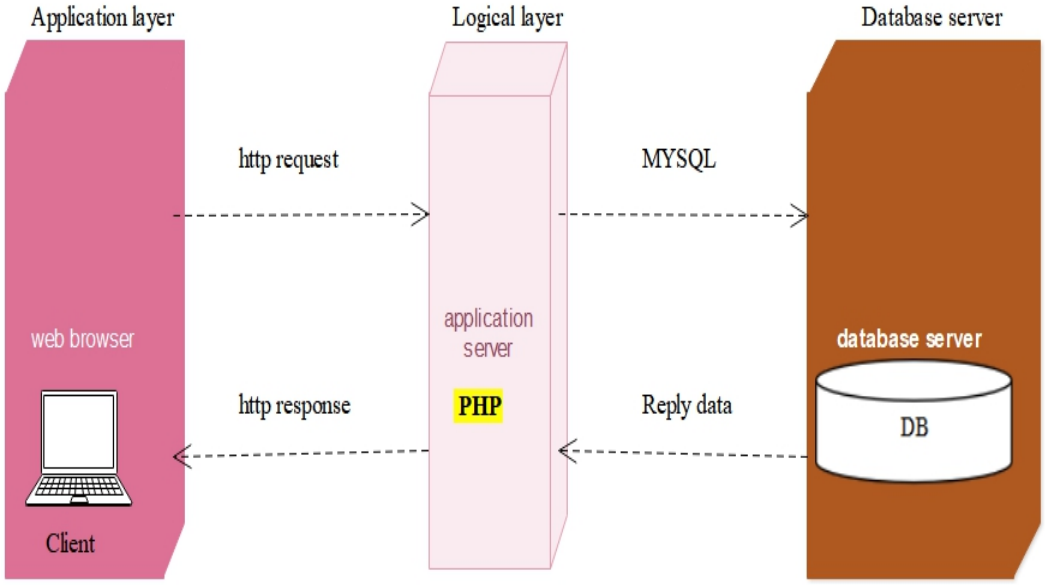


Figure 5. 1: overall architecture of the system

5.3.1. Hardware/Software Mapping

The deployment diagrams show the hard ware for the proposed system, the software that is installed on that hardware and the hidden ware used to connect disparate machines to one another. Therefore, the deployment diagram for the proposed system is design as follows:

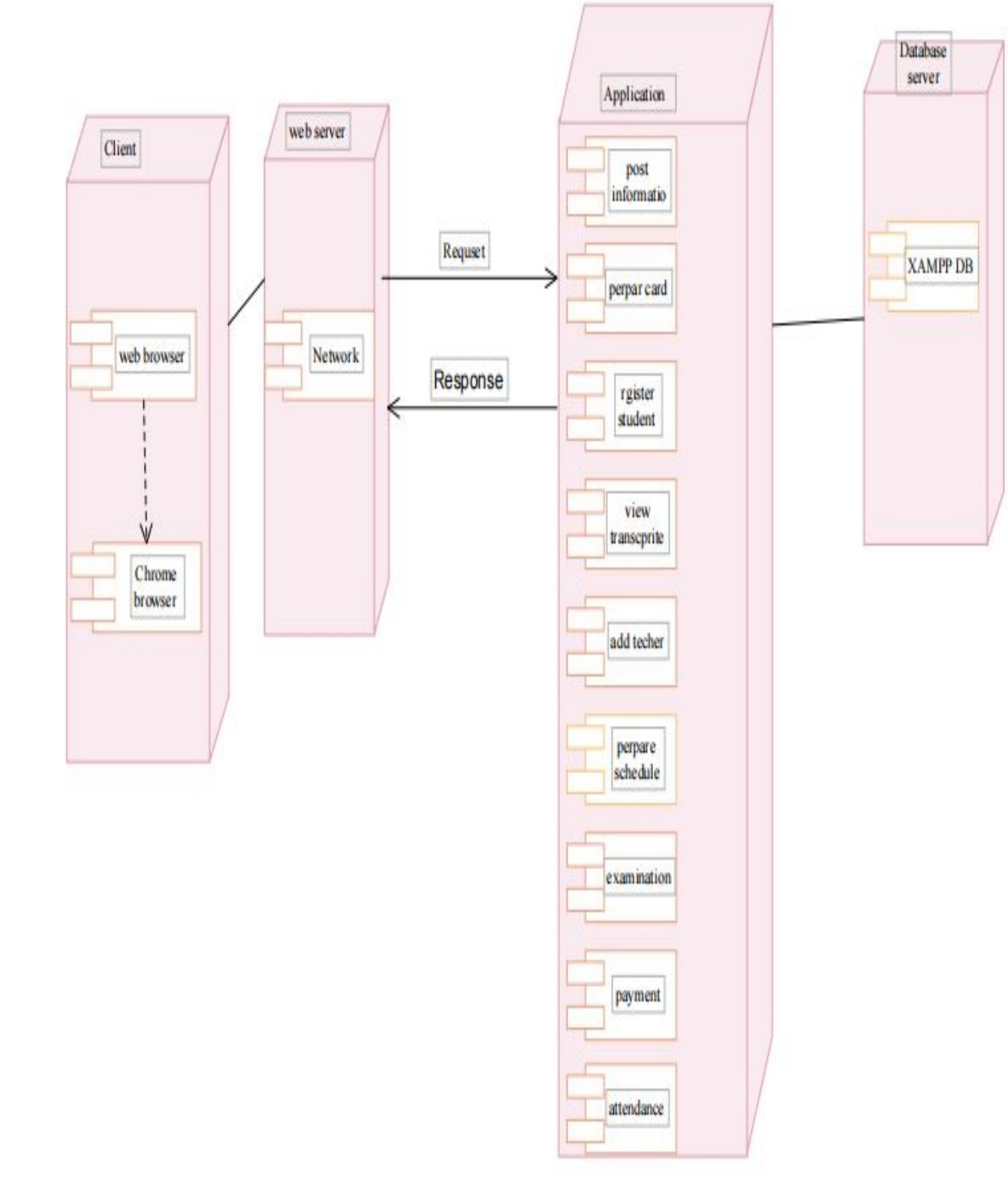


Figure 5. 2: deployment Diagram

5.3.2. Subsystem Decomposition and Description

The component diagram's main purpose is to show the structural relationships between the components of a system. UML component diagrams are great for identifying the architectural landscape for your system as they enable you to model the high-level software components, and more importantly the interfaces to those components. Normally we have sub decomposed in to different components.

Grading system in grading system there are different activates that done in the system these are also: - Average, Sum, Rank, View mark, Prepare card preparing transcript.

Attendance system in attendance there are also different activates or operations done in the system these are:-Sign attendance, view attendance.

Payment system in payment system activates done are pay payment, accept payment

Examination system in examination system prepare exam, take exam will done

Scheduling system in scheduling system there are different activities are view scheduling, arrange scheduling

Registration system in this system students are register.

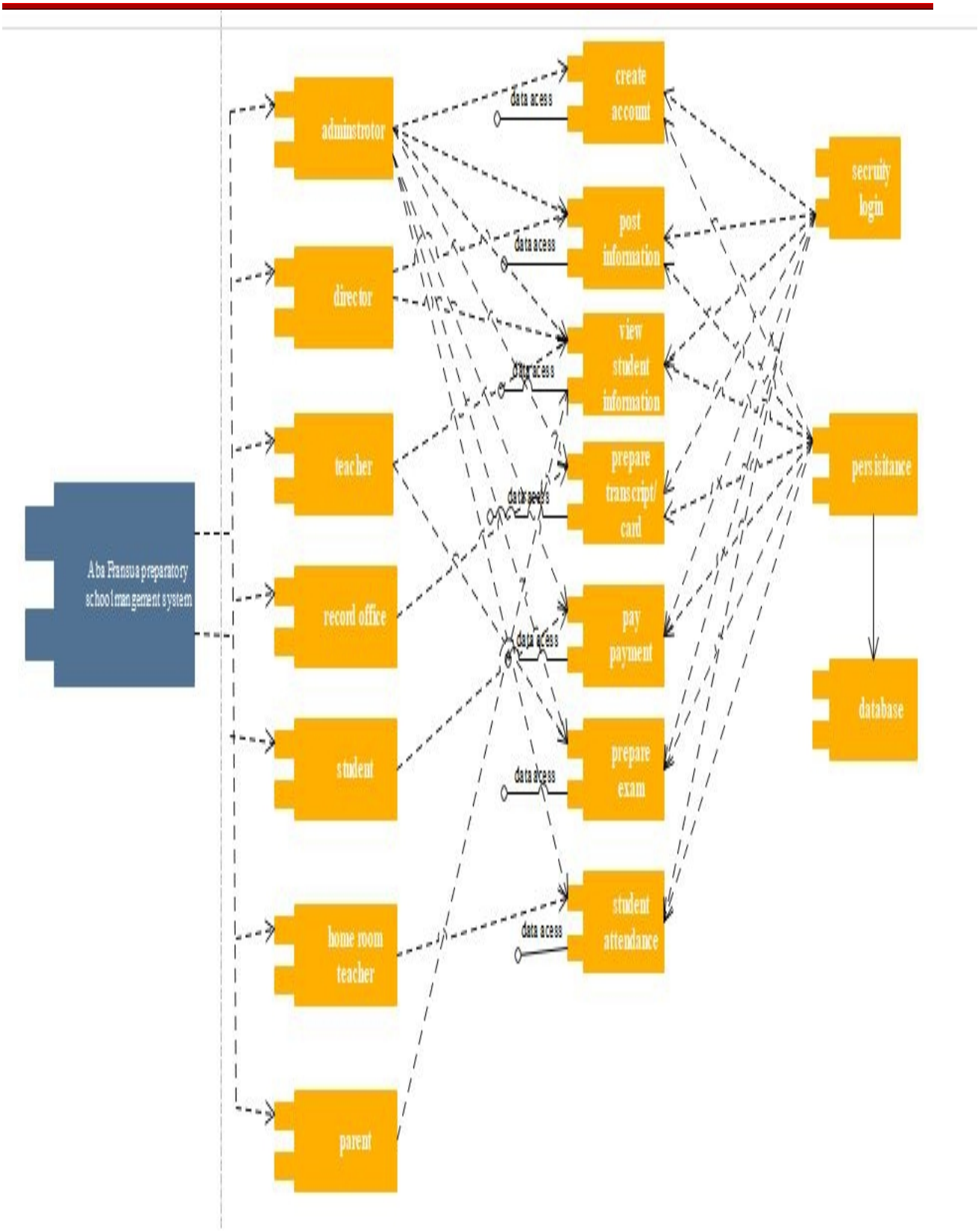


Figure 5. 3: component diagram

5.3.3 Detailed Class Diagram

Normally detailed diagram of the system Abafransua preparatory school management system shows the relationship between classes, functions, and have many relationships like aggregation, 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.

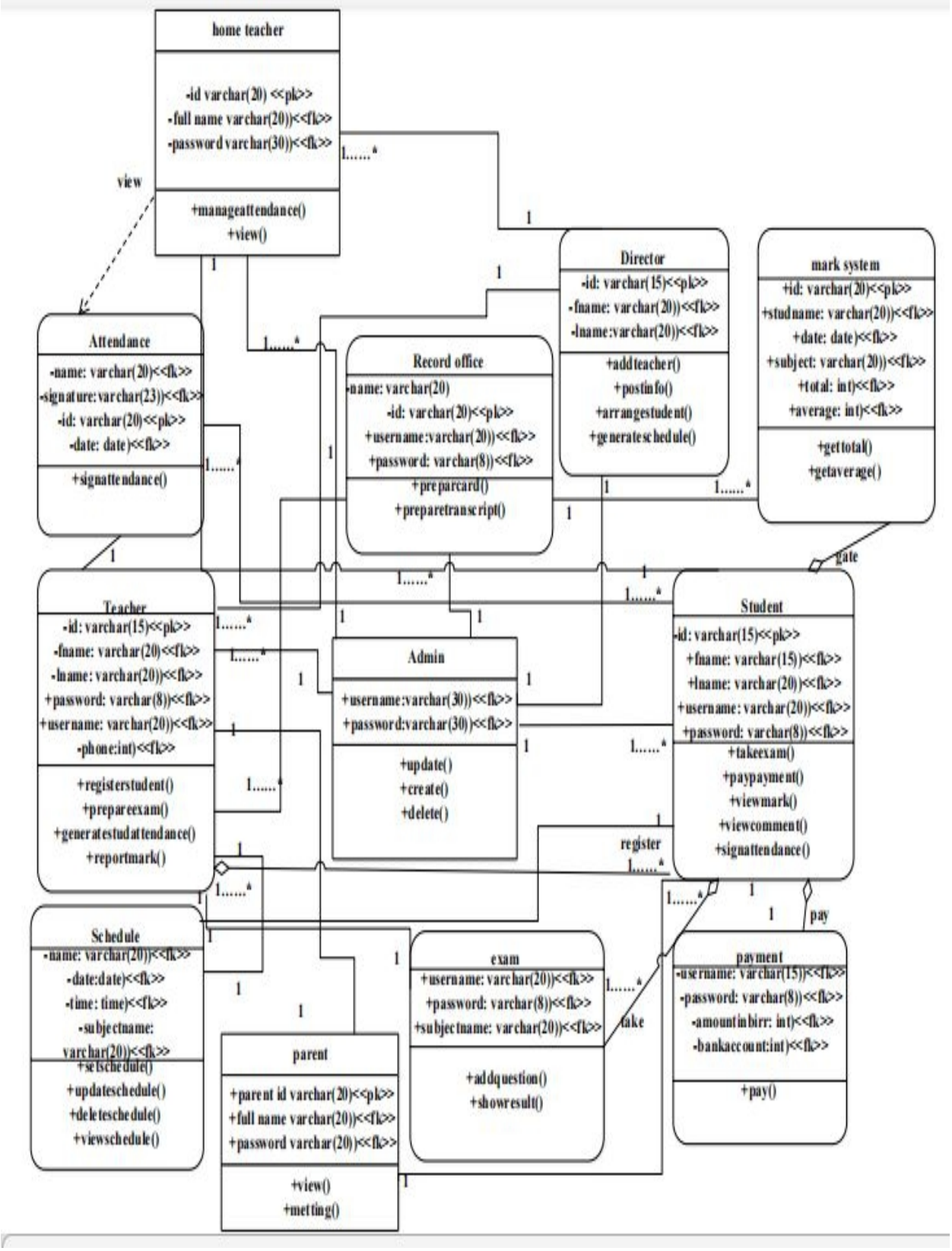


Figure 5. 4: detail class diagram

5.3.4. Persistent Data Management

Persistence modeling deals with how the persistent data i.e. file, database, etc. are stored and managed and to outlives a single execution of the system. Information related to student files and information and other related information are persistent data and hence stored on a database management system to make operate consistently. Data management issue raises the questions, which data need to be persistent, where should persistent data be stored, and how are they accessed. It is only used for object oriented database modeling; but for relational database modeling we use of persistence to show the relation of tables with each other.

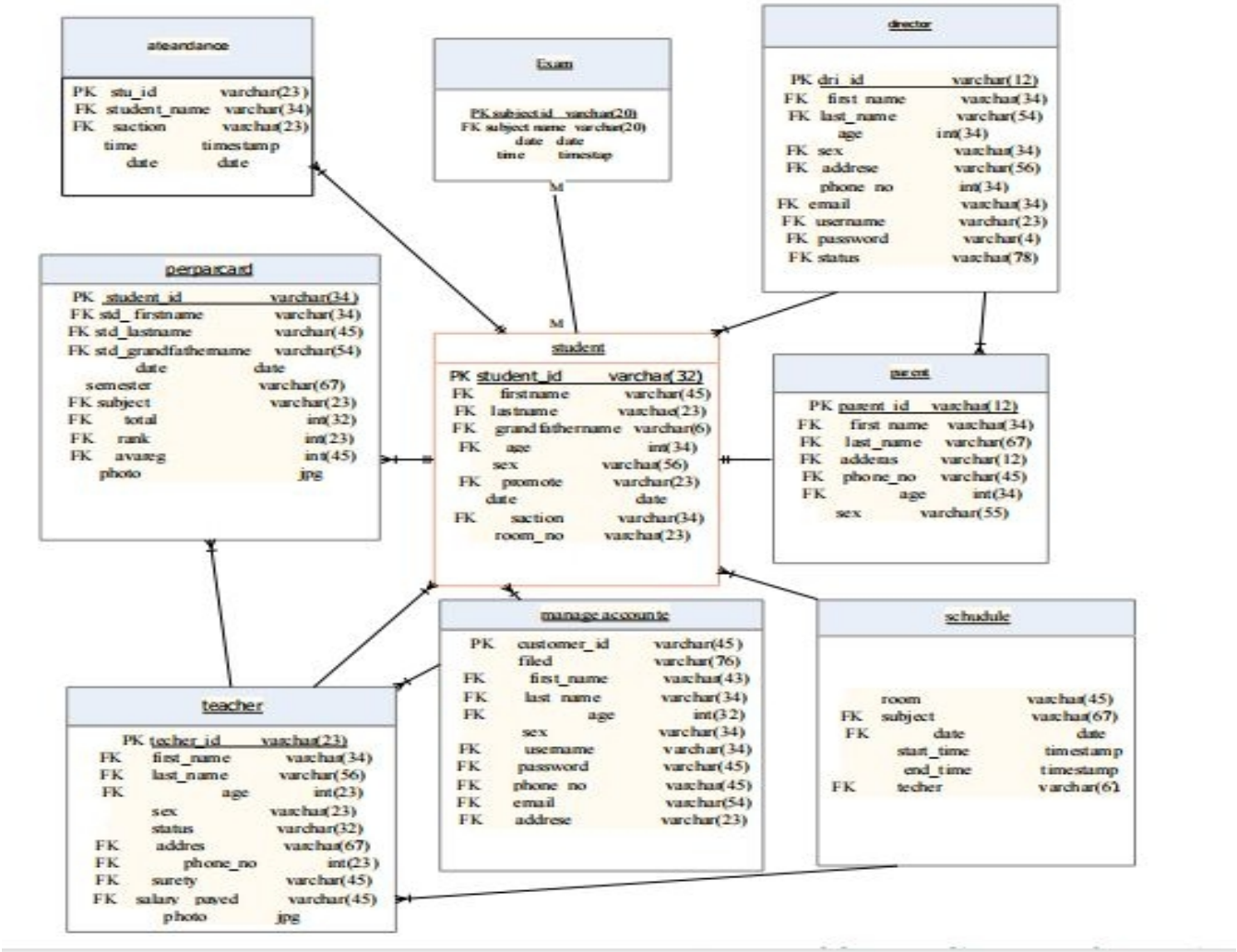


Figure 5. 5: Persistence data management diagram

5.3.5. Access Control and Security

Table 5. 1: access control and security

Actor	Conformation and access control	Account
Administrator	Update account() Delete account() Create account()	Confirm users() Create user account() Delete user account() Update user account()
Student	Take exam() View card() Sign attendance() Payment()	Login()
Teacher	Prepare exam() Register student() Sign attendance	Login()
Record officer	Prepare transcript() Prepare card()	Login()
Director	Arrange student() View attendance() Post information() Add teachers()	Login()
Parent	View grade() Meetings()	Login()
Home room teacher	View student attendance()	Login()
Cashers	View payment() Receipt()	Login()

5.4. Package

UML package diagram describe the decomposition of subsystems into packages and the file organization of the code. This includes an overview of each package, its dependencies with other packages, and its expected usage.

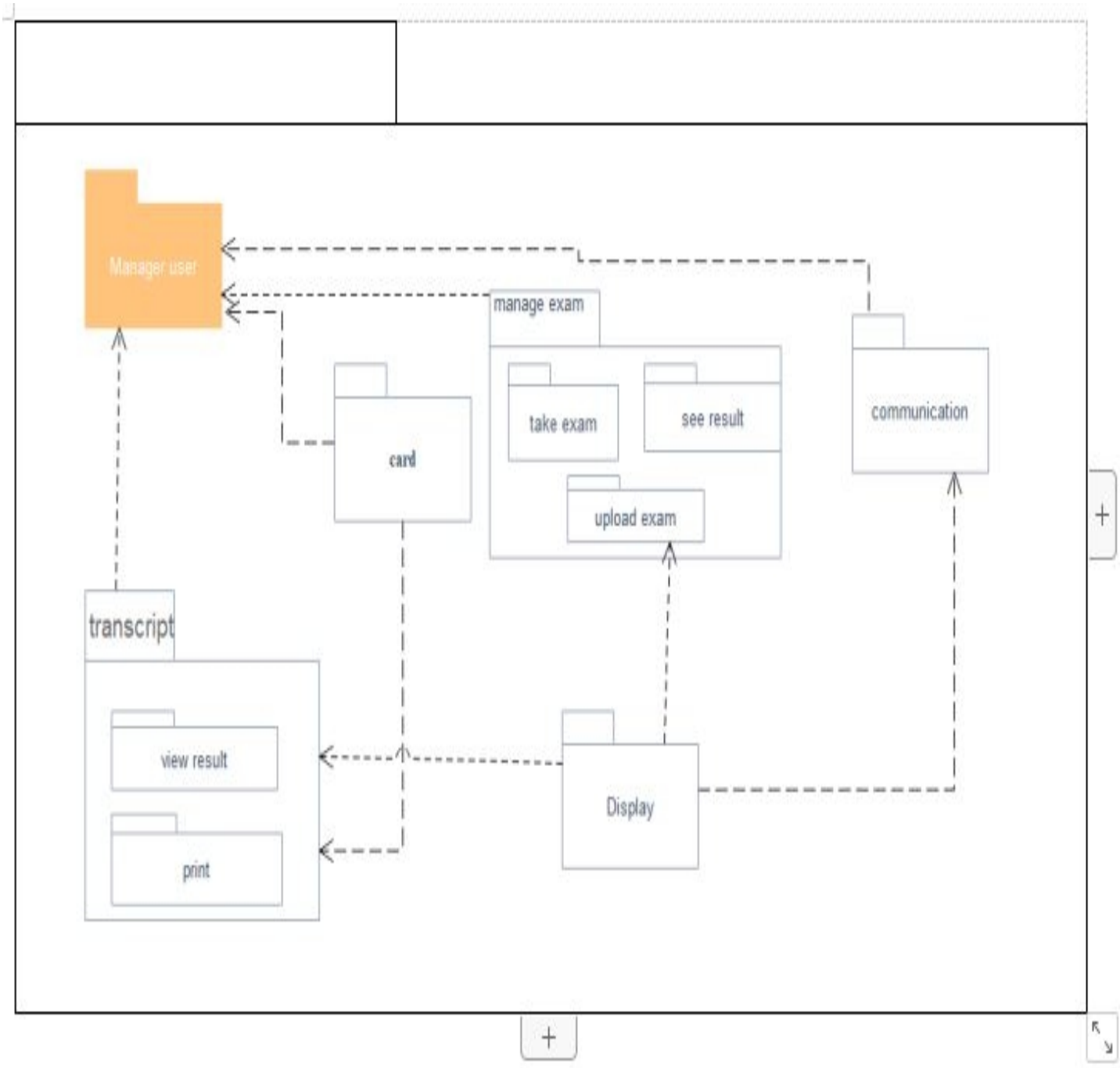


Figure 5. 6: package diagram

5.5. Algorithm design

Pseudo code for login

Method name login

Begin

Get Username

Get Password

IF (Username==entered username && Password==entered password)then put cookies Login successfully

ELSE Login failed display error message

END

Pseudo code for logout

Begin

If the user login to the system then and click logout link then

Delete cookies and display home page

End

Admin

Algorithm: Create account (id, name, fname, lname, username, password, role, phone no)

Input: id, fname, lname, username, password, role, phone no

Output: account created successfully

Process: click create account link

Begin

Enter id number

Search data from database

If data found

Begin if display create account form

Fill the form and click create button

Give an account number for each user data fields give username and password

Save information on database

End

If else

Begin

Else

Data not found from database

Display not found message

Try again

End else end

5.6. User interface design

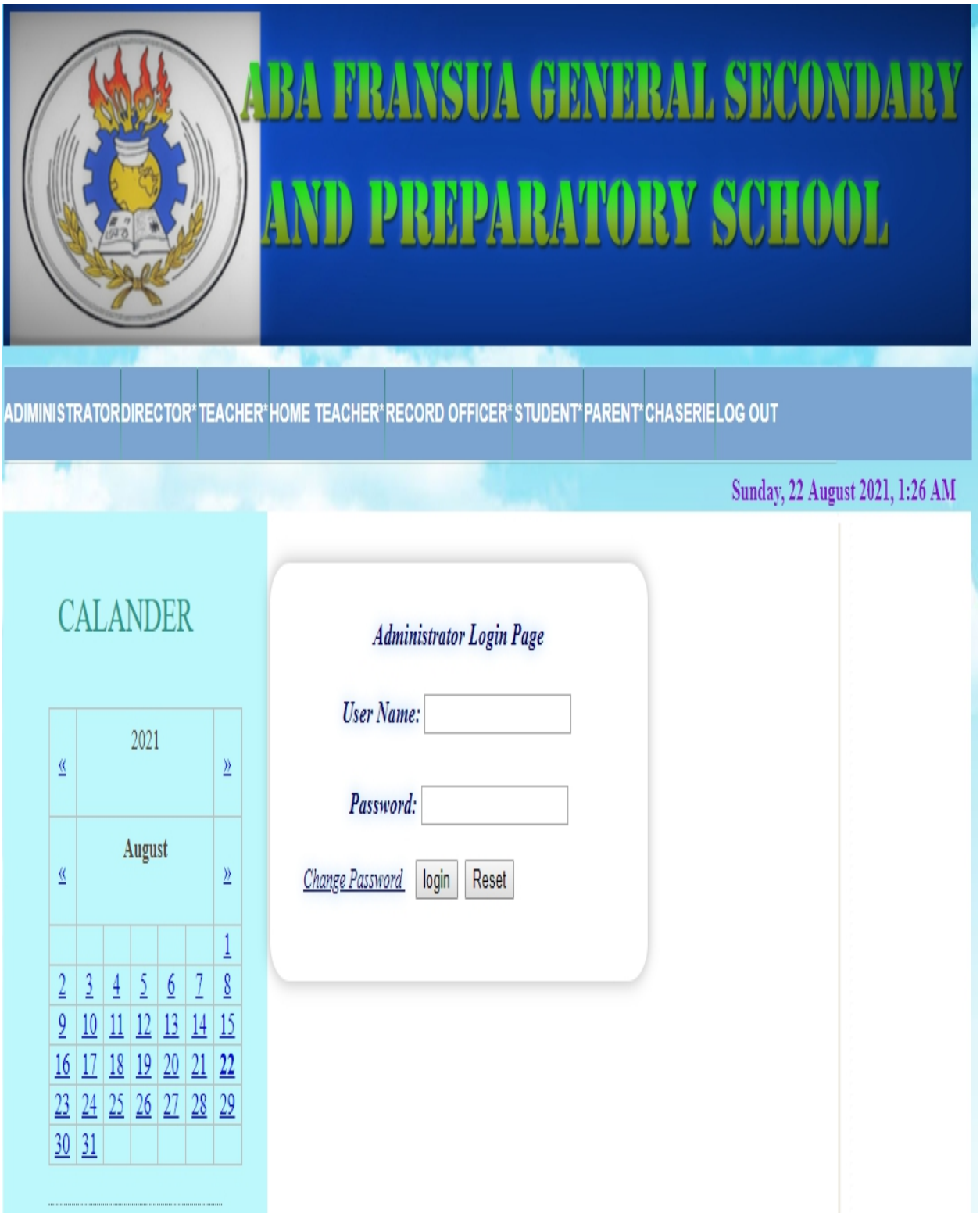


Figure 5. 7: login page

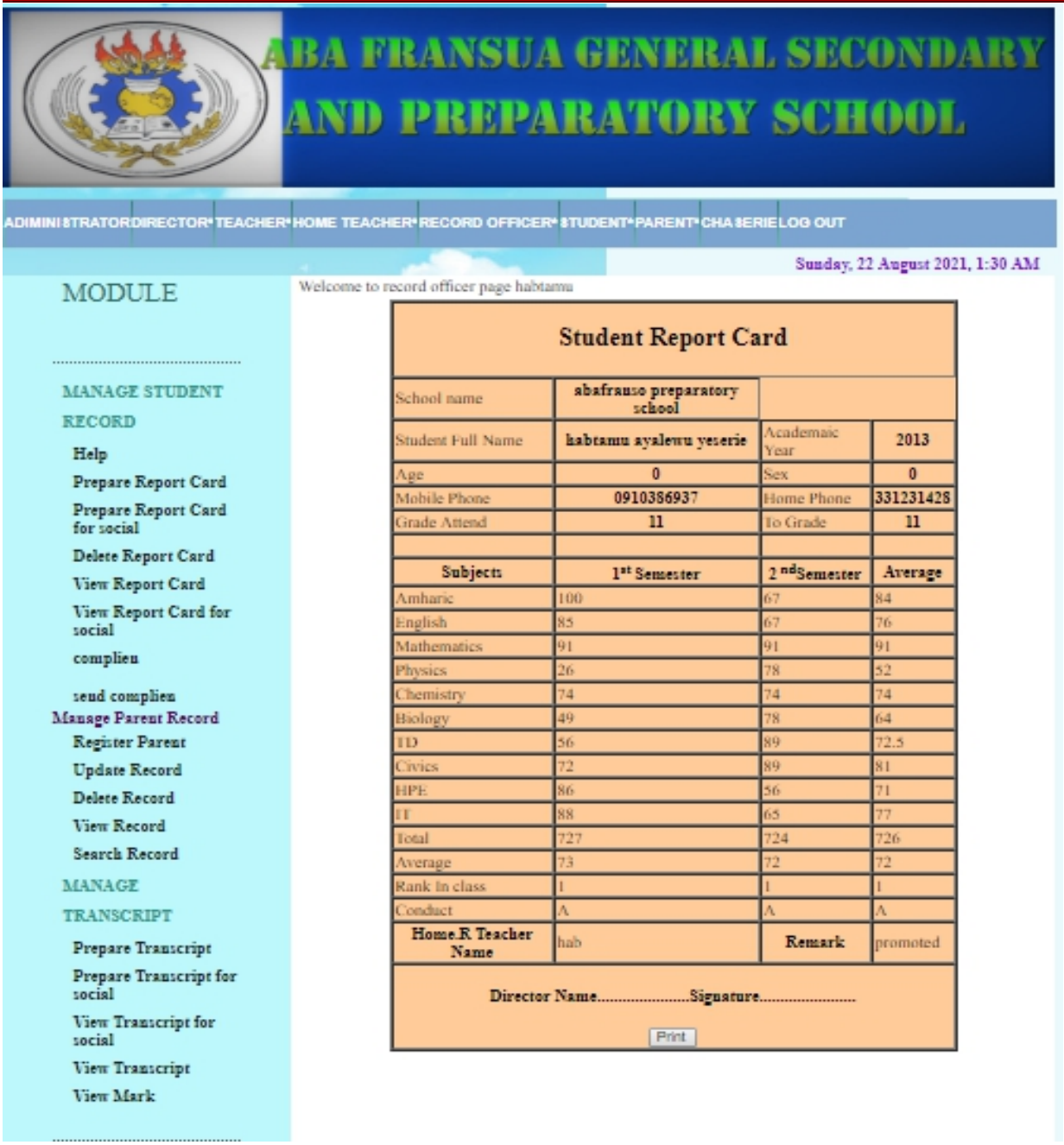


Figure 5. 8: view report card form

CHAPTER 6

6. IMPLEMENTATION AND TESTING

Normally this phase consists of implementing the requirements and design in to code. We used latest and simple technologies to implement this project such as PHP for script and HTML and bootstrap to implement the beautiful interface and MYSQL server for database. The interface design based on the use cases and the database based on the class diagram. The implementation phase is the life cycle is where the actual software is implemented. The result of this code is consist of source code, class diagram implementation, together with documentation more readable. This is what the software implementation is the purpose of this activates is to convert the final physical system specification in to working model with reliable software and hard ware document. Implementation concerned with the type of material (Hard ware and Software required), objective of implementation, code sample of the system. The following section describe some of the tool that are often used in implementation of Abafransua preparatory school management system.

6.1. Implementation of the Database

In order to implement the main application language is MYSQL that used to store database values and used to store it for long time. We have used MYSQL database because that can run any operating or browser application. MYSQL is easy fast and can used for any type of database weather it is relational or simple database, large or small database.so MYSQL server for implementation our system or used to store our data.

6.2. Implementation of the Class Diagram

In this class diagram implementation we have implement the class diagram structure. That implements the view of an application, visualizing, describing and documenting different aspect of the system. And also implements the attribute and operation of the class.

Table 6.1: implementation of class diagram

Class	Method	Variable
+class Admin	+update(); +create(); +delete();	-varchar username; -varchar password;
+class Director	+addTeacher(); +postinfo(); +arrangestudent(); +generateschedule();	-varchar id; -varchar fname; -varchar lname;
+class Teacher	+registerstudent(); +prepareexam(); +generatestudattendance(); +reportmark();	-varchar id; -varchar fname; -varchar lname; -varchar password; -varchar username; -int phone;
+class Student	+takeexam(); +paypayment(); +viewmark(); +viewcomment(); +signattendance();	- varchar id; -varchar fname; -varchar lname; -varchar username; -varchar password;
+class Record office	+preparcard(); +preparetranscript();	-varchar name; -varchar id; -varchar username; -varchar password;
+class Home teacher	+manageattendance(); +view();	-varchar id; - varchar full name; -varchar password;

+class Attendance	+signattendance();	- varchar name; -varchar signature; -varchar id; -date: date;
+class Mark system	+gettotal(); +getaverage();	-varchar id ; -varchar studname; -date: date; - varchar subject; -int total; - int average;
+class schedule	+setschedule(); +updateschedule(); +deleteschedule(); +viewschedule();	-varchar name; -date:date; -time: time; -varchar subjectname;
+class Parent	+view(); +meeting();	-varchar parent id; -varchar full name; -varchar password;
+class Payment	+pay();	-varchar username; -varchar password; -int amountinbirr; -int bankaccount;
+class exam	+addquestion(); +showresult();	-varchar username; -varchar password; -varchar subjectname;

6.3. Configuration of the Application Server

We use XAMMP server because XAMMP is simple, lightweight Apache distribution it is extremely easy to create a local web server for testing and deployment purpose. Everything you needed is to set up a web server – server application (Apache), database (Maria DB) , and scripting language (PHP).XAMMP works equally well on Linux, Mac and Windows. Since it is suitable and the function we listed above we use XAMMP application server.

6.4. Configuration of Application Security

Our system called Abafransua preparatory school management system is an academic issue validates all the inputs by returning error message and suggesting to try again when invalid input occurred. We implement encryption for user password by MD5 when the system admin creates a user account for teachers, student, record officer, homeroom teacher, director and others create their account or the user changes their password system encryption the password. The system has its security principles that control unauthorized authentication.

6.5. Implementation of User Interface

We implement the system we have developed has good and friendly user interface that user can easily navigate the system. The user interface is portion of an interactive computer system that user can communicate the system. Design of the interface includes any aspect of the system that is visible to user. The system has consistence user interface when we navigating through page.

6.6. Testing

Developing a software is a complex process. No matter how hard try to eliminate all faults simply by going the phases of requirements, design and implementation, however through good practice we can make sure that most series faults does not occur in the first place. In addition, we need separate testing phase with the goal of eliminating all remaining fault before releases.

6.6.1. Testing Tools and Environment

In our testing mechanism sublime, google chrome or Firefox and XAMMP server are used.

6.6.2. Unit Testing

It is done at the source code level for language specific programming language errors such as bad syntax, logical errors or to test a particular function or code of modules a system. The unit test case shall be designed to test the validity of a program correctness. It is a way of testing each of a

functionality independently. Accordingly, we have tested each and every activates of the system and the rest of accompanying activates independently using different user input ,different login mechanism and any technique of fault finding so that the incorrect functioning of the activates was corrected at the expected time.

6.6.3. Integration Testing

In this level of testing we have examined how the different procedure or sub components can work together. The type of integration testing that we have followed is bottom up procedure. Since Abafransua preparatory school management system is a web based management system each and every access protocol is depending on hypertext transfer protocol (HTTP) so we have integrate each component from a single functionality(individual interface) to the main function incrementally step by step link like tag.

6.6.4. System Testing

In this level of process, we have examined how a whole subsystem of Abafransua preparatory school management system for academic issue of Abafransua preparatory school work together to achieve user requirements of the system. The goal of the system testing is to detect that can only be exposed by testing entire integrated or major parts of it .Generally system testing mainly concerned with areas such as performance, security ,validation and configuration sensitivity of the system.

6.6.5. Acceptance Testing

Acceptance testing process is completed by the testing team and the entire product or application is handed over to the customer or few users of customers to test for its acceptability. The user can participate on the acceptance testing of the system. In this case those that are in Abafransua students and employers who are work in that school are end users and they can try to check system and their user requirements. Accordingly to system requirements and other resource (documentation, source code) test cases are generated to determine (Validation and verification) weather the system satisfies need and expectation to solve users problem.in this case we have tested users requirement as a user's using the documentation.

CHAPTER SEVEN

7. CONCLUSION AND RECOMMENDATION

7.1. CONCLUSION

The development and advancement of computer technology makes computer to be a part of everyday human life activity. Education is an area where the human is involved day to day activity of their life.it is an area it needs attention, for it deals with behavioral, attitude and skill change. The same is true for the use of computer education.

The system Abafransua preparatory school is a web based system, this system choice has enabled school members can easily function their activity. This project is normally designed to solve different problem that are found in the school. In order to do our project we have use different methodology in order gather or collect information that are help to us in order to do our project. The project included the main functionality like marking system attendance, scheduling, payment, arrangement of a student in a class, preparing student transcript and others are the main functionality of the system. In order to do this we have used programming language like preprocessor hypertext markup language. In front side we have used HTML, java script, cascading style sheet and in back end we have used MYSQL, PHP.

In our system, we developed an automated preparatory school Management System for Abafransua School that facilitates the various activities taking place at preparatory school. Through this way teachers can easily prepare student exams and easily calculate students mark in their course and record officer can easily prepare student mark and register student. Director can easily post information and others. registering, The web application facilitates prepare transcript and report card by the Record officers and students and also to view transcript and report card, to view students status and comment using the internet of the school. As we have try to include the main functionality of the system we have done the main in the system.

7.2. Recommendation

According to the scope and requirement of the client we have tried to develop Abafransua preparatory school management system. All most we have done what we have mentioned in the functional requirements. The system we have developed is a web based Abafransua preparatory school management system essentially needs a skilled person to work with the system. As a new system developer, we recommend the system needs more functional requirements that will do for future. As our recommend the following functional will done for the future like, employee information management system that work in the school, employee salaries pay rolling and employees attendance.

REFERENCES

- ✓ Wolkite university industrial guideline
- ✓ Strial project guideline Rumbaugh, J., Jacobson, I., & Booch, G. (2003). The Unified Modeling Language.
- ✓ . Malan, Ruth, “Functional Requirements and Use Cases” Hewlett-Packard Company,
- ✓ GradyBooch, in Object-Oriented Analysis and Design.
- ✓ Lane,D andWilliams,H.E.(2004)Web Database Applications with PHP & MySQL.

APPENDICES

We use MYSQL for the database implementation.

Database: `school`

Table structure for table `account`

```
CREATE TABLE `account` (
  `recordno` int(3) NOT NULL,
  `email` varchar(50) NOT NULL,
  `firstname` varchar(30) NOT NULL,
  `lastname` varchar(30) NOT NULL,
  `password` varchar(30) NOT NULL,
  `passconf` varchar(40) NOT NULL,
  `phone` int(30) NOT NULL,
  `username` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Table structure for table `classtimetable`

```
CREATE TABLE `classtimetable` (
  `id` int(11) NOT NULL,
  `subject` varchar(34) NOT NULL,
  `day` varchar(30) NOT NULL,
  `section` varchar(20) NOT NULL,
  `room` varchar(20) NOT NULL,
  `grade` int(11) NOT NULL,
  `namee` varchar(30) NOT NULL,
```

```
  `start_time` varchar(30) NOT NULL,
  `end_time` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Table structure for table `commentform`
CREATE TABLE `commentform` (
  `date` int(11) NOT NULL,
  `username` varchar(30) NOT NULL,
  `studid` varchar(30) NOT NULL,
  `comment` varchar(300) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
Table structure for table `examtable`
CREATE TABLE `examtable` (
  `id` int(11) NOT NULL,
  `subject` varchar(30) NOT NULL,
  `day` varchar(23) NOT NULL,
  `grade` varchar(23) NOT NULL,
  `namee` varchar(43) NOT NULL,
  `exam_name` varchar(23) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
 Table structure for table `homlogin`
CREATE TABLE `homlogin` (
  `recordno` int(11) NOT NULL,
  `username` varchar(30) NOT NULL,
```

```
  `email` varchar(30) NOT NULL,
  `password` varchar(30) NOT NULL,
  `passconf` varchar(30) NOT NULL,
  `firstname` varchar(50) NOT NULL,
  `lastname` varchar(50) NOT NULL,
  `phone` int(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Table structure for table `markform1`
CREATE TABLE `markform1` (
  `stuid` int(11) NOT NULL,
  `sname` varchar(40) NOT NULL,
  `Semester` varchar(20) NOT NULL DEFAULT 'I',
  `gsection` varchar(20) NOT NULL DEFAULT '11',
  `tid` int(15) NOT NULL,
  `sex` varchar(20) NOT NULL,
  `amharic` float NOT NULL,
  `english` float NOT NULL,
  `maths` float NOT NULL,
  `physics` float NOT NULL,
  `biology` float NOT NULL,
  `chemistry` float NOT NULL,
  `civics` float NOT NULL,
  `it` float NOT NULL,
```

```
  `economics` float NOT NULL,
  `hpe` float NOT NULL,
  `condact` varchar(3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
Table structure for table `mst_question`
CREATE TABLE `mst_question` (
  `que_id` int(5) NOT NULL,
  `test_id` int(5) DEFAULT NULL,
  `que_desc` varchar(150) CHARACTER SET utf8mb4 DEFAULT NULL,
  `ans1` varchar(75) CHARACTER SET utf8mb4 DEFAULT NULL,
  `ans2` varchar(75) CHARACTER SET utf8mb4 DEFAULT NULL,
  `ans3` varchar(75) CHARACTER SET utf8mb4 DEFAULT NULL,
  `ans4` varchar(75) CHARACTER SET utf8mb4 DEFAULT NULL,
  `true_ans` int(1) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
Table structure for table `mst_question1`
CREATE TABLE `mst_question1` (
  `que_id` int(5) NOT NULL,
  `test_id` int(5) NOT NULL,
  `que_desc` varchar(150) NOT NULL,
  `ans1` varchar(75) NOT NULL,
  `ans2` varchar(75) NOT NULL,
  `ans3` varchar(75) NOT NULL,
```

```
  `ans4` varchar(75) NOT NULL,
  `true_ans` int(3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Table structure for table `mst_result`
CREATE TABLE `mst_result` (
  `login` varchar(20) CHARACTER SET utf8mb4 DEFAULT NULL,
  `test_id` int(5) DEFAULT NULL,
  `date` timestamp(6) NOT NULL DEFAULT current_timestamp(6) ON UPDATE current_timestamp(6),
  `score` int(3) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
Table structure for table `mst_subject`
CREATE TABLE `mst_subject` (
  `sub_id` int(5) NOT NULL,
  `sub_name` varchar(25) CHARACTER SET utf8mb4 DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
Table structure for table `mst_test`
CREATE TABLE `mst_test` (
  `test_id` int(5) NOT NULL,
  `sub_id` int(5) DEFAULT NULL,
  `test_name` varchar(30) CHARACTER SET utf8mb4 DEFAULT NULL,
  `total_que` varchar(15) CHARACTER SET utf8mb4 DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table `mst_useranswer`

```
CREATE TABLE `mst_useranswer` (
  `sess_id` varchar(80) CHARACTER SET utf8mb4 DEFAULT NULL,
  `test_id` int(11) DEFAULT NULL,
  `que_des` varchar(200) CHARACTER SET utf8mb4 DEFAULT NULL,
  `ans1` varchar(50) CHARACTER SET utf8mb4 DEFAULT NULL,
  `ans2` varchar(50) CHARACTER SET utf8mb4 DEFAULT NULL,
  `ans3` varchar(50) CHARACTER SET utf8mb4 DEFAULT NULL,
  `ans4` varchar(50) CHARACTER SET utf8mb4 DEFAULT NULL,
  `true_ans` int(11) DEFAULT NULL,
  `your_ans` int(11) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Table structure for table `parentform`

```
CREATE TABLE `parentform` (
  `pid` int(11) NOT NULL,
  `firstname` varchar(30) NOT NULL,
  `middlename` varchar(30) NOT NULL,
  `lastname` varchar(30) NOT NULL,
  `sex` varchar(30) NOT NULL,
  `background` varchar(30) NOT NULL,
  `phone` varchar(30) NOT NULL,
  `rship` varchar(30) NOT NULL,
  `stuid` int(30) NOT NULL,
```

```
  `username` varchar(30) NOT NULL,
  `password` varchar(30) NOT NULL DEFAULT '123456'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
Table structure for table `post_info`
CREATE TABLE `post_info` (
  `date` varchar(30) NOT NULL,
  `date1` varchar(30) NOT NULL,
  `title` varchar(100) NOT NULL,
  `information` varchar(500) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
 Table structure for table `reportcard`
CREATE TABLE `reportcard` (
  `schname` varchar(30) DEFAULT 'abafranso preparatory school',
  `stuid` int(30) NOT NULL,
  `stname` varchar(100) NOT NULL,
  `age` int(10) NOT NULL,
  `sex` int(10) NOT NULL,
  `mphone` varchar(30) NOT NULL,
  `hphone` int(11) NOT NULL,
  `ayear` int(10) NOT NULL,
  `grade` int(20) NOT NULL DEFAULT 11,
  `promoted` int(20) NOT NULL,
  `amharic1` int(10) NOT NULL,
```

```
`amharic2` int(10) NOT NULL,
`amharic3` int(10) NOT NULL,
`english1` int(10) NOT NULL,
`english2` int(10) NOT NULL,
`english3` int(10) NOT NULL,
`maths1` int(10) NOT NULL,
`maths2` int(10) NOT NULL,
`maths3` int(10) NOT NULL,
`physics1` int(10) NOT NULL,
`physics2` int(10) NOT NULL,
`physics3` int(10) NOT NULL,
`biology1` int(10) NOT NULL,
`biology2` int(10) NOT NULL,
`biology3` int(10) NOT NULL,
`chemistry1` int(10) NOT NULL,
`chemistry2` int(10) NOT NULL,
`chemistry3` int(10) NOT NULL,
`economics1` float NOT NULL,
`economics2` float NOT NULL,
`economics3` float NOT NULL,
`civics1` int(10) NOT NULL,
`civics2` int(10) NOT NULL,
`civics3` int(10) NOT NULL,
```

```
  `hpe1` int(10) NOT NULL,
  `hpe2` int(10) NOT NULL,
  `hpe3` int(10) NOT NULL,
  `it1` int(10) NOT NULL,
  `it2` int(10) NOT NULL,
  `it3` int(10) NOT NULL,
  `total1` int(10) NOT NULL,
  `total2` int(10) NOT NULL,
  `total3` int(10) NOT NULL,
  `avarage1` int(10) NOT NULL,
  `avarage2` int(10) NOT NULL,
  `avarage3` int(10) NOT NULL,
  `rank1` int(10) NOT NULL,
  `rank2` int(10) NOT NULL,
  `rank3` int(10) NOT NULL,
  `con1` varchar(10) NOT NULL,
  `con2` varchar(10) NOT NULL,
  `con3` varchar(10) NOT NULL,
  `tname` varchar(30) NOT NULL,
  `remark` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Sample code for login

```
<?php
```

```
$con = mysqli_connect("localhost","root","");
if (!$con)
{
die('Could not connect: ' . 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:".mysqli_error());
}
$i=2;
$query="select * from login";
$result=mysqli_query($con,$query);
while($row=mysqli_fetch_array($result)){
if($_POST['username']==$row['username'] && $_POST['password']==$row['password'])
{$i=$i+1;
$_session_register("username");
$_SESSION['username']=$row['username'];
}}
$sql="INSERT INTO login (username, password) VALUES ('$_POST [username] ',' $_POST
[password]')";
if (!mysqli_query($con,$sql))
{
die('Error: ' . mysqli_error());
}
else
echo "1 record added";
mysqli_close($con);
if($i>=1)
header("location:ahome.php");
```

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

Sample code for student register

```
<?php
//session_start();
$con=mysqli_connect("localhost","root","");
if(!$con){
die("error:".mysqli_error());}
$db=mysqli_select_db($con,"school");
if(!$db){
die("error:".mysqli_error());}

?>
<?php
$duery= "select invoiceid from studentreg order by invoiceid desc";
$sql=mysqli_query($con,$duery);
$row =mysqli_fetch_array($sql);
$lastid=$row['invoiceid'];
if(empty($lastid))
{
$number="aba-as001";
}
else
{
$idd=str_replace("aba-as", "",$lastid );
$id=str_pad($idd+1, 3,0, STR_PAD_LEFT);
$number='aba-as'.$id;
}
```

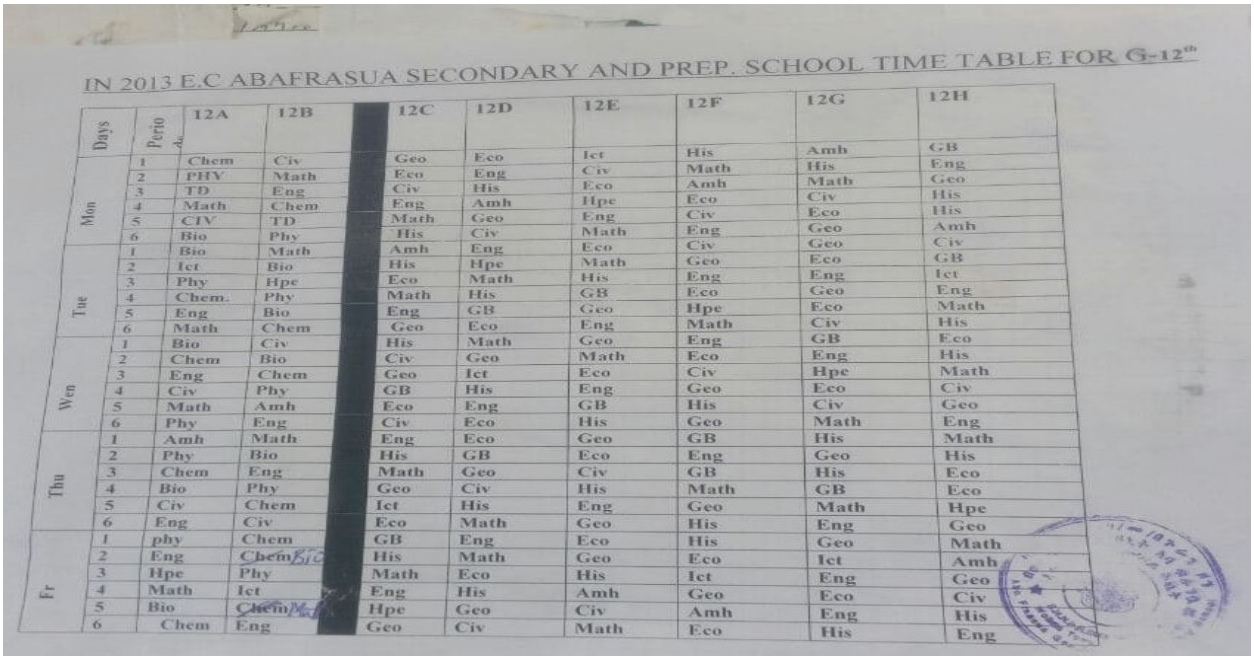
```
?>
<?php
if($_SERVER["REQUEST_METHOD"]=="POST")
{
$invoiceid=$_POST['invoiceid'];
$var1=$_POST['stuid'];
$var2=$_POST['firstname'];
$var3=$_POST['middlename'];
$var4=$_POST['lastname'];
$var5=$_POST['age'];
$var6=$_POST['sex'];
$var7=$_POST['kebele'];
$var8=$_POST['section'];
$var9=$_POST['grade'];
$stream=$_POST['stream'];
$var10=$_POST['avarege'];
$var11=$_POST['recit'];
$var12=$_POST['remark'];
$var13=$_POST['pid'];
$var14=$_POST['cfname'];
$var15=$_POST['clname'];
$target_path='images/';
$target_path=$target_path.basename($_FILES['photo']['name']);
if(move_uploaded_file($_FILES['photo'] ['tmp_name'],$target_path))
{
$a="select section from studentreg where section='A'";
$aa=mysqli_query($con,$a);
$aaa=mysqli_num_rows($aa);
$b="select section from studentreg where section='B'";
$bb=mysqli_query($con,$b);
$bbb=mysqli_num_rows($bb);
```

```
$c="select section from studentreg where section='C'";
$cc=mysqli_query($con,$c);
$ccc=mysqli_num_rows($cc);
$d="select section from studentreg where section='D'";
$dd=mysqli_query($con,$d);
$ddd=mysqli_num_rows($dd);
$e="select section from studentreg where section='E'";
$ee=mysqli_query($con,$e);
$eee=mysqli_num_rows($ee);
$f="select section from studentreg where section='F'";
$ff=mysqli_query($con,$f);
$fff=mysqli_num_rows($ff);
$g="select section from studentreg where section='G'";
$gg=mysqli_query($con,$g);
$ggg=mysqli_num_rows($gg);
$h="select section from studentreg where section='H'";
$hh=mysqli_query($con,$h);
$hhh=mysqli_num_rows($hh);
$i="select section from studentreg where section='I'";
$ii=mysqli_query($con,$i);
$iii=mysqli_num_rows($ii);
$j="select section from studentreg where section='J'";
$jj=mysqli_query($con,$j);
$jjj=mysqli_num_rows($jj);
$k="select section from studentreg where section='K'";
$kk=mysqli_query($con,$k);
$kkk=mysqli_num_rows($kk);

if($var8=="A" and $var9=="11")
{
if(!$con)
```

```
{
die("connaction is fialed");
}
elseif($aaa<=40)
{
$query="INSERT INTO studentreg (invoiceid, stuid, firstname, middlename, lastname, age, sex, kebele, section, grade, stream, avarege, recit, remark, pid, cfname, clname, photo) values ('$invoiceid','$var1','$var2','$var3','$var4','$var5','$var6','$var7','$var8','$var9','$stream','$var10','$var11','$var12','$var13','$var14','$var15' ,'$target_path')";
$r=mysqli_query($con,$query);
$query="INSERT INTO studlogin (stuid, username, passconf ,password) values ('$var1 ',' $var2 $var1','$invoiceid$var2','$invoiceid$var2')";
$r=mysqli_query($con,$query);
}else{$i=$i+1;}
if($r)
{
$duery= "select invoiceid from studentreg order by invoiceid desc";
$sql=mysqli_query($con,$duery);
$row =mysqli_fetch_array($sql);
$lastid=$row['invoiceid'];
if(empty($lastid))
{
$number="aba-as001";
}
else
{
$idd=str_replace("aba-as", "",$lastid );
$id=str_pad($idd+1, 3,0, STR_PAD_LEFT);
$number='aba-as'.$id;
}
echo "<center><h3>Succesfully Register! Thank You! .</h3></center>"."<br>";?> }
```

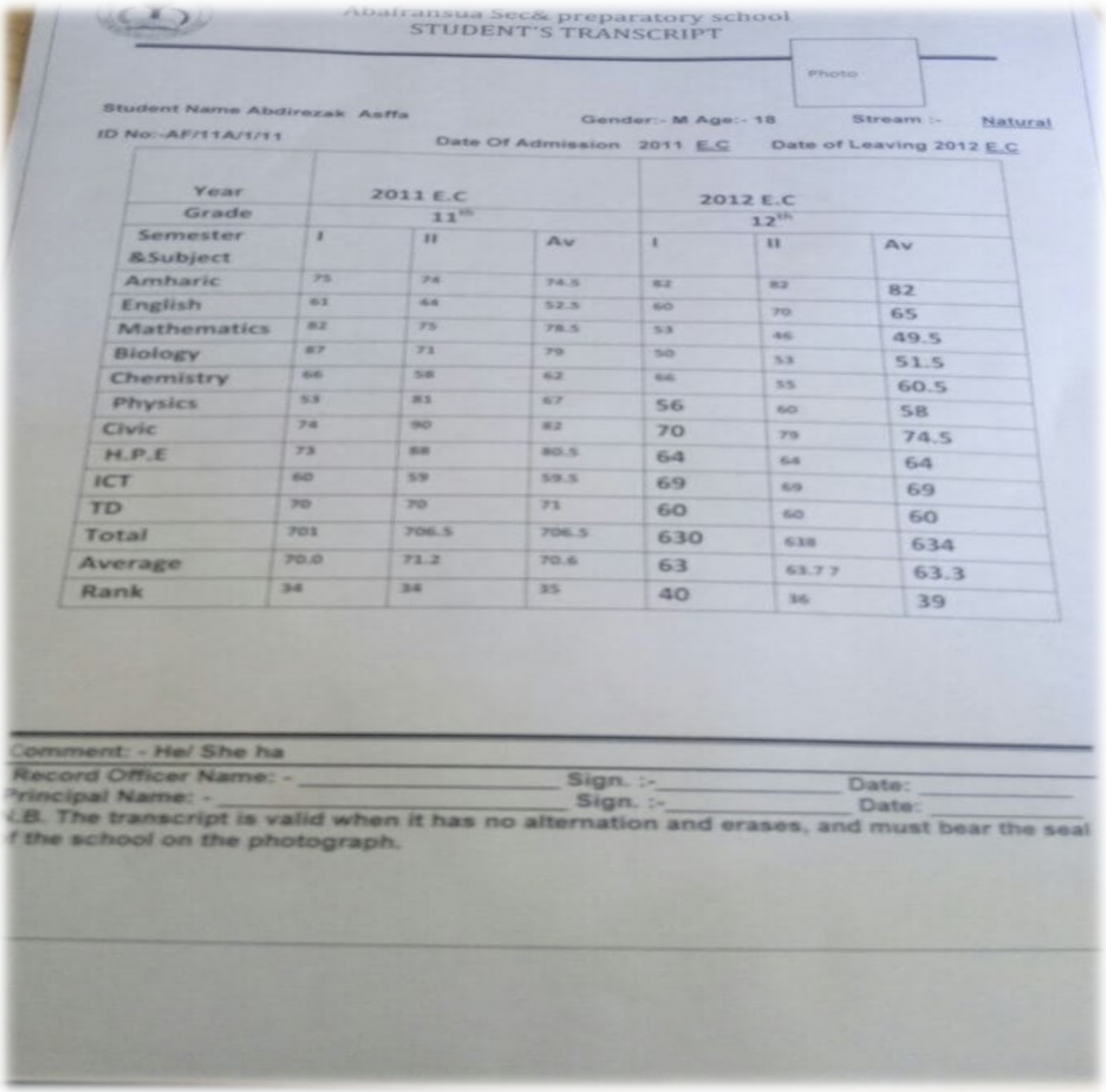
Existing System Forms and Reports.



IN 2013 E.C ABAFRASUA SECONDARY AND PREP. SCHOOL TIME TABLE FOR G-12th

Days	Period	12A	12B	12C	12D	12E	12F	12G	12H
Mon	1	Chem	Civ	Geo	Eco	Ict	His	Amh	GB
	2	PHY	Math	Eco	Eng	Civ	Math	His	Eng
	3	TD	Eng	Civ	His	Eco	Amh	Math	Geo
	4	Math	Chem	Eng	Amh	Hpe	Eco	Civ	His
	5	CIV	TD	Math	Geo	Eng	Civ	Eco	His
	6	Bio	Phy	His	Civ	Math	Eng	Geo	Amh
Tue	1	Bio	Math	Amh	Eng	Eco	Civ	Geo	Civ
	2	Ict	Bio	His	Hpe	Math	Geo	Eco	GB
	3	Phy	Hpe	Eco	Math	His	Eng	Eng	Ict
	4	Chem.	Phy	Math	His	GB	Eco	Geo	Eng
	5	Eng	Bio	Eng	GB	Geo	Hpe	Eco	Math
	6	Math	Chem	Geo	Eco	Eng	Math	Civ	His
Wen	1	Bio	Civ	His	Math	Geo	Eng	GB	Eco
	2	Chem	Bio	Civ	Geo	Math	Eco	Eng	His
	3	Eng	Chem	Geo	Ict	Eco	Civ	Hpe	Math
	4	Civ	Phy	GB	His	Eng	Geo	Eco	Civ
	5	Math	Amh	Eco	Eng	GB	His	Civ	Geo
	6	Phy	Eng	Civ	Eco	His	Geo	Math	Eng
Thu	1	Amh	Math	Eng	Eco	Geo	GB	His	Math
	2	Phy	Bio	His	GB	Eco	Eng	Geo	His
	3	Chem	Eng	Math	Geo	Civ	GB	His	Eco
	4	Bio	Phy	Geo	Civ	His	Math	GB	Eco
	5	Civ	Chem	Ict	His	Eng	Geo	Math	Hpe
	6	Eng	Civ	Eco	Math	Geo	His	Eng	Geo
Fr	1	phy	Chem	GB	Eng	Eco	His	Geo	Math
	2	Eng	Chem	His	Math	Geo	Eco	Ict	Amh
	3	Hpe	Phy	Math	Eco	His	Ict	Eng	Geo
	4	Math	Ict	Eng	His	Amh	Geo	Eco	Civ
	5	Bio	Chem	Hpe	Geo	Civ	Amh	Eng	His
	6	Chem	Eng	Geo	Civ	Math	Eco	His	Eng

Figure 5. 9: time table



Abafransua Sec& preparatory school
STUDENT'S TRANSCRIPT

Photo

Student Name Abdirezak Asffa Gender:- M Age:- 18 Stream :- Natural
ID No:-AF/11A/1/11 Date Of Admission 2011 E.C Date of Leaving 2012 E.C

Year	2011 E.C			2012 E.C		
Grade	11th			12th		
Semester &Subject	I	II	Av	I	II	Av
Amharic	75	74	74.5	82	82	82
English	61	44	52.5	60	70	65
Mathematics	82	75	78.5	53	46	49.5
Biology	87	71	79	50	53	51.5
Chemistry	66	58	62	66	55	60.5
Physics	53	81	67	56	60	58
Civic	74	90	82	70	79	74.5
H.P.E	73	88	80.5	64	64	64
ICT	60	59	59.5	69	69	69
TD	70	70	71	60	60	60
Total	701	706.5	706.5	630	638	634
Average	70.0	71.2	70.6	63	63.7 7	63.3
Rank	34	34	35	40	36	39

Comment: - He/ She ha
Record Officer Name: - _____ Sign. :-_____ Date: _____
Principal Name: - _____ Sign. :-_____ Date: _____
N.B. The transcript is valid when it has no alternation and erases, and must bear the seal of the school on the photograph.

Figure 5. 10: transcript