Class Schedule Management System for Secondary School

A Project Report

Submitted By:

Name	ID
Minalu Chalie	TER/1250/06
Bazezew Melaku	TER/1212/06
Yenenesh Dires	TER/1284/06
Endalamaw Malkie	TER/1222/06

in partial fulfillment for the award of the degree of

BACHELOR OF SCINCE IN INFORMATION TECHNOLOGY

Under the guidance of

Instructor Solomon Asemu

…………………………

ADVISOR SIGNATURE

DEPARTMENT OF INFORMATION TECHNOLOGY

COLLAGE OF TECHNOLOGY

DEBRE MARKOS UNIVERSTITY

DEBRE MARKOS

May 2009 E.C

Abstract

The manual system of preparing time table in high school within large number of students is very time consuming and usually ends up with various classes clashing either at same room or with same teachers having more than one class at a time. To overcome these problems people usually take the previous year's timetable and modify it. But still it is a tedious job to incorporate changes every time. To overcome all these problems, we propose to make an automated system. The system will take various inputs like details of students, subjects and class rooms and teachers available. Depending upon these inputs, it will generate a possible time table, making optimal utilization of all resources in a way that will best suit any of constraints or rules.

So our aim is to develop a general-purpose automated system which can effectively generate automatic high school class schedules when given with several input constraints.

Acknowledgement

First and foremost we would like to thank God for giving us the necessary guidance, energy and drive to complete the project.

We owe a great deal of gratitude to our advisor MSC Solomon Asemu for his unfailing patience unreserved and valuable advises during the course of writing this paper. He gave us support to each of the difficulties encountered starting from writing the project proposal to the finalization of this portion of the project, filled with sharp intelligence and understanding.

Thanks to those who gave us information for the completion of our project, to Deber Markos town teacher association for their valuable comment and helpful information sharing.

Our Thanks is also forwarded to individuals who directly or indirectly supported us during our project work.

Last but not least, we would like thank our families and friends for providing us priceless support and encouragement during our study.

Table of Contents

List of Figure

List of Table

List of acronyms

HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
PHP	Hyper Text Preprocessor
UML	Unified Modeling Language
UC	Use Case
UI	User Interface
MYSQL	My Structured Query Language
WAMP	Windows Apache MySQL

Chapter One

1. Introduction

1.1. Introduction

Scheduling is the process of arranging, controlling and optimizing work and workloads in a production process or manufacturing process. Timetabling concerns all activities with regard to producing schedule that must be subjective to different constraints.

The problem is to manage a time table of all classes of school, according to the teacher, for any school class scheduling management is very complex and time consuming task. The school used the manual way of preparing the class schedule. With the manual system, more time and labor force is required to plot, arrange, and revise the class schedules, room utilization and teacher load provided by the program committee. With these problems, the project team will propose a project of creating an automated class scheduling system that uses its own algorithm to assign resources in most efficient and accurate manner. This schedule management system module automatically creates schedules for the class and the class teacher.

1.2. Background of the project

Education system forms the backbone of every nation. And hence it is important to provide a strong educational foundation to the younger generation to ensure the development of open minded global citizens securing the future for everyone.

Secondary school education follows eight years of primary education and is for the children age 14 and above. 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.

1.3. Statement of the problem

Secondary schools have many problems regarding paper based system. Most of Ethiopia high schools still use a manual system to mining class schedule. Although most of the schools administrative work has not been computerized, the teacher time table scheduling is still mostly done manually due to its technical difficulties. The manual scheduling of teacher-timetable requires considerable time and efforts.

There are many existing problems in the current class scheduling system due to different reasons, they are listed as follows:

- ✓ It takes more resources and time for mining scheduling.
- ✓ Difficult to manage class and laboratory room schedule.
- ✓ Overlapping of class due to tiredness of checking each schedule.
- ✓ Less security since the system is manual.
- ✓ It takes more employees and the working condition is the tides.
- ✓ Users may not get the information at the required time.
- ✓ It has data or information redundancy.

1.4. Objectives of the project

The objective of this project is to clearly understand the problem of manual class scheduling management system in secondary school and develop web based class schedule system which minimize work load and time consuming, improve the quality of service delivered to the student.

1.4.1. General objectives

The general objective of this project is to develop automatic class scheduling management system for secondary school.

1.4.2. Specific objectives

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

- ✓ To create a database that used to store subject, teacher, room and section
- ✓ To add users of the system.
- ✓ To enable for post notice or information
- ✓ To enable for storing required input parameters or information.
- ✓ To develop an algorithm that accepts or take input parameters
- ✓ To enable for students and teachers to search and retrieve schedule.

- ✓ To enable for post exam schedule.
- ✓ To make edits whenever needed or required.

1.5. Scope of the project

Most secondary schools perform class schedule management system using the manual processing system. However, the manual processing has its own problems, so the school requires automatic class scheduling to perform their task efficiently.

Generally, the scope of the project team will to develop automatic class scheduling management system for secondary schools that can be providing the following functionalities:

- ❖ The system will generate a general report
- ❖ Assign listed classroom to each section.
- ❖ Edit assigned class and laboratory room information.
- ❖ Store class and laboratory room information to the database.
- ❖ Record and edit teacher information to and from database.
- ❖ Edit listed class and laboratory rooms from the database.
- ❖ View, update and post exam schedule.
- ❖ Edit and store assigned teacher information.
- ❖ View class and laboratory room schedule.
- ❖ Add, view and block of the user of the system.

1.6. Limitation of the project

The automate web based class schedule management system only automates sixth or seven period per a day for the each available section. And also the automate system does not jump a holiday time. Due to shortage of enough time to develop the system, because we are take simultaneously other course.

1.7. Significance of the project

Currently the school operates a manual record system. With the introduction of an automated based class scheduling management system the following will be achieved;

- Reduce manual process and administrative cost to maintain existing systems.
- Reduce time, cost and redundancy of information or data.
- Improving efficiency, control and security of the class schedule management system.
- Users view notice and information easily.
- Reduce manpower.
- Students and teachers view exam schedule, class and laboratory schedule.
- Avoid class and laboratory room schedule overlapping.
- Improve flexibility in timetable construction.
- Easily post exam schedule.
- Easily change exam schedule.
- Easily add necessary information or notice.

1.8. System requirements

In order to develop a web base system, it is very important to choose the correct hardware, software and technology. Below are some explanations of the hardware, software and technology chosen as development tools to the class scheduling management system using algorithm.

1.8.1. Hardware Requirements

The Followings are hardware requirements for developing the system such as:

- ✓ Computer: used to develop the system or application.

  Specifically:-

  - ➢ Intel (R) Core i3-4160 CPU 3.60GHZ
  - ➢ 4GB RAM
  - ➢ 450GB hard disk space or higher
- ✓ 8GB flash: For store the data and secure for back up

1.8.2. Software Requirements

There are many software requirements of the class scheduling management system.

- ✓ Edrawmax: For draw diagrams.
- ✓ Notepad++: For text editor
- ✓ Window -7: for running and processing the tasks
- ✓ Mozilla Firefox: For displaying the system
- ✓ Microsoft Power Point 2010: For presentation
- ✓ Microsoft office word 2010: For writing documentation.

1.8.3. Programming and database tools

The following sections discuss the programing language, database technology and server technology used to develop the class scheduling system.

- ➢ Server side programming script, PHP

  Because:

  - ✓ It is open source
  - ✓ Human-friendly language.
  - ✓ Ease of Use: PHP is easy to learn compared to many other scripting languages.
  - ✓ Speed
  - ✓ PHP can be easily fixed directly into, HTML and CSS
  - ✓ PHP is platform independent.
  - ✓ Easy to understand
  - ✓ It resists the virus than other language
- ➢ Database system, MySQL or Wamp server

  Because:

  - ◆ Open –source
  - ◆ Security
  - ◆ Easy connection to programming to indicate
- ➢ Client side language, HTML, CSS, JavaScript

1.9. Data collection methodology

Data collection methodologies are methods used to collect different data from different data sources (documents, users and organizations etc.)

The following are the data collection methods used for requirement

Primary data source

- ✓ **Interview**: We used interview as one of the major data collection methods. During the interview our team got different necessary information from the school director.
- ✓ **Observation:** as part of the learning process and as a student we have gone through the scheduling process.
- ✓ **Document Analysis**: we have analyzed different documents from the nigus Teklehaymanot School and other schools.

Secondary source data

- ✓ **Internet:** Internet aids us to see the available sample on the internet and to download different types of tutorials which help us in doing the project.

1.10. Feasibility study

This analysis helps the organization in determining whether they should proceed with a project or not.

1.10.1. Technical feasibility

The development process of web based class schedule management system will be technically acceptable to the organization since the system can develop using available software technology and also upgrade after developed.

The technical needs of the system that satisfy the user requirement may include:

Front-end selection:

- ➢ Easy to manipulate the system.
- ➢ It must have a simple graphical user interface that assists employees.
- ➢ Must provide excellent reporting features with good printing support.
- ➢ Platform independent.
- ➢ Easy to access and searching

According to the above stated features we select the combination of CSS and HTML developing front end of the system.

Back-end Selection:

- Multiple user support.
- Efficient data handling.
- Efficient data retrieval and maintenance.
- Platform independent.
- Easy to install.
- All works support the front-end of the sytem

According to above stated features we selected PHP and MySQL as the backend. And to implement the web based class schedule management system, there will be network infrastructure within different offices of the school.

1.10.2. Operational feasibility

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. This system has been operationally feasible and the system has been user friendly. The essential questions that help in testing the operational feasibility of a system are following.

- **Does management support the project?** The school director makes a strong commitment to implement the project because they know how tiresome and difficult the process of scheduling is. They provide necessary information and resource to the project, all these shows that their willingness to the success of the new system.
- **Are the users not happy with current business practices? Will it reduce the time (operation) considerably?** If yes, then they will welcome the change and the new system. So anything that will reduce the time and resource and also enhance the current system will be defined acceptable by the user.

1.10.3. Economic Feasibility

Economic analysis is most frequently used for evaluation of the effectiveness of the system. It involves the cost incurred on the system development team, estimated cost of hardware and software, the cost of performing feasibility studies and so on.

The system will not be requiring much more cost beyond the capacity of the schools capital when automating the system such as cost for networking equipment's, hardware and other infrastructures. Since the Schools are well equipped with the required hardware and software, the project was found to be economically feasible. Therefore, our system will be acceptable, economically towards solving problems.

1.10.4. Legal feasibility

The system will not have any conflict with the rule and regulation of secondary schools policy. The system is legally acceptable since it respects the school rule and regulation as well as other constitutional laws of the country.

Chapter Two

2. System analysis

2.1. Overview of existing systems

This project emphasizes on school class schedule management system in Ethiopian secondary schools. At the beginning of each academic year, which class starts in September, the students get registered and assigned rooms. As we know all high school have its own class scheduling, manage and maintaining these will be difficult. Schools are currently using a manual system to assign class and laboratory room schedule, assign an exam schedule, delivering schedule result, storing, managing and controlling schedules. The existing systems have lack of a well-organized database system and data are not easily accessible due to its placed in different offices. In the existing system the school program committee assign exam, class and laboratory schedule through paper.

The school director will assign classrooms to sections accordingly by allocating it in a time frame of periods and shift. The day is divided in periods which will start at 2:00 AM the morning until 12:00 PM. If there is any class confliction the program committee will rearrange the subject and time frame again and again until it become conflict free. Then first draft of the scheduling will be posted and if there is a problem for teachers, the program committee again makes the rearrangement and the second draft will roll out. If there is still a conflict correction will be made and the final draft will be posted which will be used for the whole term.

The same procedure will be followed for the examination scheduling, but differ a little bit from class scheduling because the examination is a onetime event so the exam scheduling is temporary.

Users of existing systems

In existing system, there are 6 users. These are director, department head, record officer, program committee, teacher, and student.

Record officer

Description: A record officer is the person who registers teacher. He or she has responsible for keeping and managing records.

Director

Description: A director is a person, who manages class and laboratory rooms, school information and section.

Department head

Description: Department head is a person who assigns teachers to each section.

Program committee

Description: Program committee refers the person or group of person who develop class and laboratory room schedule.

Teacher

Description: Teacher refers a person who views class and laboratory room schedule and also view exam schedule.

Student

Description: Student refers the person who view class and laboratory room schedule and also view exam schedule.

2.2. Overview of the proposed system

The proposed automated class scheduling system has solutions to the problems that exist in the existing manual system. They system will provide many new functionality and flexibility which couldn't be achieved by the manual system. In the automated system user will be able to view the planned schedule in an easily readable format, class conflict will be avoided, more than one class assignment for teacher will be detected and handle by the system.

The automated class scheduling system is a web based system which is developed for teachers and students, they access online timetable scheduling. It would be assign class room scheduling, lab scheduling and examination scheduling. There will be a director who will provide the input parameters and impose a change if there one to be made after scheduling. The director will have a user name and password for security purpose.

Generally, this proposed system will generate class schedule automatically to allow students and teachers access online. And also it will be keeping record data accurately; easy and fast accessing information, reduce manpower, reduce cost and saving time.

2.3. System requirement specifications

After all the effort to understand the existing system and features of the new system, it is now time to organize and specify requirements in the form of functional and non-functional requirements.

2.3.1. Functional requirements

A functional requirement defines the capabilities of a system that must be able to perform successfully. The functional requirement is the study of what a system should be able to do, the functions it should perform and describes the interactions between the system and its environment. And also explain what has to be done by identifying the necessary task, action or activity that must be accomplished. [1]

The developing system is expected to provide the following functionalities:

- ✓ Teachers and students view generate class schedule.
- ✓ Display information about the schools notice or information
- ✓ Students and teachers view exam schedule.
- ✓ Allow department head assign classrooms to each section
- ✓ Allow director register subject and its credit hours
- ✓ Allow director head add classroom and section
- ✓ Allow record officer register teacher information
- ✓ Allow director register period detail information
- ✓ Allow post exam schedule

2.3.2. Non Functional requirements

Though the system must support all the functional requirements, non-functional requirement is a systematic approach to build quality into the system software. The web based class scheduling management system should accommodate the following extra requirements.

Authentication

Proper implementation of the system will be used to avoid accessing unauthorized person. The security service provided by the system will maintain the security of the system. Users will have their own password and username to access the system.

Performance

The system uses an algorithm, it should perform the scheduling task in a short response time and high rate of processing with valid scheduling information.

Usability

The system is designed to have user-friendly interfaces and easy navigation, which enhances users' efficiency. It is also designed in such a way that users can easily learn how to interact with the system.

Authorization

The proposed system should be authenticating the user of the system by asking username and password and determine which users are authorized to access or perform specific activities or operations.

Modifiability

The system should be modifiable by authorized users of the system whenever it needed.

Reliability

The system is expected to be used smoothly without being corrupted and frequently failure. The system should satisfy the user.

2.3.3. Business rules

A business rule is a statement that describes a business policy or procedure that must be fulfilled and obligated in order the system will function properly and effectively. The process of identifying business rules is often iterative of this system, where rules begin as general statements of policy of the system.

BR1: Department head must be assigning his department teacher for each section.

BR2: The user of the system must have a legal account to fill required information.

BR3: The total number of periods in the week is proportional to the sum of credit hours for each subject.

BR4: Teacher teaches 22 hours ½ or 30 credit hour per week.

BR5: Assign section for class room not more than or above two sections.

BR6: In class room not assign two sections at the same time.

BR7: Maximum number assign sections in laboratory room depend on lab hours per week.

BR8: To view class schedule first assign class and laboratory rooms.

2.4. System requirement analysis

2.4.1. Actor and use case identification

Actor: An actor represents a type of users of the system or external system that plays a role in one or more interactions with our system.

Use case: A use case describes a sequence of actions that provide a measurable value to an actor. A use case is drawn as a horizontal ellipse on a UML use case diagram.

2.4.1.1. Actor identification

The proposed class schedule management system has 7 users. These are:-

Record officer: The person who register teacher information.
Director: A person who registers class and laboratory rooms, post school information.

Department head: A person who assigns teachers to each section.

Teacher: A person who views class and laboratory room schedule and also view exam schedule.

Student: A person who view class and laboratory room schedule and also view exam schedule.

Web admin: A person who manages and controls the system.

Program committee: The people who set period and assign room with each section.

2.4.1.2. Use case identification

Table 1 Use case identification

Use case name	Use case Id	Interact with
Register teacher	UC01	Login
Register room	UC02	Login
Edit record room	UC03	Login
Assign teacher	UC04	Login
Register record officer	UC05	Login
Post notice	UC06	Login
Edit notice	UC07	Login
Update subject	UC08	Login
Edit teacher	UC09	Login
Register section	UC10	Login
Edit section	UC11	Login
Edit record officer	UC12	Login
Edit assigned teacher	UC13	Login
Search class schedule	UC14	
Post exam schedule	UC15	Login
Search exam schedule	UC16	
Edit exam schedule	UC17	Login
Create account	UC18	Login
Search account	UC19	Login

Bock account	UC20	Login
Edit assigned teacher	UC21	Login
Add period	UC22	Login
Edit period	UC23	Login
Add subject	UC24	Login
Login	UC25	
Logout	UC26	Login

2.4.1.3. Use case diagram

Use Case diagrams show the various activities the users can perform on the system. The system is something that performs a function. They model the dynamic aspects of the system. A use case diagram illustrates a set of use cases for a system, the actors of these use cases, the relations between the actors and these use cases, and the relations among the use cases. [2]

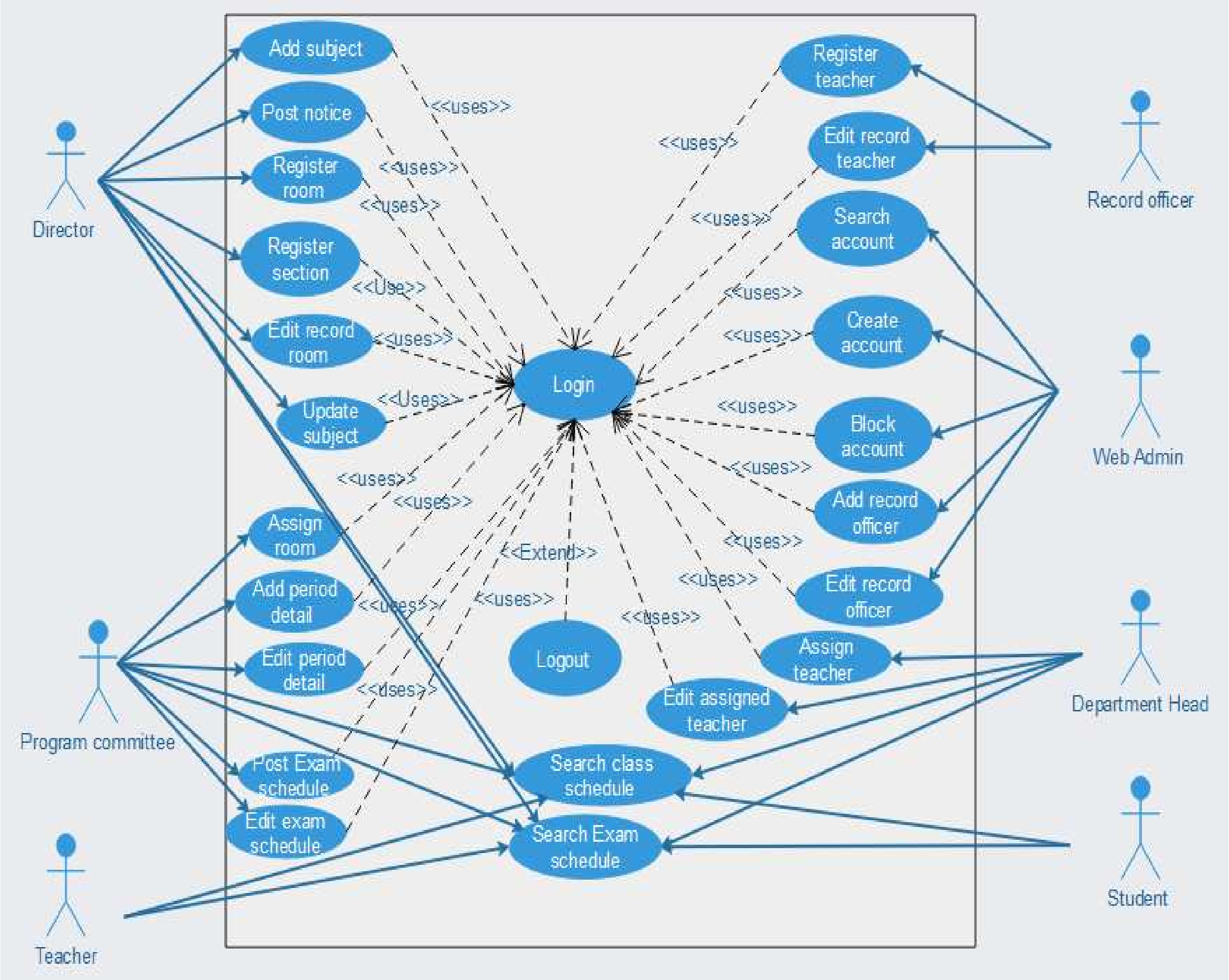


Figure 1 Use Case Diagram

Table 2 Use case description of register teacher

Name	Register teacher
Id	UC01
Actor	Record officer
Description	Record officer should able to register new teacher
Precondition	Record officer should have their own legal account
Post condition	The system can take or accept teacher information to generate

	schedule report
Basic flow	1. Record officer login to the system 2. The system directs to record officer page 3. Record officer click on register teacher link 4. The system displays teacher registration form 5. Record officer fill form and press register button 6. The System displays successful register 7.Use case ends
Alternative flow	A1. The system returns the record officer to login form to reenter valid username and password. **A5**. The system returns the record officer to register teacher form to fill missed required fields.

Table 3 Use case description for register room

Name	Register room
Id	UC02
Actor	Director
Description	This use-case enables the director to register class and laboratory room information
Precondition	The director should get their own legal account from system admin to login and after login register room.
Post condition	The system takes or accepts this class information to generate automate class schedule.
Basic flow	1. Directly login to the system 2. System directs to director page

	3. Director click on register room link 4. System display room registration form 5. Director fill form and press add button 6. The System display successful register 7.Use case ends
Alternative flow	A1. The system returns the director to login form to reenter valid username and password. A5. The system returns the director to register room information form to fill missed required fields and continue at step 4.

Table 4 Use case description for register section

Name	Register section
Id	UC10
Actor	Director
Description	This use case enable to director register section that present school.
Precondition	Director should have their own account login into the system
Post condition	The system takes register section that used to generate automate class and laboratory schedule
Basic flow	1. Director login to the system 2. System directs to Director page 3. Director click on add section link 4. System display section registration form 5. Director fill required field and press add button

	6. The system display successfully register section 5.Use case ends
Alternative flow	A1. System can display error message to enter valid username and password A6. System can be display “required information’s are not filling” and process continue at step 4.

Table 5 Use case description for add subject

Name	Add subject
Id	UC24
Actor	Director
Description	Director should able to register subject and its credit hours
Precondition	Users should have their own account login into the system
Post condition	The system takes subject and its credit hours to generate schedule report
Basic flow	1. Director login to the system 2. System directs to director page 3. Director click on add subject link 4. System should display subject registration page 5. Director fill required form and press add button 6. System display successfully register subject 7. Use case ends
Alternative flow	A1. System can display error message to enter valid username and password A6. System can be display “required information’s are not

	filling”.

Table 6 Use case description for post exam schedule

Name	Post exam schedule
Id	UC15
Actor	Director
Description	School director should able to post exam schedule
Precondition	School director should have their own account login into the system
Post condition	Users view exam schedule
Basic flow	1. School director login to the system 2. System directs to School director page 3. Users click on post exam schedule link 4. System can display exam post page 5. School director fill form 6. School director click post button 7. The System display successful post 8.Use case ends
Alternative flow	A1. System can display error message to enter valid username and password A7. System can be display “cannot be successfully post”.

Table 7 Use case description for register assigned teacher information

Name	Assign teacher
Id	UC04
Actor	Department head
Description	Department head should able to register teacher replacement information system
Precondition	First department head should have their own legal account login to the system
Post condition	The system can take assigned teacher information to generate automate class schedule.
Basic flow	1. Department head login to the system 2. System directs to department head page 3. Department head press on add assigned teacher link 4. System display teacher assigned information registration form 5. Department head fill form and press register button 6.The System display successful register 7.Use case ends
Alternative flow	A1. The system returns the department head to login form to reenter valid username and password. **A5**. The system returns to register assigned teacher information form to fill missed required fields.

Table 8 Use case description for add period detail

Name	Add period
Id	UC22

Actor	Director
Description	Director should able to register the starting time of the class and add period information the subject and break time. And also assign how many periods per a day student learn.
Precondition	Users should have their own account login into the system
Post condition	The system take period length to assign automate class schedule.
Basic flow	1. Director login to the system 2. System directs to director page 3. Director click on Add period link 4. System should display period information registration form 5. Director fill required form and press add button 6. System display successfully register period information 7. Use case ends
Alternative flow	A1. System can display error message to enter valid username and password A6. The system returns to Add period information form to fill missed required fields and process to continue at step 5

Table 9 Use case description for post notice

Name	Post notice
Id	UC06
Actor	Director
Description	Director should able to post necessary notice or schedule information to the student and teacher

Precondition	Director should have their own legal account login into the system
Post condition	Users of the system can view current notice or schedule information
Basic flow	1. Director login to the system 2. System can direct to director page and click on post notice 4. System can display notice page 5. Director fill form 6. Director click post button 7. The System display successful post 8. Use case ends
Alternative flow	A1. The system returns invalid username and password A7. The system return " cannot be successfully post"

Table 10 Login use case description

Name	Login
ID	UC25
Actor	Department head, Director, Record officer, Program committee
Description	Users are authenticated and taken on their own user interface
Pre-condition	Users must get an account from the system administrator
Post-condition	The user is authenticated and taken to his/her own user interface

Main flow	1. The user opens the main home page by writing the URL of the website. 2. The system display the Main Home page 3. The user inputs user name and password and submit 4. The system validates the account and displays the user require information. 5. The system directs to his own homepage 6. Use case ends.
Alternate flow	A: If the login name or password is invalid A4.The system displays invalid username or password message A5. The user reenters the username and password A6. The use case continues from step 3

2.4.2. UML Sequence Diagram

A UML Sequence diagram showing the sequence of interactions among objects and used to represent or model the flow of messages, events and actions between the objects or components of a system. Sequence Diagrams are also used primarily to design, document and validate the architecture and interfaces of the system by describing the sequence of actions that need to be performed to complete a task. [3]

Register Teacher

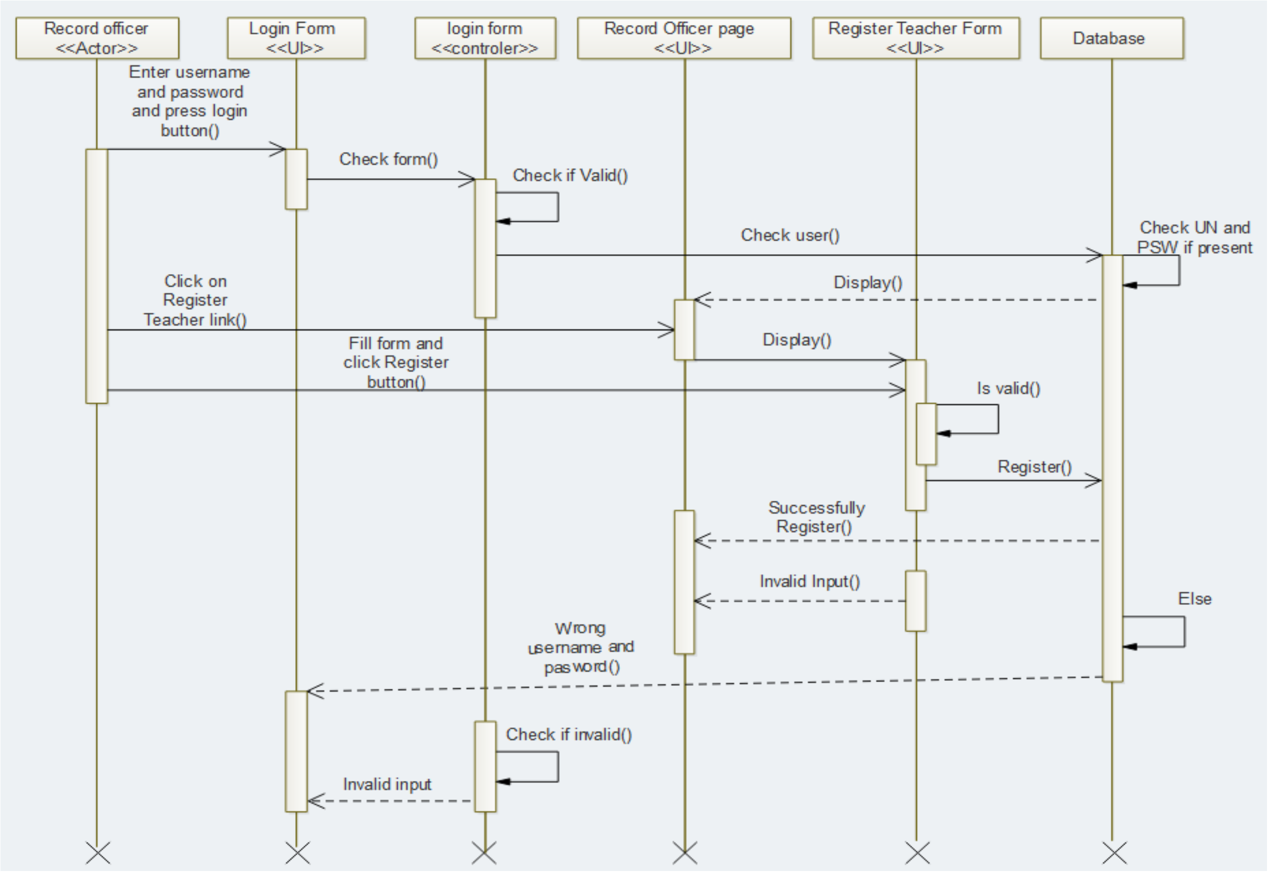


Figure 2 Register teacher sequence diagram

Register Room

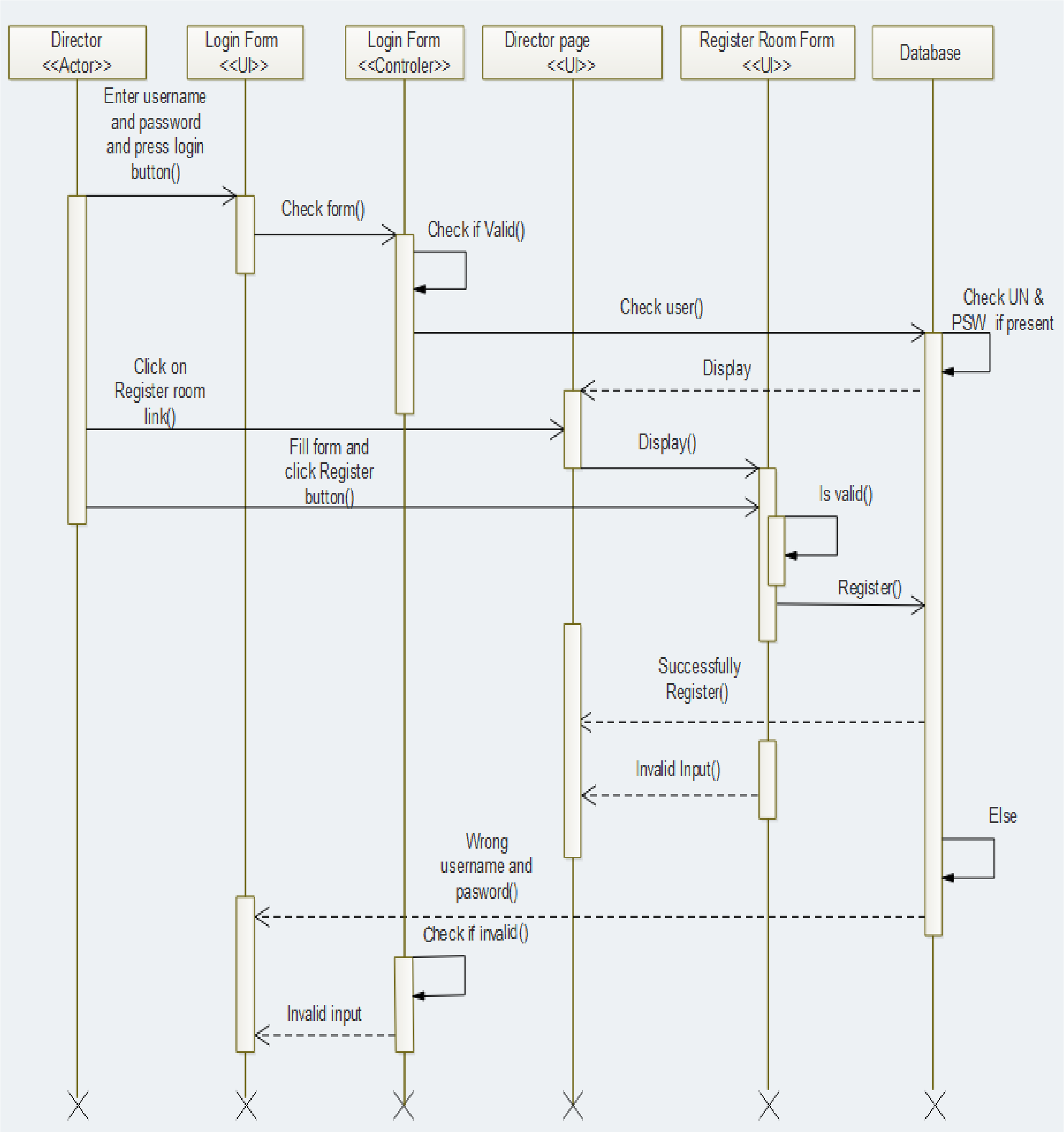


Figure 3 Register room sequence diagram

Register Section

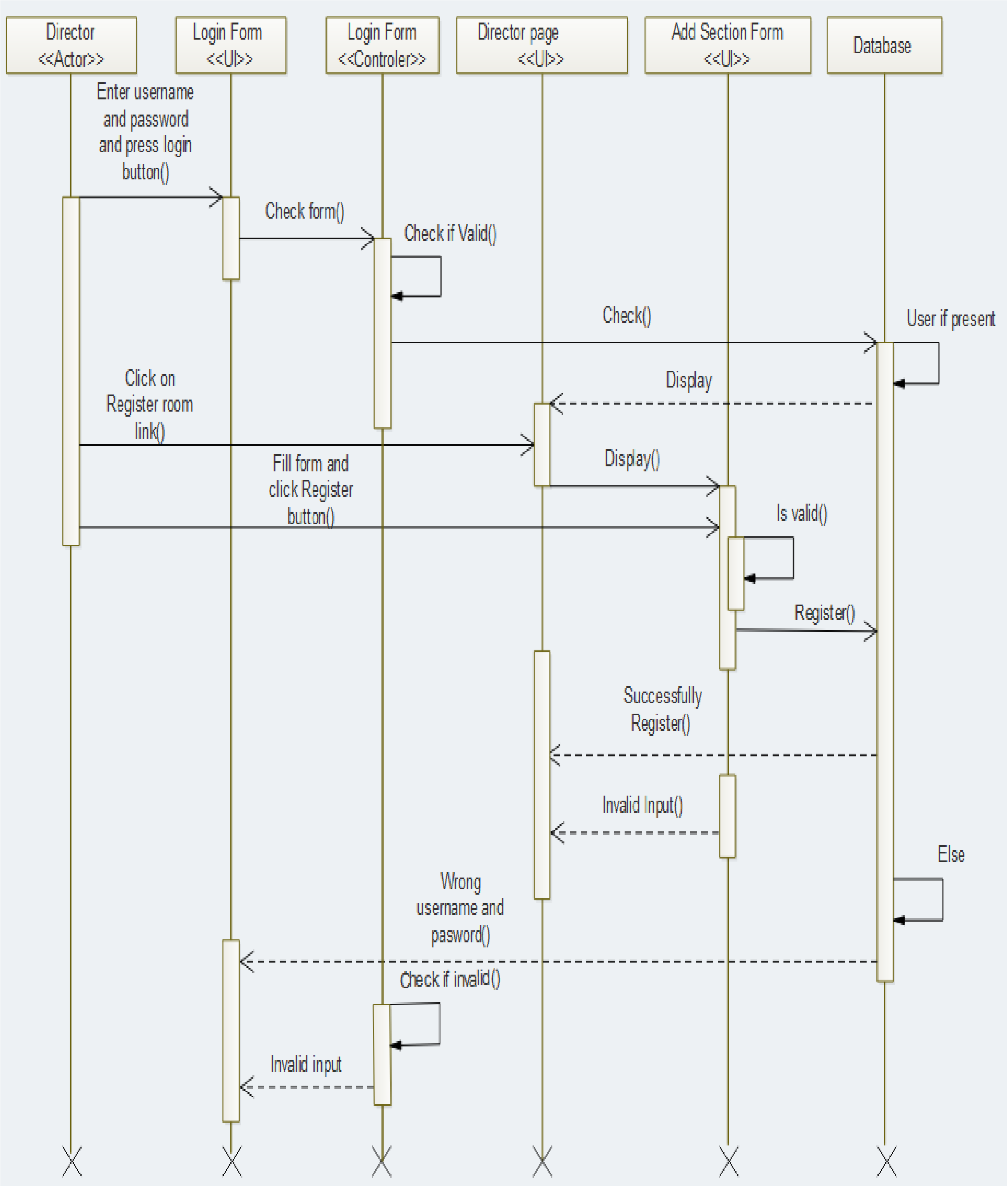


Figure 4 Register section sequence diagram

Add period

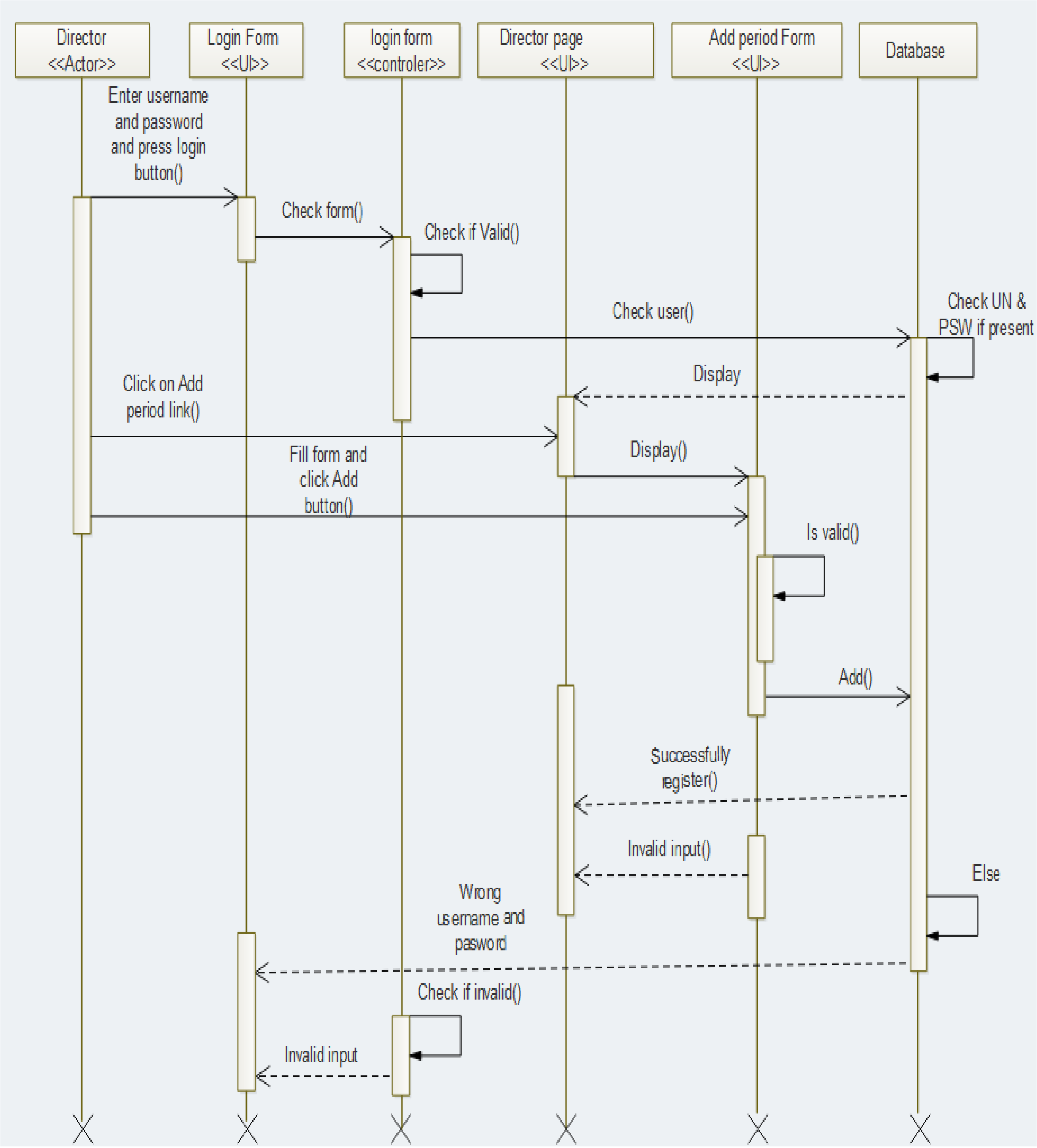


Figure 5 Add period detail sequence diagram

Create Account

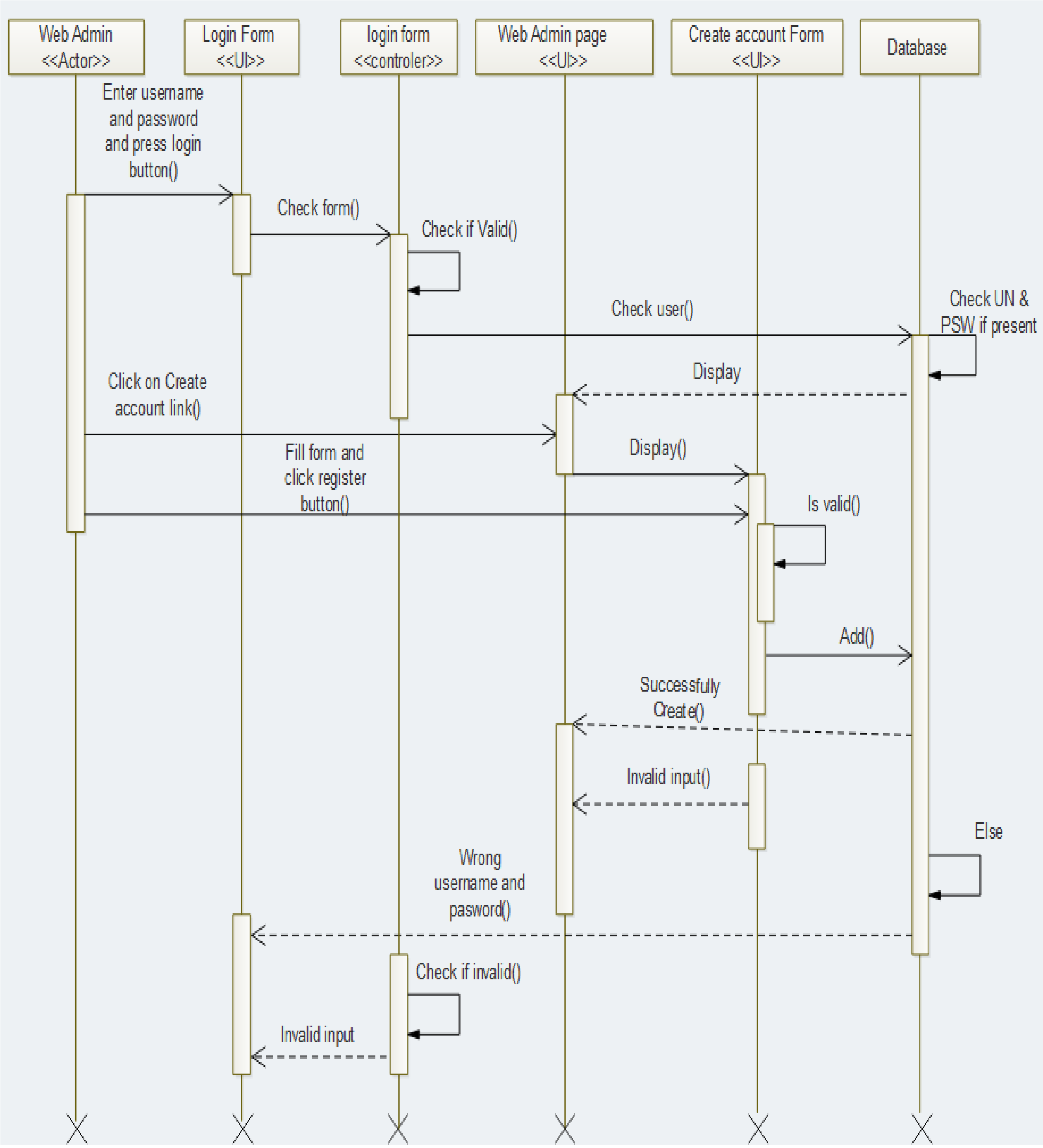


Figure 6 Create account sequence diagram

2.4.3. UML Activity Diagram

Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all types of flow control by using different elements like fork, join etc.

Register Teacher

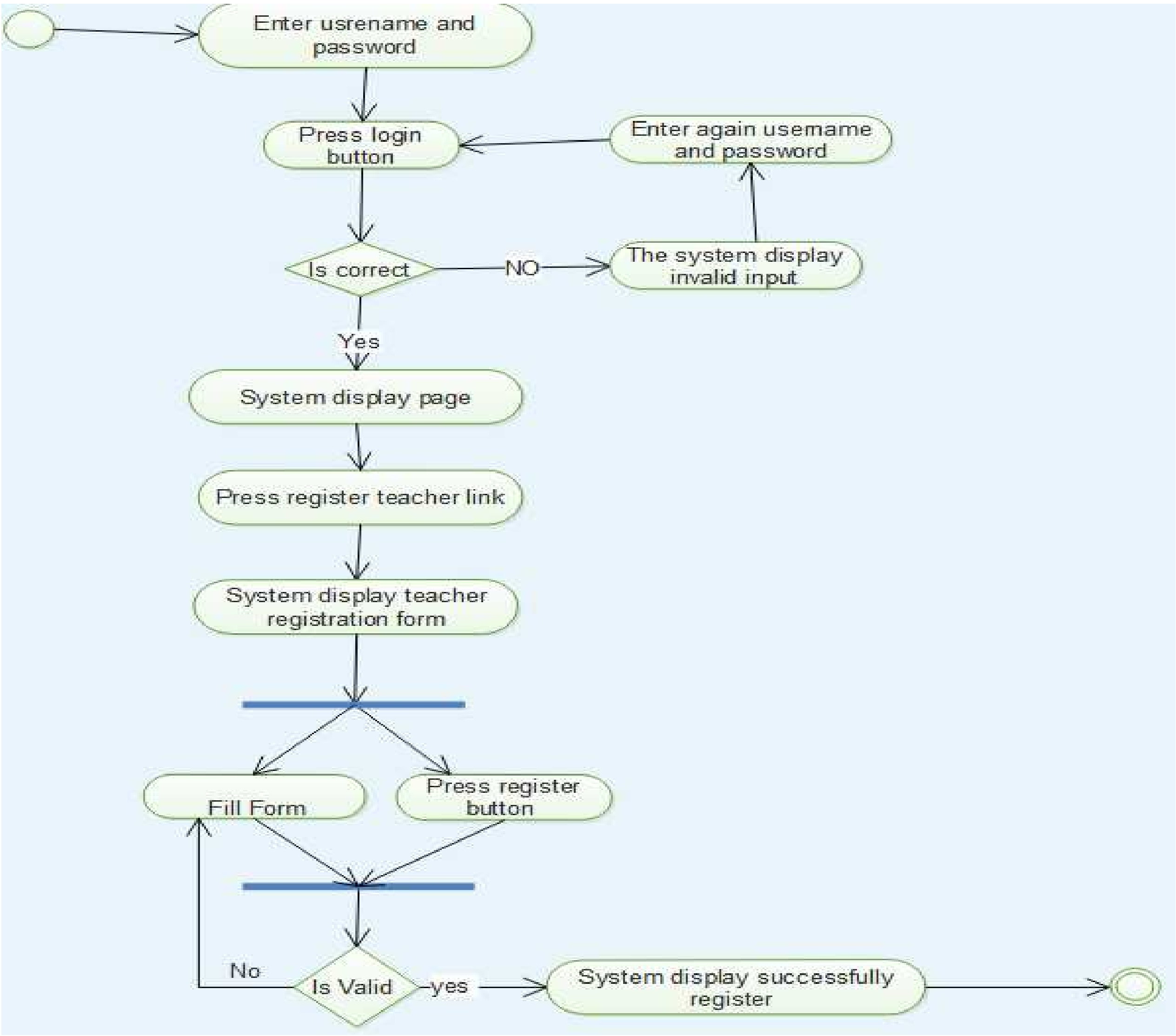

Enter usrename and password
Press login button
Enter again usemame and password
Is correct
NO
The system display invalid input
Yes
System display page
Press register teacher link
System display teacher registration form
Fill Form
Press register button
No
Is Valid
yes
System display successfully register

Figure 7 Register teacher activity diagram

Register Room

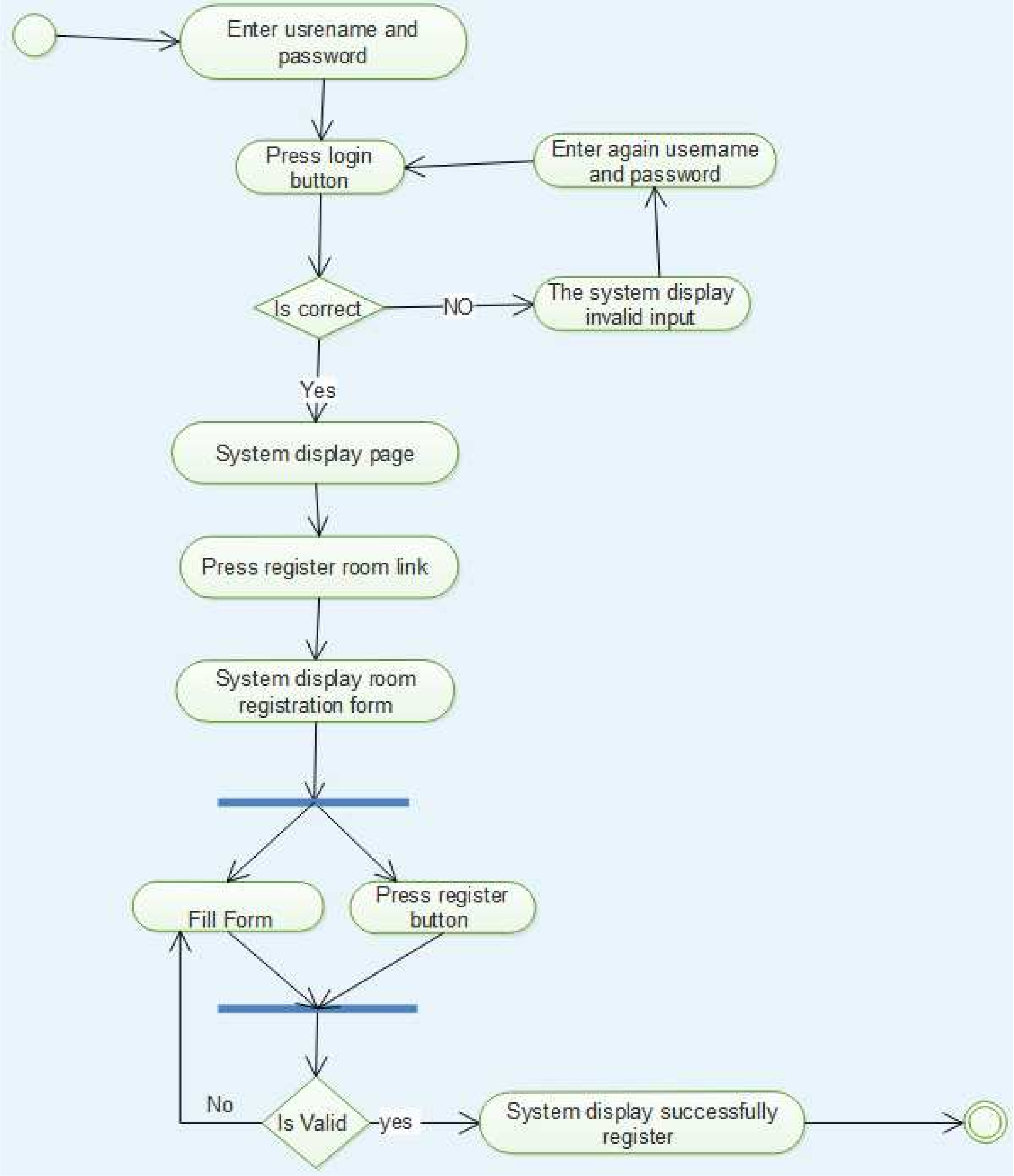

Enter usrename and password
Press login button
Enter again usemame and password
Is correct
NO
The system display invalid input
Yes
System display page
Press register room link
System display room registration form
Fill Form
Press register button
No
Is Valid
yes
System display successfully register

Figure 8 Register room activity diagram

Add Period

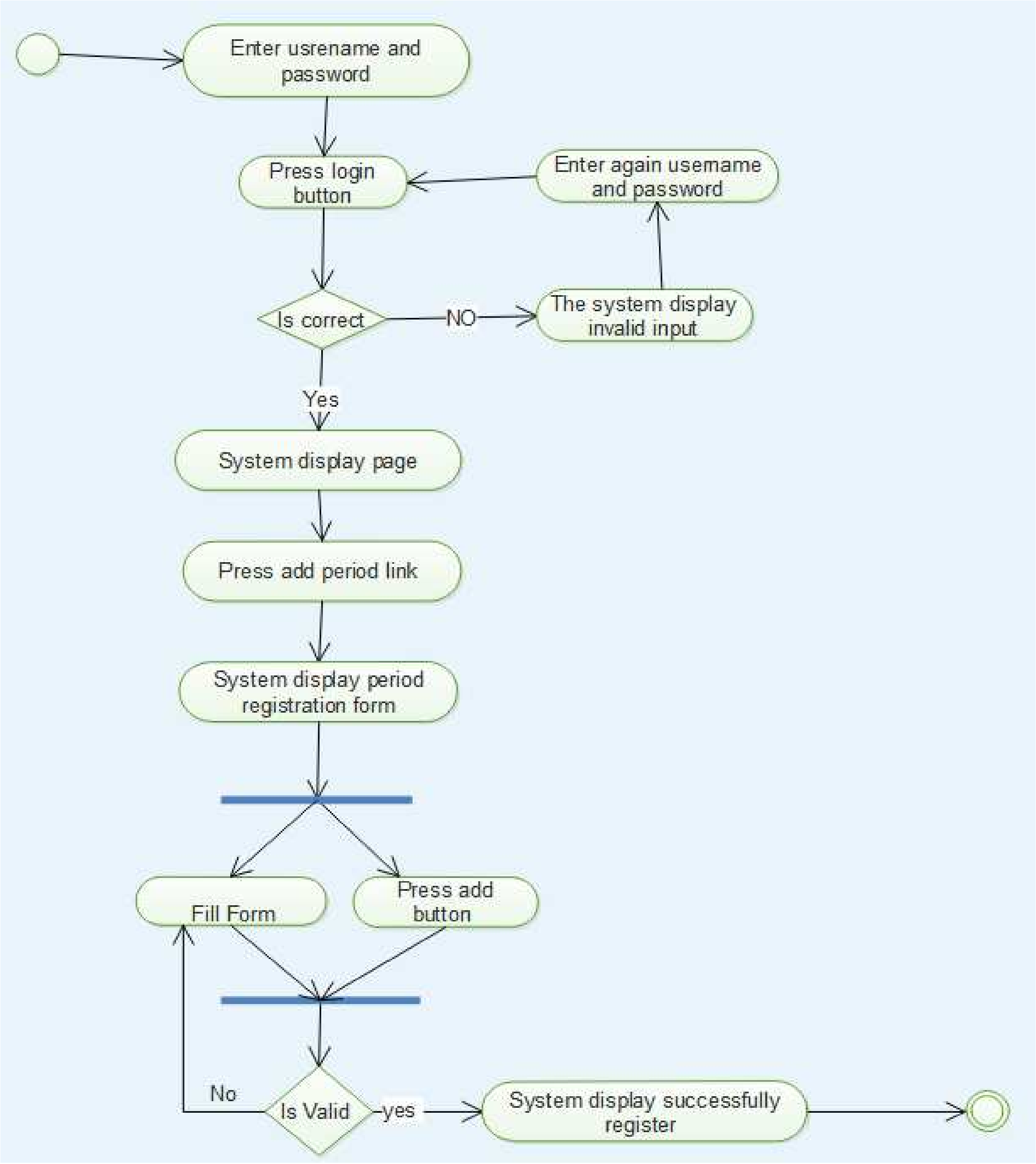

Enter usrename and password
Press login button
Enter again usemame and password
Is correct
NO
The system display invalid input
Yes
System display page
Press add period link
System display period registration form
Fill Form
Press add button
No
Is Valid
yes
System display successfully register

Figure 9 Add period detail activity diagram

Assign Teacher

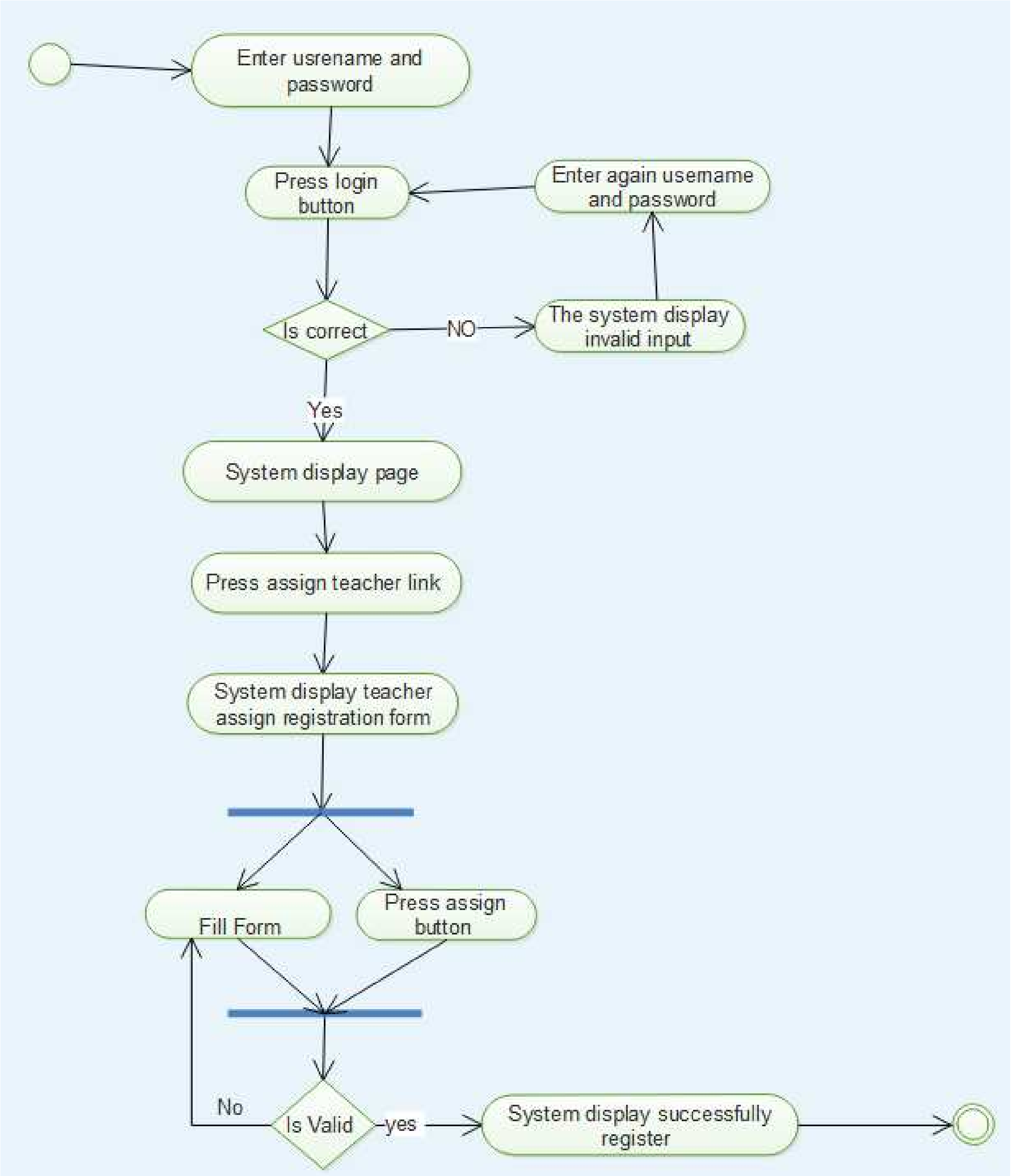

Enter usrename and password
Press login button
Enter again usemame and password
Is correct
NO
The system display invalid input
Yes
System display page
Press assign teacher link
System display teacher assign registration form
Fill Form
Press assign button
No
Is Valid
yes
System display successfully register

Figure 10 Assign teacher activity diagram

2.4.4. Analysis UML class diagram

A class diagram is a static model that shows the classes and the relationships among classes that remain constant over the time. Class is the main building block of class diagrams, which stores and manages information in the system. In the phase of conceptual class modeling we just create or classes and their interrelationship. [4]

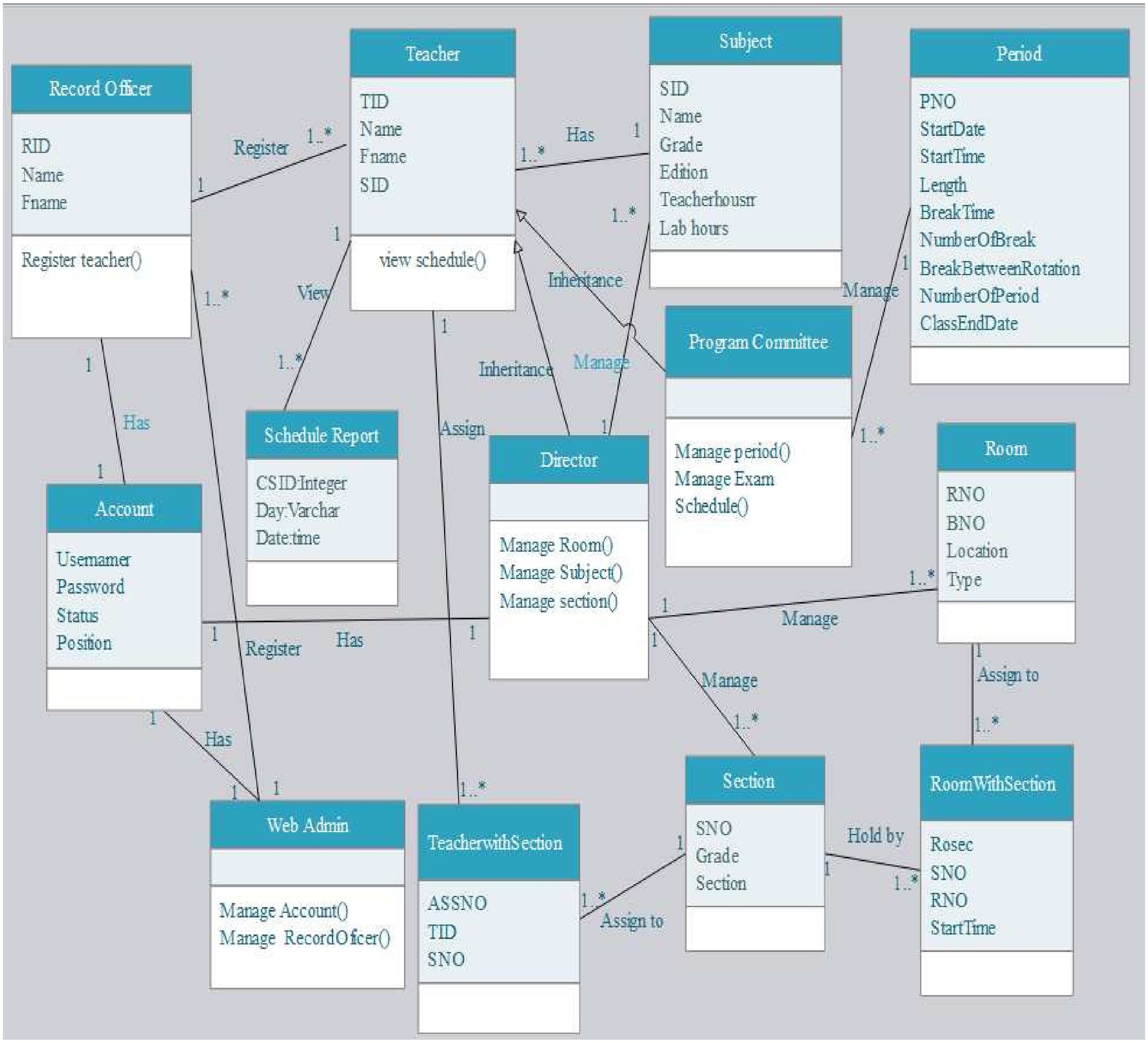


Figure 11 Analysis UML class diagram

Chapter three

3. System design

3.1. Introduction

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

The purpose of designing is to show the direction how the system is built and to obtain clear and enough information needed to drive the actual implementation of the system. It is based on understanding of the model the software built on. 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 change to the system after it has been put in to operation depends on the quality of the system design.

3.2. Design Class diagram

The class diagram represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The classes diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram for this the team developed the following class diagram.[2]

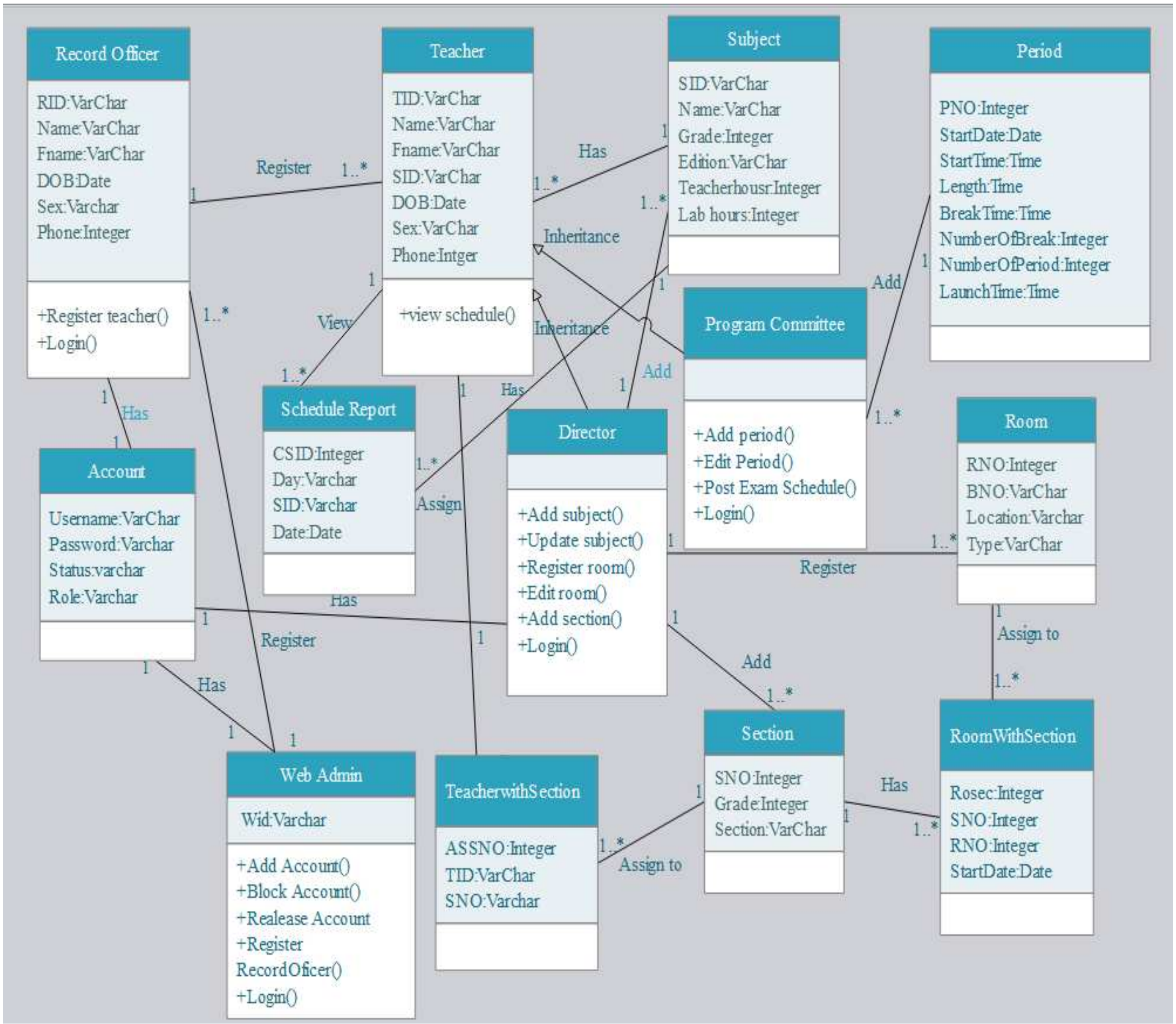


Figure 12 Design class diagram

3.3. Physical data model

Physical data model represents how the model built in the database. A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables[2].

Rules for translation:

- ✓ Entities become tables in the physical database.

- ✓ Attributes become columns in the physical database. Choose an appropriate data type for each of the columns.
- ✓ Unique identifiers become columns that are not allowed to have NULL values.
- ✓ Relationships are modeled as foreign keys.

Table 11 Subject table

Column name	Data type	Primary key	Foreign key	Uniqueness
Subject code	Varchar(50)	Yes		Yes
Subject name	Varchar(50)			Yes
Edition	Varchar(50)			
Credit hour	Integer			
Lab Hour	Integer			

Table 12 Room table

Column name	Data type	Primary key	Foreign key	Uniqueness
Room id	Integer	Yes		Yes
Room Number	Integer			
Building Number	Varchar(50)			Yes
Location	Varchar(50)			
Room Type	Varchar(50)			

Table 13 Teacher table

Column name	Data type	Primary key	Foreign key	Uniqueness
Teacher ID	Varchar(50)	Yes		Yes
Name	Varchar(50)			
Father Name	Varchar(50)			
Date of Birth	Date			
Sex	Varchar(50)			
Phone Number	Integer			
Subject Code	Varchar(50)		Yes	

Table 14 Period table

Column name	Data type	Primary key	Foreign key	Uniqueness
Period ID	Integer	Yes		Yes
Start Date	Date			
Start Time	Time			
Length of Period	Time			
Break Time	Time			
Number of Break	Integer			
Break Between Rotation	Time			
Number of Break per a day	Integer			

Class End	Date			

Table 15 Section table

Column name	Data type	Primary key	Foreign key	Uniqueness
Section Number	Integer	Yes		Yes
Grade	Integer			
Section	Varchar(50)			

Table 16 Teacher _Section table

Column name	Data type	Primary key	Foreign key	Uniqueness
AssNo	Integer	Yes		Yes
Teacher Id	Varchar(50)		Yes	
Section Number	Integer		Yes	

Table 17 Room _Section table

Column name	Data type	Primary key	Foreign key	Uniqueness
RoSec	Integer	Yes		Yes
Room Number	Varchar(50)		Yes	
Section Number	Integer		Yes	

3.4. Deployment diagram

Deployment modeling is used to show the hardware of the system, the software that is installed in the hardware and also shows how the software and the hardware components work together.

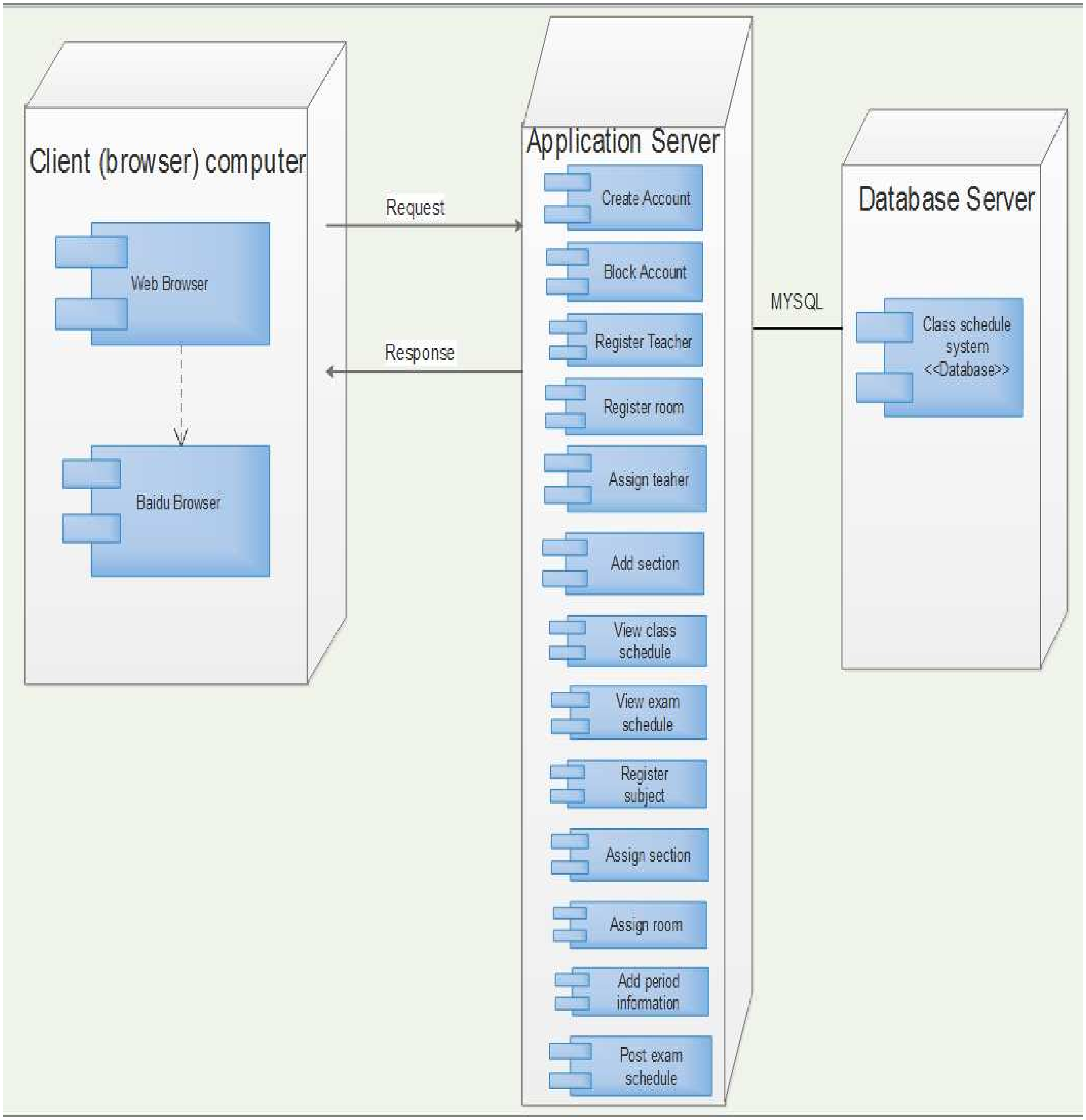
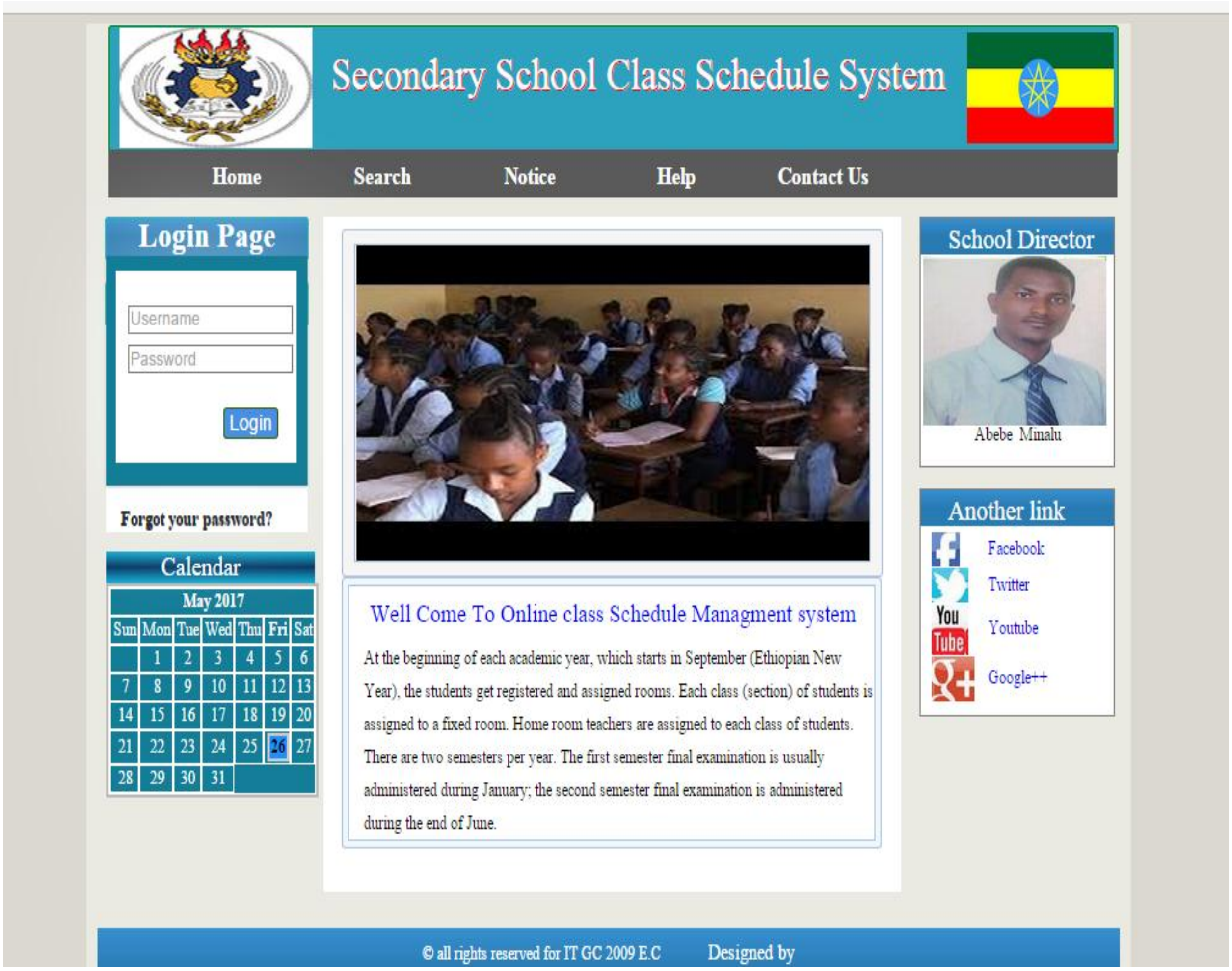


Figure 13 Deployment diagram

3.5. Graphical user interface



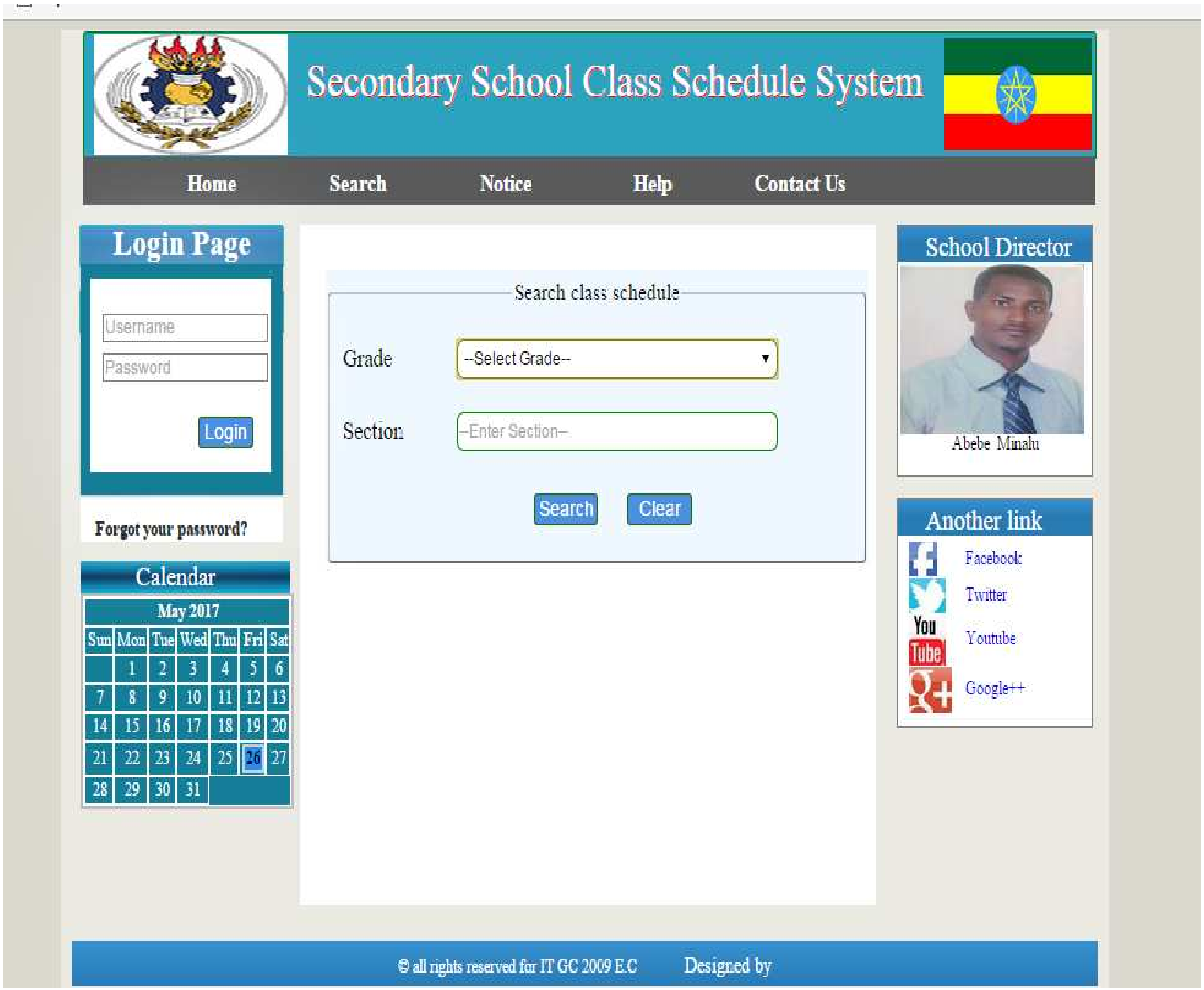
May 2017						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Well Come To Online class Schedule Managment system

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.

© all rights reserved for IT GC 2009 E.C Designed by

Figure 14 Home graphical user interface



May 2017

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Figure 15 Search class schedule graphical user interface

Chapter four

4. Implementation and testing

4.4. Implementation

Implementation in the system includes implementing the attributes and methods of each object and integrating all the objects in the system, to function as a single system the implementation activity spans the gap between the detailed objects designed model and a complete set of source code file that can be compiled together.

4.5. Algorithm design

4.5.1. Pseudo code

Pseudo code is compact and informal high-level description of a computer programming algorithm that uses the structural conventions of a programming language but is intended for human reading rather than machine reading. Pseudo code typically omits details that are not essential for human understanding of the algorithm, such as variable declaration, system-specific code and subroutine. The programming language is augmented with natural language descriptions of the details, where convenient, or with compact mathematical notation. The purpose of using pseudo code is that it is easier for humans to understand than conventional programming language code, and that it is a compact and environment-independent description of the key principles of an algorithm.

4.5.1.1. Pseudo code for conversion Gregorian calendar to Ethiopian colander

```
If Gregorian month is between January and August
        Ethiopian year= Gregorian year -8
Else
  Ethiopian year = Gregorian year -7
```

```
If (Gregorian month is January)
    If (remainder of Ethiopian year divided by 4 is 0)
        If (Gregorian date is less than 10)
            Ethiopian month is • ታህሳስ (December)
            Ethiopian date = (Gregorian date + 31) – 10
        Else
            Ethiopian month is ጥር (January)
            Ethiopian date = Gregorian date – 9
    Else
        If (Gregorian date is less than 9)
            Ethiopian month is • ታህሳስ (December)
            Ethiopian date = (Gregorian date + 31) – 9
        Else
            Ethiopian month is ጥር (January)
            Ethiopian date = Gregorian date – 8
Else if (Gregorian month is February)
    If (remainder of Ethiopian year divided by 4 is 0)
        If (Gregorian date is less than 10)
            Ethiopian month is ጥር (January)
            Ethiopian date = (Gregorian date + 31) – 9
        Else
            Ethiopian month is የካቲት (February)
            Ethiopian date = Gregorian date – 8
        Else
            If (Gregorian date is less than 8)
                Ethiopian month is ጥር (January)
                Ethiopian date = (Gregorian date + 31) – 8
            Else
                Ethiopian month is የካቲት (February)
                Ethiopian date = Gregorian date – 7
Else if (Gregorian month is March)
    If (Gregorian date is less than 10)
        Ethiopian month is የካቲት (February)
        Ethiopian date = (Gregorian date + 29) – 8
    Else
        Ethiopian month is መጋቢት (March)
        Ethiopian date = Gregorian date – 9
Else if (Gregorian month is April)
    If (Gregorian date is less than 9)
        Ethiopian month is መጋቢት (March)
        Ethiopian date = (Gregorian date + 30) – 8
    Else
        Ethiopian month is ሚያዝያ (April)
        Ethiopian date = Gregorian date – 8
Else if (Gregorian month is May)
    If (Gregorian date is less than 9)
```

```
        Ethiopian month is ሚያዝያ (April)
        Ethiopian date = (Gregorian date + 30) – 8
    Else
        Ethiopian month is ግንቦት (May)
        Ethiopian date = Gregorian date – 8
Else if (Gregorian month is June)
    If (Gregorian date is less than 8)
        Ethiopian month is ግንቦት (May)
        Ethiopian date = (Gregorian date + 31) – 8
    Else
        Ethiopian month is ሰኔ (June)
        Ethiopian date = Gregorian date – 7
Else if (Gregorian month is July)
    If (Gregorian date is less than 8)
        Ethiopian month is ሰኔ (June)
        Ethiopian date = (Gregorian date + 30) – 7
    Else
        Ethiopian month is ሀምሌ? (July)
        Ethiopian date = Gregorian date – 7
Else if (Gregorian month is August)
    If (Gregorian date is less than 7)
        Ethiopian month is ሀምሌ? (July)
        Ethiopian date = (Gregorian date + 31) – 7
    Else
        Ethiopian month is ነሀሴ(August)
        Ethiopian date = Gregorian date – 6
Else if (Gregorian month is September)
    If (remainder of Ethiopian year divided by 4 is 0)
        If (remainder of Gregorian year divided by 4 is 0)
            If (Gregorian date is less than 6)
                Ethiopian month is ነሀሴ (August)
                Ethiopian date = (Gregorian date + 31) – 6
            Else
                If (Gregorian date is less than 12)
                    Ethiopian month is ጳጉሜ
                    Ethiopian date = Gregorian date – 5
                Else
                    Ethiopian month is መስከረም (September)
                    Ethiopian date = Gregorian date – 10
        Else
            If (Gregorian date is less than 6)
                Ethiopian month is ነሀሴ(August)
                Ethiopian date = (Gregorian date + 31) – 7
            Else
                If (Gregorian date is less than 12)
                    Ethiopian month is ጳጉሜ
```

```
                       Ethiopian date = Gregorian date – 6
                   Else
                      Ethiopian month is መስከረም (September)
                      Ethiopian date = Gregorian date – 10
Else
     If (remainder of Gregorian year divided by 4 is 0)
         If (Gregorian date is less than 5)
                   Ethiopian month is ነሀሴ (August)
                   Ethiopian date = (Gregorian date + 31) – 5
        Else
            If (Gregorian date is less than 11)
                    Ethiopian month is ጳጉሜ
                    Ethiopian date = Gregorian date – 4
            Else
                  Ethiopian month is መስከረም
                 Ethiopian date = Gregorian date – 10
    Else
            If (Gregorian date is less than 5)
                  Ethiopian month is ነሀሴ (August)
                  Ethiopian date = (Gregorian date + 31) – 6
        Else
            If (Gregorian date is less than 11
                      Ethiopian month is ጳጉሜ
            Ethiopian date = Gregorian date – 5
    Else
        Ethiopian month is መስከረም (September)
        Ethiopian date = Gregorian date – 10
Else if (Gregorian month is October)
    If (remainder of Ethiopian year divided by 4 is 0)
        If (Gregorian date is less than 12)
            Ethiopian month is መስከረም (September)

            Ethiopian date = (Gregorian date + 30) – 11
        Else
            Ethiopian month is ጥቅምት (October)
            Ethiopian date = Gregorian date – 11
    Else
        If (Gregorian date is less than 11)
            Ethiopian month is መስከረም (September)
            Ethiopian date = (Gregorian date + 30) – 10
        Else
            Ethiopian month is ጥቅምት
            Ethiopian date = Gregorian date – 10
Else if (Gregorian month is November)
    If (remainder of Ethiopian year divided by 4 is 0)
        If (Gregorian date is less than 11)
```

```
              Ethiopian month is ጥቅምት (October)
              Ethiopian date = (Gregorian date + 31) – 11
          Else
              Ethiopian month is ህዳር (November)
              Ethiopian date = Gregorian date – 10
      Else
         If (Gregorian date is less than 10)
             Ethiopian month is ጥቅምት (October)
             Ethiopian date = (Gregorian date + 31) – 10
        Else
            Ethiopian month is ህዳር (November)
            Ethiopian date = Gregorian date – 9
Else if (Gregorian month is December)
      If (remainder of Ethiopian year divided by 4 is 0)
          If (Gregorian date is less than 11)
            Ethiopian month is ህዳር (November)
            Ethiopian date = (Gregorian date + 30) – 10
         Else
            Ethiopian month is • ታህሳስ (December)
            Ethiopian date = Gregorian date – 10
     Else
         If (Gregorian date is less than 10)
              Ethiopian month is ህዳር (November)
              Ethiopian date = (Gregorian date + 30) – 9
        Else
              Ethiopian month is • ታህሳስ (December)
          Ethiopian date = Gregorian date – 9[3]
```

4.5.2. Pseud code for login

```
 Fill the Login Form
Click the Login button
If (Form is filled)
    If (valid)
        Generate SQL select queries
        Connect to database
        Pass queries to database
If (any query fails)
        Display error message
        Else
Read session
If session exists on database, user is already logged in,
Display the page
Else
If they're correct
Create session ID
Store session ID on database
                Display the page
            End if
        End if
    Else
     Display error message
     Ask the user to refill the form
  End if
End if
```

4.5.3. Pseudo code assign teacher

```
User involved: department head
Fill login form
Click login button
 If check database present
     Fill assign form
     Click assign button
      If section is registered
          If check database previously assigned
               If check other teacher assigned
                    Display assigned to other teacher
               Else
                    Display successfully assigned
          Else
               Display previously assigned
     Else
     Display section is not registered
Else
Display error message
End If
```

4.6. Sample testing code

Sample code for assign teacher

```
<? Php
session start ();
include ("connection. Php");
?>
<html>
<head>
<title>
Teacher Assign
</title>
<link rel="stylesheet" type="text/css" href="setting.css">
</head><body>
<table><tr><td>
<?php require("header.php"); ?></td></tr>
<tr><td colspan="3">
<?php require("menu4.php");?>
</td></tr> <tr><td>
 <div id="sideleftdhead">
  <div id="sideleftdirector1">
```

```
            Aditional Links</div><?php
        if($_SESSION['uname']!=null&&$_SESSION['upass']!=null)
        {include("headmanagment.php");}
        echo "</div>"; echo "<br><br><div id='sideleftheaddate'>";
        echo "<div id='sideleftdirectordate1'>Calendar</div>";
        if($_SESSION['uname']!=null&&$_SESSION['upass']!=null)
        {include("date.php");}else
        {
            header("location:login.php");
        }
        ?></div>
</td>
        <td><div id="contenthead">


        <br><center><form action="" method="post">
        <table width="550px"><tr><td colspan="2" height="70px" style="font-size:21px;">
        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;Teacher Assign  Registeration Form</td></tr>
            <?php
        if($_SESSION['uname']!=null&&$_SESSION['upass']!=null)
        {echo "<form action='' method='post' >";  ?>
         <tr><td height="50px" style="font-size:21px;"> Teacher ID</td><td>
            <?php
```

```
echo "<select name='tid' style='font-size:12pt;color:black;border:1px solid gray;padding:2px;
width:175px;width:230px;'><option>Select teacher Id</option>";
    $td=$_SESSION['uname'];
$sdd=mysql_query("SELECT * from account where username='$td'") or die(mysql_error());
            while($trow=mysql_fetch_array($sdd))
            {$uid=$trow['uid'];
      $te=mysql_query("SELECT * from teacher where tid='$uid'") or die(mysql_error());
                  while($terow=mysql_fetch_array($te))
                  {$si=$terow['sid'];
      $sub=mysql_query("SELECT * from subject where sid='$si'") or die(mysql_error());
                        while($subrow=mysql_fetch_array($sub))
                        { $s=$subrow['sid'];
$teach=mysql_query("SELECT * from teacher where sid='$s'") or die(mysql_error());
                              while($tecrow=mysql_fetch_array($teach))
                              {$tt=$tecrow['tid'];
echo "<option value=$tt>$tt</option>";
                              }      }}}
   ?>
   </select></td></tr>
       <tr><td height="50px" style="font-size:21px;" >
       Grade </td><td><select name="grade" style='font-size:12pt;color:black;border:1px
solid gray;padding:2px; width:175px;width:230px;'><option>--Choose grade--</option>
            <?php
```

```
                    $arra=array("9","10");
                    foreach($arra as $row)
                    {
?>
</script><?php
                    echo "<option value=$row>$row</option>";
                    }
    ?>
    </select>
            </td></tr>
            <tr><td height="50px" style="font-size:21px;">
            Section</td><td ><input type="text" name="section" patern='[A-Za-z]+'
placeholder="Enter section" style='font-size:12pt;color:black;border:1px solid gray;padding:2px;
width:175px;width:230px;'>
</td></tr>
                    </tr><td height="85px"></td><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input
type='submit' name='register' value='Assign' style="font-size:12pt;color:black;border:1px solid
#336600;padding:2px;">&nbsp;&nbsp;&nbsp;&nbsp;
                    <input type='reset'  value='Clear' style="font-size:12pt;color:black;border:1px
solid #336600;padding:2px;"></td></tr>
                    </form></table></form></center>
                    <?php
  if(isset($_POST['register']))
  {
```

```
$ti=$_POST['tid'];

                    $dse=$_POST['section'];

                    $graf=$_POST['grade'];

                    $sdw=mysql_query("SELECT * from section where grade='$graf' &&
section='$dse'") or die(mysql_error());

                    $bun=mysql_num_rows($sdw);

                    if($bun!='0')

                    {

                    $sq=mysql_query("SELECT * from teacher where tid='$ti'") or
die(mysql_error());

     while($row=mysql_fetch_array($sq))

           {

                     $sid=$row['sid'];

                     $se="SELECT * from subject where sid='$sid'";

                     $see=mysql_query($se) or die(mysql_error());

                     while($row1=mysql_fetch_array($see))

                     { $credit=$row1['credit']; $lab=$row1['lab'];

                               $sum=$credit+$lab; $sq2=mysql_query("SELECT * from teacherscetion
where tid='$ti'") or die(mysql_error());

 $rows = mysql_num_rows($sq2); $total=$sum*$rows;

                                if($total<=30)

                               {

          $td=$_POST['tid']; $gra=$_POST['grade'];  $se=$_POST['section']; $d=0;
```

```
                                        $assign=0;
 $sq3=mysql_query("SELECT * from teacherscetion where tid='$ti'") or die(mysql_error());
                                        while($ro=mysql_fetch_array($sq3))
                                        {           if($ro['grade']==$gra && $ro['section']==$se)
                                         {   $d=1;
                                         }  }  if($d=='0')  {
$sq4=mysql_query("SELECT * from teacher where sid='$sid' && tid!='$td'") or
die(mysql_error());
                    while($row3=mysql_fetch_array($sq4))
                                                  {$to=$row3['tid']; $sq5=mysql_query("SELECT * from
teacherscetion where tid='$to' && grade='$gra' && section='$se'") or die(mysql_error());
$num=mysql_num_rows($sq5);
 if($num!='0'){  $assign=1; }  }
 if($assign=='0')
                                                  {
 $dd=mysql_query("Update schedule set value='0',value1='0' where sid='1'") or
die(mysql_error());
$sql="INSERT INTO teacherscetion (tid,grade,section)
VALUES('$_POST[tid]','$_POST[grade]','$_POST[section]')";
mysql_query($sql) or die(mysql_error());
if(!$sql) { echo '<script type="text/javascript">alert("Data not inserted!!!");</script>';}
else{echo '<script type="text/javascript">alert("successfully register!!!");</script>';}}
else { echo '<script type="text/javascript">alert("This section assigned to another person so that
not assign!!!");</script>';   } }
```

```
else {  echo '<script type="text/javascript">alert("This section is assigned before not
assigned!!!");</script>';  } }

 else  { echo '<script type="text/javascript">alert("He or she can not hold more than 30 credit
hour per week!!!");</script>';

}} }}

else {  echo '<script type="text/javascript">alert("Section not register!!!");</script>'}

   }

  ?> <?php}else {header("location:login.php")}?><br> </div></td>

            <td><div id="siderightahead">

             <div id="siderightdir"><center>


             </center>

 </div>

 <div id="siderightdir1">

             <?php

             $name=$_SESSION['uname'];

                        $sq1=mysql_query("SELECT * from account where username='$name'") or
die(mysql_error());

                        $row1=mysql_fetch_assoc($sq1);

                        $uid=$row1['uid'];

             $s=mysql_query("SELECT * from storeimage where uid='$uid'") or die(mysql_error());

             $ros=mysql_fetch_assoc($s);

             $filename=$ros['photo'];
             $path=$filename;

             echo "<img src='$path' width='206'   height='154' />" ;
             $sd=mysql_query("SELECT * from teacher where tid='$uid'") or die(mysql_error());
             $ro=mysql_fetch_assoc($sd);
             $name=$ro['name'];
             $fname=$ro['fname'];
             echo "<center>$name&nbsp;&nbsp;$fname</center>";
             ?>
             </div>
            </div>
            </td>
            <td>
            <div id="siderightindexadationalhead">
            <div id="siderightindexadationaldirector1">
            Another link </div>
            <div id="siderightindexadationaldirector2">
            <table>
             <tr><td><div id="facebook"></div></td><td>
            <p><a href="https://www.facebook.com/" style="text-decoration: none;">&nbsp;&nbsp;&nbsp;Facebook</a><p></td></tr>
            <tr><td><div id="twitter"></div></td><td><p><a href="https://www.twitter.com/" style="text-decoration: none;">&nbsp;&nbsp;&nbsp;Twitter</a></p></td></tr>
```

```
            <tr><td><div id="you"></div></td><td><p><a href="https://www.youtube.com/"
style="text-decoration: none;">&nbsp;&nbsp;&nbsp;Youtube</a></p></td></tr>

            <tr><td><div id="googleplus"></div></td><td><p><a href="https://plus.google.com/"
style="text-decoration: none;">&nbsp;&nbsp;&nbsp;Google++</a></p></td></tr></table>


            </div></div></td></tr><tr><td>

<?php include("footer.php");?>

</td></tr>

</div>

</body>

</html>
```

Chapter five

5. Testing

5.1. Unit testing

Unit testing is a software verification and validation method in which a programmer tests if individual units of source code are fit for use. The automated web based class schedule management system units are tested one by one.

5.2. Integration testing

Combining modules and testing them is called integration testing. Integration testing is gradual. First we test the highest level, or coordinating module, and only one of its subordinate modules.

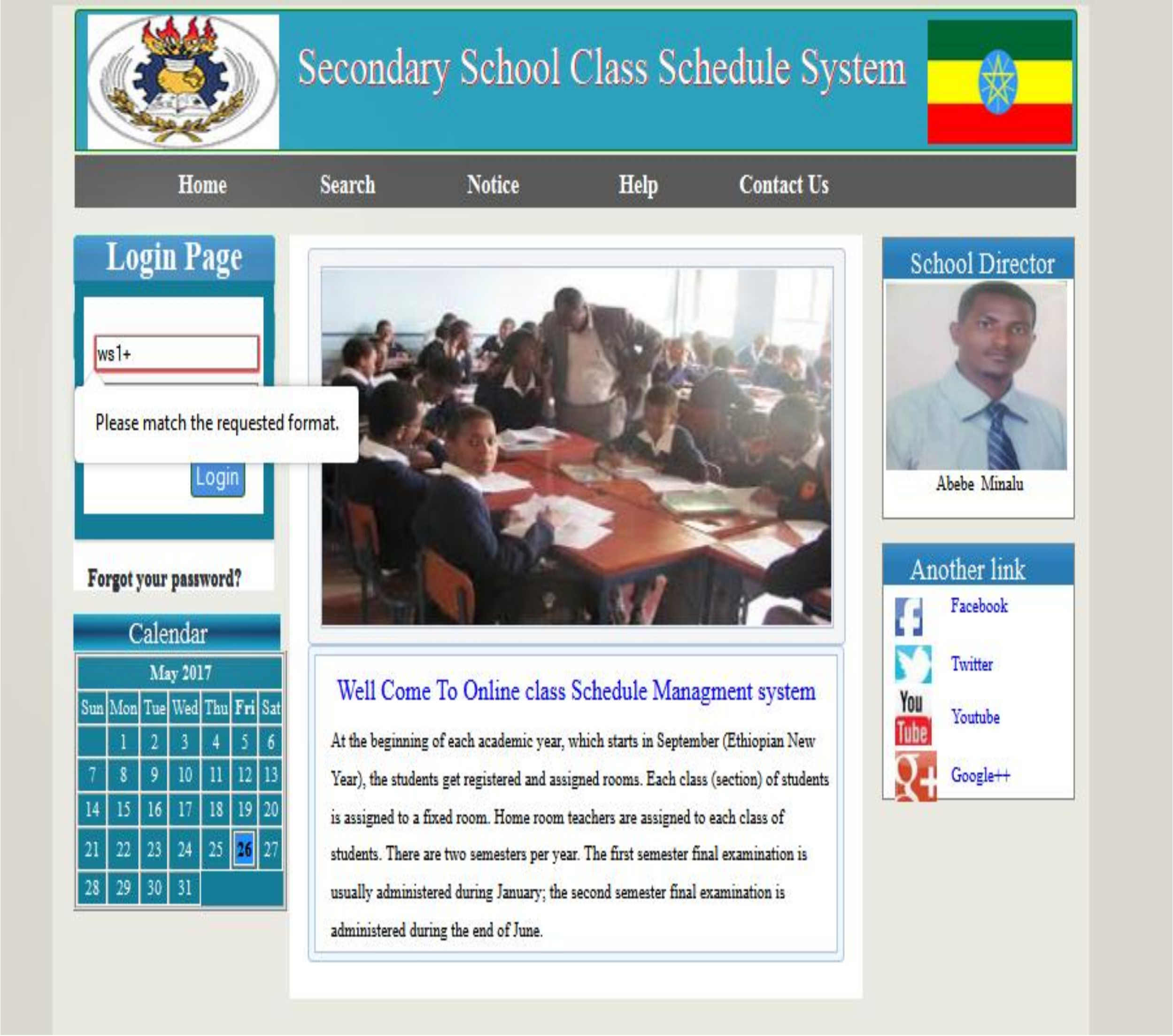
After unit testing, the automated web based class schedule management system is also tested whether every unit is integrated to each other.

5.3. System testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. It is a testing process in which the aim is to ensure that the overall system works as defined by the requirements.

- Is a testing process in which the aim is to ensure that the overall system works as defined by the requirements?
- Web based class schedule management system is functionally tested based on the use case model developed during the analysis phase.

5.3.1. Sample test



ws1+

Please match the requested format.

May 2017

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Well Come To Online class Schedule Managment system

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.

Abebe Minalu

Figure 16 Sample test for login

Chapter sex

6. Conclusion and recommendation

6.1. Conclusion

This class schedule management system is comprehensive web-based class schedule system management system software. It is designed for better interaction between students, teachers, record officer, director, department head and system administrator. This management software is very gracefully handles all the requirements for easy class schedule management system. The software being web based can be accessed from anywhere in the world which enables the students , teachers, department head, record officer, director and administrator the management be in touch with each other at all times.

6.2. Recommendations

The system we have developed is a automate class schedule management system it needs a skilled person to work with the system. So, we recommend the system should be required the responsible and skilled person. We highly recommend the system should be kept in highly safe and favorable condition.

6.3. Future enhancement

As a new system developer,we recommend the following functionalities are more done such as:-

- ✓ Develop better security mechanism.
- ✓ Integrate with student placement management system.
- ✓ Include Ethiopian holiday date.
- ✓ Make class schedule works any period size.
- ✓ Develop to mobile based application

References:

1. Web References

1. [Online] http://www.ops.fhwa.dot.gov/functional-requirement.

2. [Online] http://en.wikipedia.org/wiki/object-oriented-anyalysis and design.

3. [Online] http://etd.aau.edu.et /123456789/9283/1/Zenebe%20Nigussie.pdf

2. Bibliography

3. **Hoffer.** *Modern System Analysis and Design by Jeffery A. Hoover Fred R.* 2000.

4. **W.Ambler.** *The application developer's guide to object oriention and the UML.* Second edition. united States : 2001.

5. **W.Erben, E.Burke and.** *Practice and Theory of Automated Timetabling, Third International Conference.* 2000