

ONLINE STUDENT UNION VOTING SYSTEM IN DEBRE MARKOS UNIVERSITY

A PROJECT PROGRESSIVE REPORT

Submitted by:

NAME-----ID NUMBER

Melsew Dagnaw-----TER/4745/07

Gizachew Erkyihun -----TER/4733/07

Walie Sinishaw -----TER/4757/07

Ensmaw Kihale -----TER/4726/07

In partial fulfillment for award of degree of

BACELOR OF SCIENCE IN INFORMATION TECHNOLOGY

Advisor: Haylemaryam A

ADVISOR SIGNATURE -----



DEPARTMENT OF INFORMATION TECHNOLOGY

INSTITUTE OF TECHNOLOGY

DEBRE MARKOS UNIVERSITY

Debre Markos

DEBRE MARKOS, Ethiopia

November 15/5/2010

Abstract

This documentation explains about online student's union voting system which we would develop for Debre Markos University. Currently the services that are given instudent union voting system is handled manually and the number of students who is getting service that are increasing every year. So to give the services and manage become very difficult as student union voting system handled its every day activity manually. Having the above problem, we are going to solve this by making online student voting System. This voting system claims to change the manual voting system in to Online. The system is very interactive and useful for Debre Markos university students since it is applicable anywhere in the same functionality. Online voting system used to register, vote, register voter and count the total number of voter to vote candidate. It is advantageous in reducing the complexity of the election process. To do this project the project team used deferent data collection method such as interview and document analysis.in order analyzed and design the system we are going to use object oriented approach for both analyzing and design the new system. To change the current manual system to a web based, we need deferent software and hardware tool such as PHP.

Acknowledgement

First of all we would to thank GOD for making us healthy. Without the will of GOD everything is impossible. Next we will extend our thanks and appreciation to all individuals that give important information to us. We have a great thanks and appreciation to our advisor **Haylemaryama** for his excellent advice and supervision. He has a good advice and supervision ability and smart approach to us. Next we thank and appreciate our classmates and friends for their wellness to give important information and ideas for us. Without the participation of individuals our team project has no ability to reach to this stage. Moreover, we would like to thank Debre Markos student union president for their good approach at interview time and wellness at giving important information for us.

Table of content

Contents

Abstract.....	i
Acknowledgement	ii
Table of content.....	iii
List of table.....	vi
List of figure	vii
ABBREVIATIONS AND ACRONYMS	viii
CHAPTER ONE	1
1 Introduction	1
1.1 Background of the project	1
1.2 Problem statement.....	2
1.3 Objective of the project.....	3
1.3.1 General objective of the project	3
1.3.2 Specific objective.....	3
1.4 Scope of the Project	3
1.5 Limitation of the project	4
1.6 Significant of the project.....	4
1.7 System requirement	5
1.7.1 Hardware requirement.....	5
1.7.2 Software requirement	5
1.7.3 Programming language	5
1.8 Data collection Methodology	6
1.8.1 Design Methodology.....	6
1.8.2 Data gathering technique	6
1.9 Feasibility study.....	6
1.9.1 Technical feasibility	7
1.9.2 Operational feasibility.....	7
1.9.3 Economic feasibility	7
1.9.4 Legal feasibility.....	8
CHAPTER TWO	9

2	System analysis.....	9
2.1	Overview of the existing system.....	9
2.2	Users of the existing system.....	9
2.3	Overview of Proposed system.....	10
2.4	System requirement specification	10
2.4.1	Functional Requirements	10
2.4.2	Non Functional Requirements	11
2.4.3	Business Rules	11
2.5	System analysis requirement.....	12
2.5.1	Actor identification	12
2.5.2	Use case identification	12
2.5.3	Use case diagram	13
2.5.4	Use case description	15
2.5.5	Sequence diagram.....	20
2.5.6	Analysis of class diagram	23
2.5.7	Activity diagram invalid.....	25
	CHAPTER THREE	28
3	System design	28
3.1	Design class diagram	28
3.2	Class diagram description	30
3.3	Database design\Physical data model	32
3.4	User interface design.....	36
3.5	System architecture	39
3.5.1	Deployment diagram.....	39
	CHAPTER FOUR.....	40
4	Implementation	40
4.1	Overview of programing language	40
4.2	Algorithm used.....	40
4.3	Sample code	41
	CHAPTER FIVE	46
5	Testing.....	46
5.1	Sample test	46

5.1.1	Sample test for login page	46
5.1.2	Sample test for double voting.....	47
6	CHAPTER SEX.....	48
6.1	Conclusion and recommendation	48
6.1.1	Conclusion.....	48
6.1.2	Recommendation and Future enhancement.....	48
	References.....	49

List of table

Table 1: actor identification table	12
Table 2: use case identification table	12
Table 3: login use case description	15
Table 4: Use case description for view result	16
Table 5: use case description of create account	17
Table 6: use case description of view candidate	18
Table 7: use case description of take exam.....	19
Table 8: class description of student service directorate.....	30
Table9: method for student service directorate	30
Table 10: class description of candidate	31
Table 11: method description of candidate	31
Table 12: class description of voter	31
Table 13: method description of voter	32
Table 14: class description of main registrar	32
Table 15: method description of main registrar	32
Table 16: student service director table	33
Table 17: candidate registration table	33
Table 18: promotion table	34
Table 19: voter registration table	34
Table 20: voter promotion table.....	35
Table 21: request table	35
Table 22: account table	36

List of figure

Figure 1: use case diagram.....	14
Figure 2: login sequence diagram	20
Figure 3: Sequence Diagram for take exam	21
Figure 4: sequence diagram of view exam result.....	22
Figure 5: analysis class diagram	24
Figure 6: Activity Diagram of login	25
Figure 7: Activity diagram of vote.....	26
Figure 8: Activity diagram of create account.....	27
Figure 9: design class diagram.....	29
Figure 10: login page	37
Figure 11: admin user interface	38
Figure 12: deployment diagram	39
Figure 13: sample test for login page.....	46
Figure 14: sample test for double voting	47

ABBREVIATIONS AND ACRONYMS

- ✧ CPU:ComputerProcessing Unite
- ✧ GB:Gigabyte
- ✧ RAM: Random Access Memory
- ✧ GHZ:Giga Hertz
- ✧ UML:Unified Modeling Language
- ✧ WAMP:Windows Apache MySQLPHP
- ✧ PHP: Hyper Preprocessor Scripting Language
- ✧ CSS:CascadingStyle Sheet
- ✧ HTML:Hyper Text Markup Language
- ✧ CGPA:CumulativeGrade Point Average
- ✧ BR:Business Rule
- ✧ UC:Use Case
- ✧ SSD:StudentServicedirector
- ✧ EC: Election Committee
- ✧ OOP Objects Oriented Programming
- ✧ DBMS :Database Management Systems
- ✧ PW: Password
- ✧ UN:User Name
- ✧ EC:Ethiopian calendar

CHAPTER ONE

1 Introduction

Information technology plays a big role one's country society. With the introduction of computers, the business world was changed forever. Using computers and software, businesses use information technology to ensure that their departments run smoothly. Web application is one category of information technology in which applications and information are stored on servers and users can access that information using web browsers. Online voting system is a web application that allows voter can vote their representative easily in voting systems. In online student union voting system election can be conducted in free and fair manner in every two years in secret ballots. The president, the vice president and the general secretary come to power through direct election in a secret ballot every two year per their vote they secured. The person with the highest vote shall become the president, in addition to these chief executives, the union has executive committees. The aim of our project is to develop an interactive, electronic, voting system for Debre Markos student union members' election with which students can vote using a web browser. It automates the existing manual activities like voter registration, candidate & election process voting and vote counting. The project is expected to help students as well as the university at large in overcoming the existing voting problems such as time consuming voting process, extravagant resource oriented election, geographical limited voting, and undocumented and unstructured information capture. The system makes not only the voting process easy but also assist students by providing them with information associated with the student union. The system is capable of improving the user effort and time required and reduces the resource expenditure of the university.

1.1 Background of the project

Debre Markos University is one of Ethiopian educational University established in 1999 which is supposed for excellence in Technology. Debre Markos University is established by the Federal Democratic Republic of Ethiopia. The University is located two kilometers east from the central square of the town. It is found in Northwestern part of Ethiopia at Debre Markos town. The University admitted the first 760 students in February 1999E.C, and the student union portal and election system in the university was started its work in May, 1999 EC in the form of small manual based system. When the university earned additional students per each New Year, the

task and responsibilities of the student union becomes larger even if they follow still manual systems. Online student union voting system are developed by summer student in 2009 EC but in our project there additional features that no done by summer student. Some of additional features are, Restrict election date and time, Candidate takes exam, Set student apply date, Submit discipline record student.

1.2 Problem statement

Existing system has a number of problems in the working procedure for university and students. At present Debre Markos University manual system to deal with the voting system. The existing student union members voting system has the following problems:

- ☀ **Time consuming:** The process of collecting data and entering the collected data into the existing system consumes or take too much time to conduct. Example to collect candidate data from each department to require much time.
- ☀ **Too much paper work:** since the voting system is manual, the process involves much paper work and paper to storage, which is difficult as paper become bulky with the student size. Example to register candidate from each department to require much paper work and paper act as storage material.
- ☀ **Loss of registration:** The name of the candidate is registered on the paper, since it is manual and the paper can be lost and it may require re-registration.
- ☀ **Duplication of work:** There are repetitions of works in the existing system. This duplication of works leads to losing many resources. Example the name of the candidate is registered on the paper, since it is manual and the paper can be lost then the candidate register repeatedly so duplication of works can be occurred.
- ☀ **Difficult to keep the student's interest:** Because the system is manual, the candidate information is not fairly verified to the student. Example candidate information can verified to two student that select from each class from each department to elect or to be elected the president and vice president so two student cannot be satisfied all student interest.
- ☀ **Lack of security in the existing system:** Poor security system because one can get easily the document and change whatever they want, loss of information etc.

Because there is manual unauthorized person can update, delete candidate information.

1.3 Objective of the project

1.3.1 General objective of the project

The general objective of our project is to design and develop online voting system for Debre Markos University students union.

1.3.2 Specific objective

The specific objectives of the proposed system are:

- ✨ To Study and understand the existing voting process.
- ✨ To Design the proposed system including the system architecture.
- ✨ To Develop/Implement a prototype election/voting system.
- ✨ To Evaluate and recommend deployment strategies for the proposed system.
- ✨ To select the appropriate development tools for the system.
- ✨ To maintain and keeps all the information of all the candidates and voters.
- ✨ To identify and analysis the problem with current system and functional and nonfunctional requirement of new system.
- ✨ To develop more secured system to ensure voting fairness and a good interface to make it easy to use.
- ✨ To allow legally student can register to the system and participating in the voting process.
- ✨ To manage accounts for the user.

1.4 Scope of the Project

The scope of the project refers to the area, where the ongoing system is applied and how much it is powerful to solve the problem. It also shows that how the specified specific objective achieves by the project. The following are the scope of online student union voting system for Debre Markos University.

- ✨ The system shall allow creating new account for users.

- ✧ The system shall allow login to the system only authorized and authenticated users.
- ✧ The system shall validate and authenticate the users' username and password.
- ✧ The system shall allow register candidates, and voter.
- ✧ The candidates take online exam and see their exam result.
- ✧ Calculate total number of votes for each candidate.
- ✧ The system shall allow post final result.
- ✧ At the end of the election, the system allows generate report of the election.

1.5 Limitation of the project

Some of the limitation of this project is:-

- ✧ The system uses only English language.
- ✧ The system cannot develop Security camera when candidate take the exam.

1.6 Significant of the project

The significance of the project means the important role of the project to all the societies, the users and to the concerned bodies.

- ✧ We eliminate voting repetition, while the voting will be taking place.
- ✧ We eliminate the geographical limitation by making vote from anywhere in the University.
- ✧ We enable the administrator to update, delete and edit information about the voter and candidate simpler.
- ✧ Any voter who has a local area network connection can make his/her sound to be heard equally his/her vote to the preferred candidate.
- ✧ Resource and finance expenditure in terms of meeting oriented costs will be saved and the student union can use that for another essential works such as disable student that can be supported by the student union in terms of financial resources.
- ✧ Error of counting of votes will not occur, because the system assures that counting is done automatically so that there are no voting frauds at all.
- ✧ Increases accuracy and availability and quality of the voting process and number of voters as individual will find it easier and more convenient to vote.

- ✧ It provides distributed and equal information through online for all voters (students) and Information available at time and security for the data.

1.7 System requirement

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently.

1.7.1 Hardware requirement

The hardware requirements or just requirements are the requirements of a hardware device. Hardware requirement include:

- ✧ Computer and personal computer:-used to write documentation and implementation.
- ✧ Processor Intel(R) core(TM)i3-4160 CPU @ 3.60 GHz
- ✧ RAM – 3.24GB.
- ✧ Hard Drive - 120 GB or higher.

1.7.2 Software requirement

The software requirements specification is the single most important document in the software development process. It provides the basis for development as well as for validation. The following are software requirement:

- ✧ WAMP Server: manage servers setting. The reason of selecting WAMP server are secure means they are not easily attack by virus when we compare to other servers.
- ✧ Microsoft office Visio: used to draw different UML those are necessary to structure the system. E.g. Activity Diagram, Class Diagram, Sequence Diagram and Use case Diagram. Because it is easy to use and understand.
- ✧ Text editor: Notepad++, Notepad.
- ✧ Operating System - Windows 7.
- ✧ Browser: internet explorer or Firefox.

1.7.3 Programming language

Programming language is important to complete the system in the best way and to make it good, easy and interactive with the customer.

PHP: it used to learn how to make PHP interact with WAMP server on our system to deliver instant content.

- ✨ It is popular and widely used.
- ✨ Easier to fix problems.

Html: that used to create web content so it can be displayed by a browser.

CSS: for the formatting of the system. That is it defines the style of a website's content.

JavaScript: for animation purpose and to display prompt boxes.

1.8 Data collection Methodology

1.8.1 Design Methodology

To design the system, the project team has chosen Object Oriented Modeling techniques and unified modeling language tools. Reason for choosing:

- ✨ It enables us to comprehensively model a system before we develop it.
- ✨ Modification of the object implementation is easy because objects are loosely coupled.
- ✨ Understanding of the structure is easy because object oriented modeling and tools used to represent real world entities.

1.8.2 Data gathering technique

To design and develop online student union voting system for Debre Markos student, we used the following methods to gather information about the current system and alternate ways to develop the new system.

- ✨ **Interview:** To collect information from the president of the student council and executive members of the union.
- ✨ **Document analysis:** Study the document that is used in the past, in order to ensure whether the system had been figured out before and understand about the problems in the current system, rules and procedures for processing data.
- ✨ **Other student:** to collect information from student that can be voted in the past.

1.9 Feasibility study

The feasibility study aims to provide answers to a number of issues such as Is it easy to operate? Does the return on investment justify the project and running costs? Is the system easier for

maintenance? Is the system compatible with the legal, political and social requirements of society in general and that of Debre Markos University and the country in particular? Vision

1.9.1 Technical feasibility

Online student union voting system is technically feasible. In order to ensure whether the system is technically feasible or not, the system should specify the following cases:

- ✱ The software currently possess the necessary technology: Because it achieves the required goal, as much as possible we tried to encounter all hardware and software requirements and also the technology is easily available and deployed everywhere.
- ✱ The new system posses' necessarily technical experts: In this project the team uses languages such as HTML, PHP, java script and CSS to develop the new system. Example to control double voting using voting states when the voter vote on candidate then the vote states can be voted. All these are the technology side and once the module is developed it can be easily held by non-technical person, so it doesn't require any technical expertise.

1.9.2 Operational feasibility

This project is operationally feasible because the proposed system is a good solution maker of the problem or specific solution will work in the existing system and create a good environment towards the user of the system. So, the proposed system is operationally feasible because:

- ✱ We have all the resource needed for its implementation.
- ✱ The system is accepted and supported by the users and site viewers.
- ✱ The system will minimize the time and man power needed to give fast and hospitable service to the users.

1.9.3 Economic feasibility

The current system used by the student union election result in enormous expenditure on paper, pen, time and other costs due to improper mechanism to deal with the customer information. The project resolves these additional requirements and expenditures by using a computerized system. The new system is economically feasible. Economic feasibility can be seen in two ways:

1.9.3.1 Tangible benefits

The system has Tangible benefits such as:

- ✱ Cost Reduction
- ✱ Error Reduction
- ✱ Increase Speed of activity

1.9.3.2 Intangible benefits

The system has intangible benefits such as Reduce Resource Consumption Increase, Security, reliability and trust.

1.9.4 Legal feasibility

The proposed system's role is to automate the student union election processes of Debre Markos University, it do not interfere with any of Debre Markos University, rules and regulations. Hence, the system is legally feasible.

CHAPTER TWO

2 System analysis

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. [1]

2.1 Overview of the existing system

The existing system of the Debre Markos university student union voting system used manual election first the two students from each class of each department male and female are elected directly by getting high number of counts from their class then those students who are elected are joining the student union as members of parliament of the student union. Both students who are elected should have more than 3.0 CGPA. The first session of the parliament as its members came is to elect. The president of the student union, vice president and secretary of the student union mainly. There will be assigned the members as a leader of deferent clubs that are found in the student union. The election process for president, vice president and secretary will be candidate from the parliament members and they will have “elect me” session to the parliament members then the parliament members will choose the candidate with secret ballot. The one who gets the maximum number of counts will be a president of student union, the second maximum number of count will be vice president of student union and the third maximum number of count will be secretary of student union.

2.2 Users of the existing system

Users of the existing system are:-

- ☀ **Candidate:** who are the member of our university student and candidate for this election to represent student.
- ☀ **Voters/students:** are regular student that learns in the university currently and those who apply to vote his/her representatives in the election system.
- ☀ **Student service directorate:** one who register discipline student that are not participating in election act as a candidate.

2.3 Overview of Proposed system

In Online student union voting system admin can post their notice where CGPA greater than three can act as a candidate that can be elected then the candidate create there account and send request to admin then the admin can be approve the request. The candidate login within that account and post there promotion then the voter create there account and send request to admin then the admin can be approve the request then the voter login within their account and view the candidate promotion then the voter can vote candidate. Online student union voting system is expected to help University as well as the students at large in overcoming the existing voting problems such as time consuming voting process, extravagant resource oriented election, geographical limited voting, and undocumented and unstructured information capture. The system makes not only the voting process easy but also assist students by providing them with information associated with the student union. The system is capable of improving the user effort and time required and reduces the resource expenditure of the University.

2.4 System requirement specification

System requirements specification is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. [1]

2.4.1 Functional Requirements

The functional requirements of the system describe the necessary functions for which the system is expected to fulfill. [2]The requirements specified are helpful to clearly understand the scope and the objective of the system, and consequently this will be helpful for designing the system effectively. The proposed system meets the following functional requirements:-

- ✱ The system shall allow using users' friendly and dynamically interactive online election system.
- ✱ The system shall allow creating new account for users.
- ✱ The system shall allow login to the system only authorized and authenticated users.
- ✱ The system shall validate and authenticate the users' username and password.
- ✱ The system shall allow register candidates, and voter.
- ✱ The system shall provide proper features to post information.
- ✱ The candidates take online exam and see their exam result.
- ✱ Calculate total number of votes for each candidate.
- ✱ The system shall allow post final result.

- ✱ At the end of the election, the system allows generate report of the election.
- ✱ The system shall allow user can logout to the system.

2.4.2 Non Functional Requirements

Nonfunctional requirement is one of the system requirements which the proposed system should include security, availability, and performance etc. [2] the following are different non-functional requirements of the proposed or new system can perform. Those are:

- ✱ **Security:** The system authenticates users by using username and password and also the system is secured by using encryption type md5.
- ✱ **Performance:** Since the system is web based the deliver response time of the system should be very fast. It perform its activity that are relating to the vote is accurately.
- ✱ **Usability:-**this system allows all students to participate in election easily with any place in the university.
- ✱ **Availability:** The system should be available for access at restriction day and time at election set by SSD. And also the interaction between the voters and the system is more than enough to know about the union and the election process.
- ✱ **Error handling:** - This system allows preventing or eliminating of error by displaying the message box or the system warns the users who make errors.

2.4.2.1 Technical Requirement

- ✱ The interface of the system should be user friendly (easy to use).
- ✱ The interface should display error message if it detects invalid input.
- ✱ The system should deny unauthorized accesses to the system domain.

2.4.3 Business Rules

A business rule is effectively an operating principle or policy the software must satisfy. It often relevant to access control issues, business calculations, or operating policies and principles of the organization. Therefore, our new system has the following business rules.

BR1: The Candidate should be member of Debre Markos University achieved by candidate id.

BR2: The candidate and voter should have only regular student to participate in election.

BR4: one voter select only one candidate at election time president, vice president or secretary.

BR5: in Debre Markos university student union rule the Candidate should have CGPA greater than 3.0.

2.5 System analysis requirement

The analysis of the system is the basic thing in our system development which helps the system designer (we) to find the purpose of the system .In case of the new system we develop the purpose is in order to make providing computerized information handling system.[1]

2.5.1 Actor 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. [1]Actors of the proposed system are: -

Table 1: actor identification table

Actor	Description
Admin	This actor who control and manage the systems.
candidate	Is an actor who has the member of our university student andelected by voter.
Voter	Those who are regular students that learns in the university currentlyand those who apply to vote his/her representatives in the election time.
Main registrar	The act or who submits student data which are regular students to the system.
Student service director	The actor who submit student discipline record.

2.5.2 Use case identification

Use case: A use case describes a sequence of actions that provide a measurable value to an actor. [1]A use case is drawn as a horizontal ellipse on a UML use case diagram.

Table 2: use case identification table

ID	Use Case Name	Use/Include
UC1	Login	
UC2	Approve student account	UC1
UC3	Create account	UC1
UC4	Update account	UC1
UC5	block account	UC1

UC6	View account	UC1
UC7	Take backup	UC1
UC8	Restore backup	UC1
UC9	View feedback	UC1
UC10	Delete feedback	UC1
UC11	Post promotion	UC1
UC12	View election result	UC1
UC13	Take exam	UC1
UC14	vote	UC1
UC15	View exam result	
UC16	View candidate	
UC18	Add notice	UC1
UC19	View notice	
UC20	Submit student discipline record	UC1
UC21	Set student apply date	UC1
UC22	update student apply date	UC1
UC23	Set exam date	UC1
UC24	Prepare exam question	UC1
UC25	Submit student data	UC1
UC26	View promotion	
UC27	Send account request	UC1
UC28	Give feedback	UC1
UC29	View student data	UC1
UC30	View Report	
UC31	Logout	UC1

2.5.3 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 case. [1]

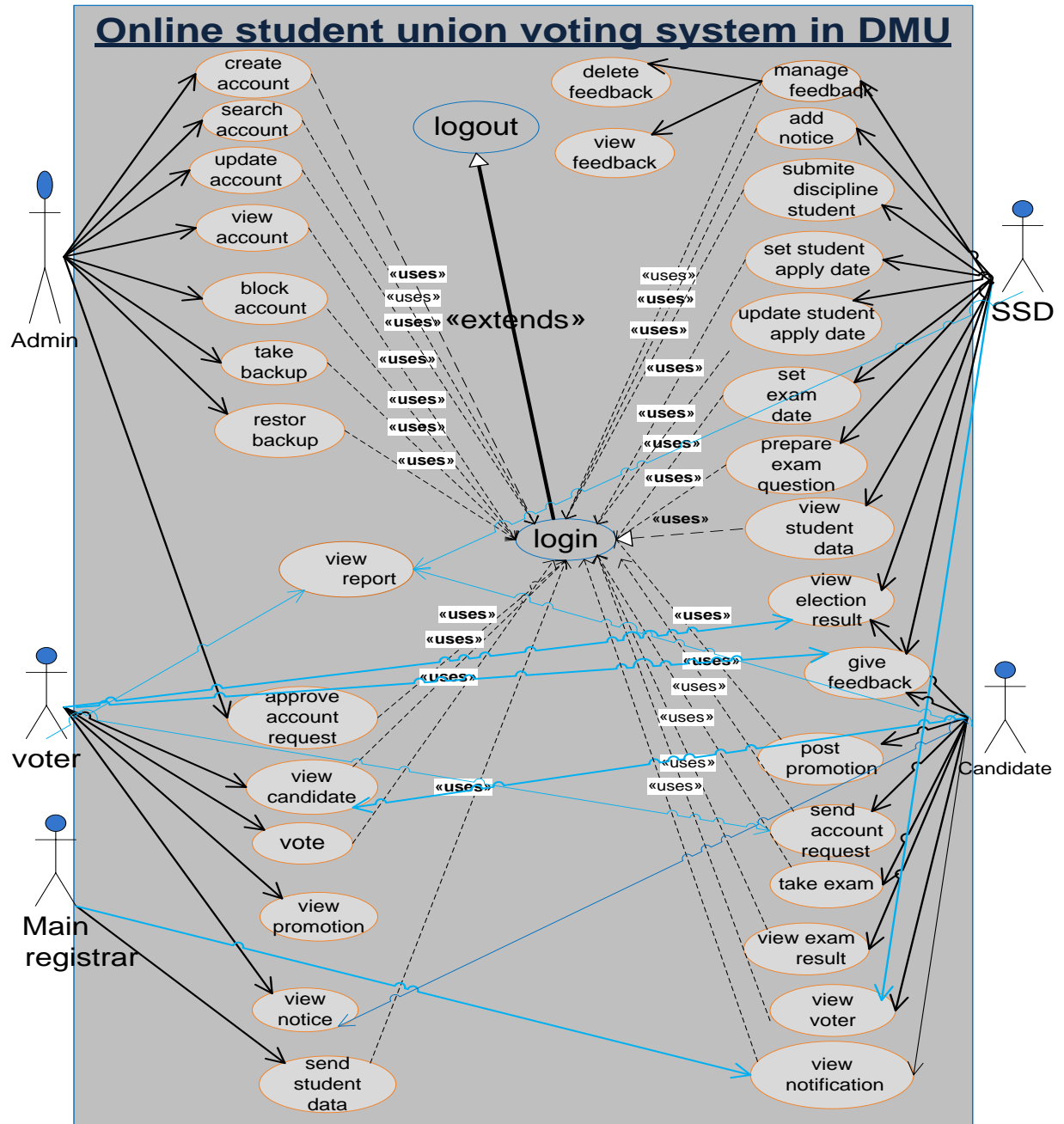


Figure 1: use case diagram

2.5.4 Use case description

The following consecutive tables show the use case description for each of the use cases. Each table contains the use case name, the actor which initiates and interacts with the use case, description of the use case and typical course of events that show the interaction between the actor and the use case which enable the team to easily depict the functions of the proposed system.

Table 3: login use case description

Use case ID	UC1
Name	Login
Actor	user
Description	This describes how the users log into the system.
Precondition	Users of the system should have user name and password.
Basic course of action	<ol style="list-style-type: none">1. Open web site.2. User enters username and password.3. Click login button.4. System verifies username and Password.5. If username and password is valid.6. User authenticated and gets access to the system.7. Use case ends.
Alternative course action	<ol style="list-style-type: none">1. User is not authenticated and is denied access to the system.2. System displays an incorrect username and password message.3. System enables user to try again 4 times and wait for a few second.4. Use case returns to step 2 to fill the correct username and Password.
Post condition	Login in to the system successfully.

Table 4: Use case description for view result

Use case ID	UC12
Name	View election result
Actor	Voter, candidate, student service director.
Description	This describes the process of how the user views the election results by using the system.
Precondition	Time must be run over the limit.
Basic course of action	<ol style="list-style-type: none">1. The user must be open website.2. The user can ask information they want to know.3. After searching necessary information click on view button.4. After getting necessary information they can view.5. Use case end.
Alternative course of action	If all users can't see the result of the election, try again and login to the system.
Post Condition	The user knows the wanted information.

Table 5: use case description of create account

Use case ID	UC3
Use case name	Create Account
Actor	Administrator
Description	the system administrator registers authorized users to the System.
Pre- condition:	System administrator must login and should manually get list of eligible users' information first.
Basic course of action (Flow of event):	<ol style="list-style-type: none">1. Clicks on register link.2. Registration form displayed3. Enter user's profile...With username & password.4. System administrator clicks on register button5. validate data entry6. Display successfully registered message.7. Use case ends
Alternative course of action	<ol style="list-style-type: none">1. If the data entry not valid, system displays invalid Input message.2. If user has been already registered, user registered message displayed.
Post- condition	The System Administrator created account so users can login the system.

Table 6: use case description of view candidate

Use case ID	UC16
Name	View candidate
Actor	Voter, student service director, candidate.
Description	This describes the process of how the user views the candidate information by using the system.
Precondition	Candidate must fulfill the required information and rules.
Basic course of action	<ol style="list-style-type: none">1. The user must be open website.2. The user can ask information they want to know.3. After searching necessary information click on candidates view button.4. After getting necessary information they can view.5. Use case end.
Alternative course of action	If all users can't see the candidates' information, try again and login to the system.
Post condition	The user knows the wanted information.

Table 7: use case description of take exam

Use case ID	UC13
Name	Take exam
Actor	Candidates
Description	This use case describes how the candidate to login into the System to takes exam.
Precondition	The user must have username and password.
Basic course of action	<ol style="list-style-type: none">1. Open web site.2. User enters username and password.3. Candidate enter exam password.4. Click login button.5. System verifies username and Password and exam password.6.If exam password, username and password is valid7. Candidate authenticated and takesExam on the system.8. Use case ends.
Alternative course of action	<ol style="list-style-type: none">1. User is not authenticated and is denied access to the system.2. System displays an incorrect username and password message.3. System enables user to try again.4. Use case returns to step 2 to fill the correct exam password, username And password.
Post condition	User gets access to the system according to their predefined system privileges.

2.5.5 Sequence diagram

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 or scenario. [3]

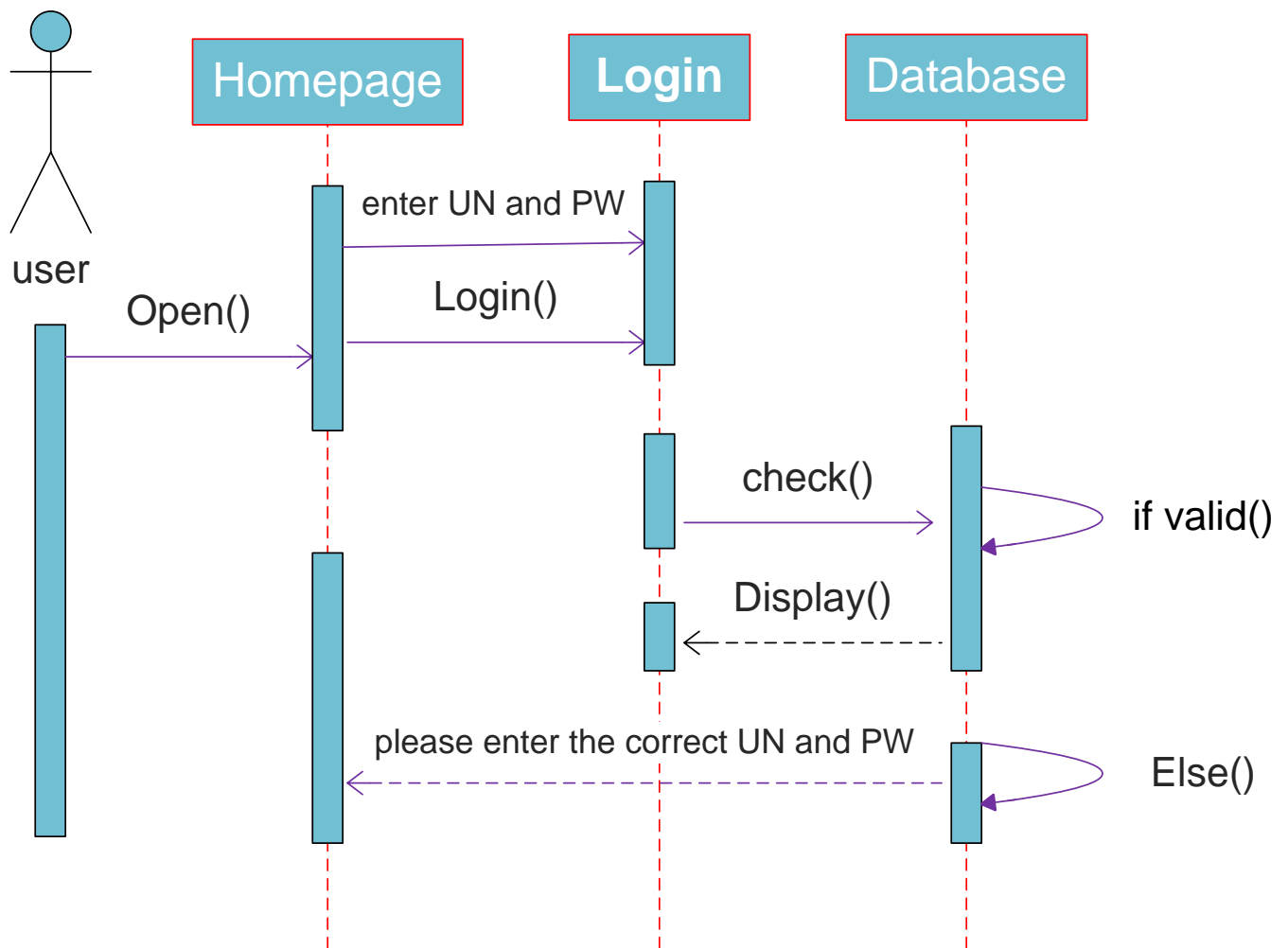


Figure 2: login sequence diagram

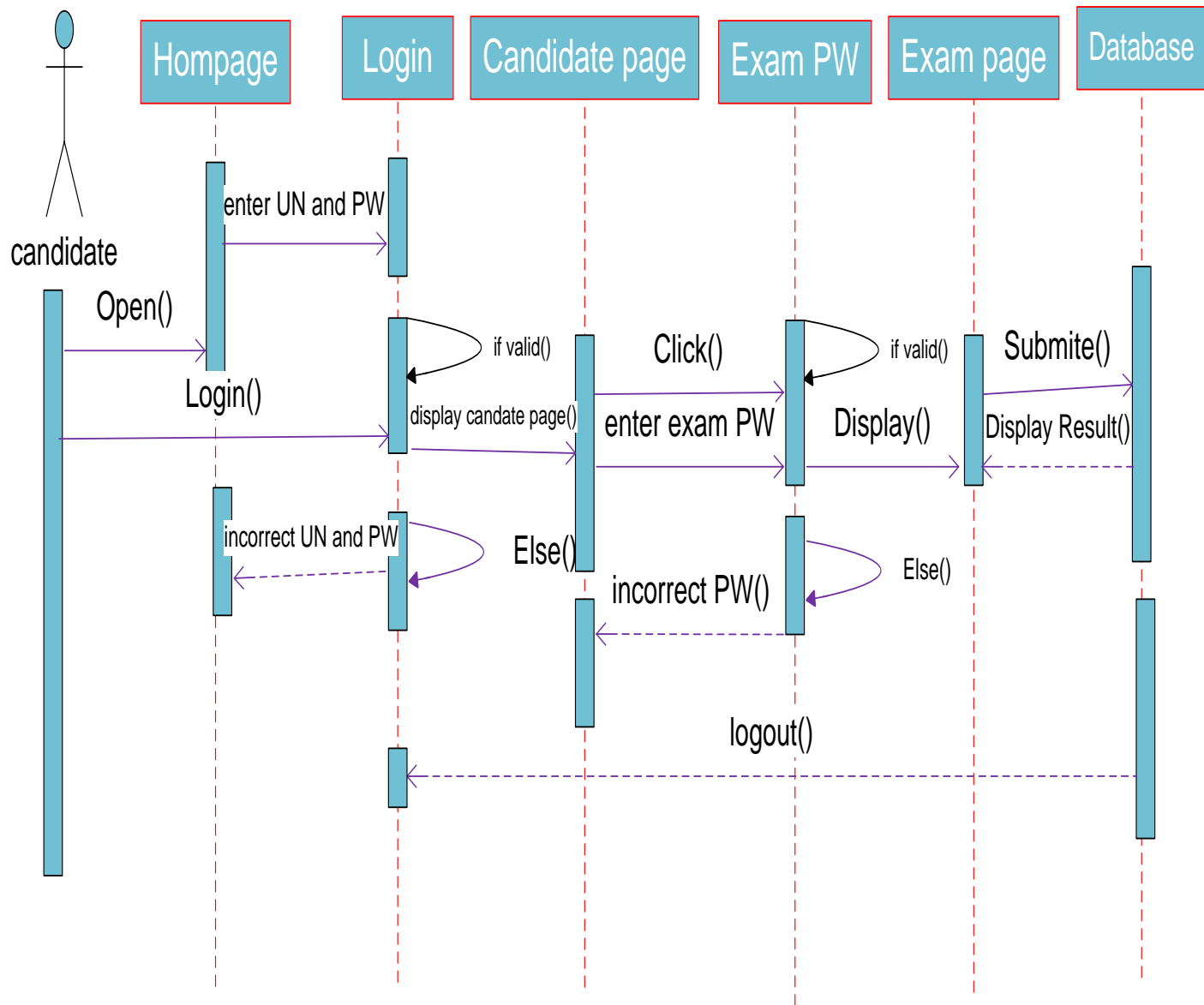


Figure 3: Sequence Diagram for take exam

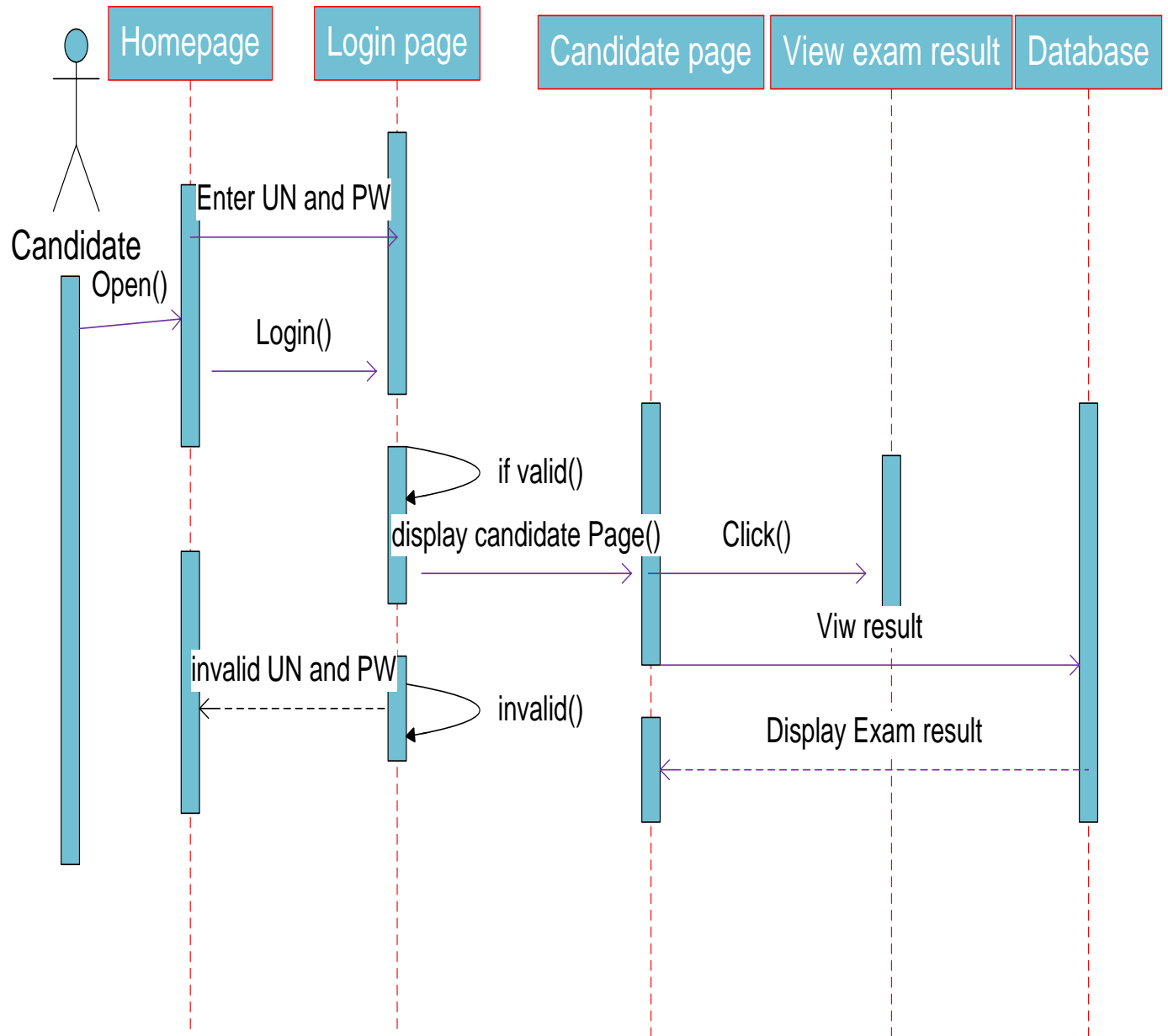


Figure 4: sequence diagram of view exam result

2.5.6 Analysis of class diagram

Class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. In this class diagram the team members try to describe the types of object in the system and the various kinds of static relationships that exist among them as well as depicted the detailed understanding of problem domain of the system. These Class diagrams are developed based on the functional requirement. [5]

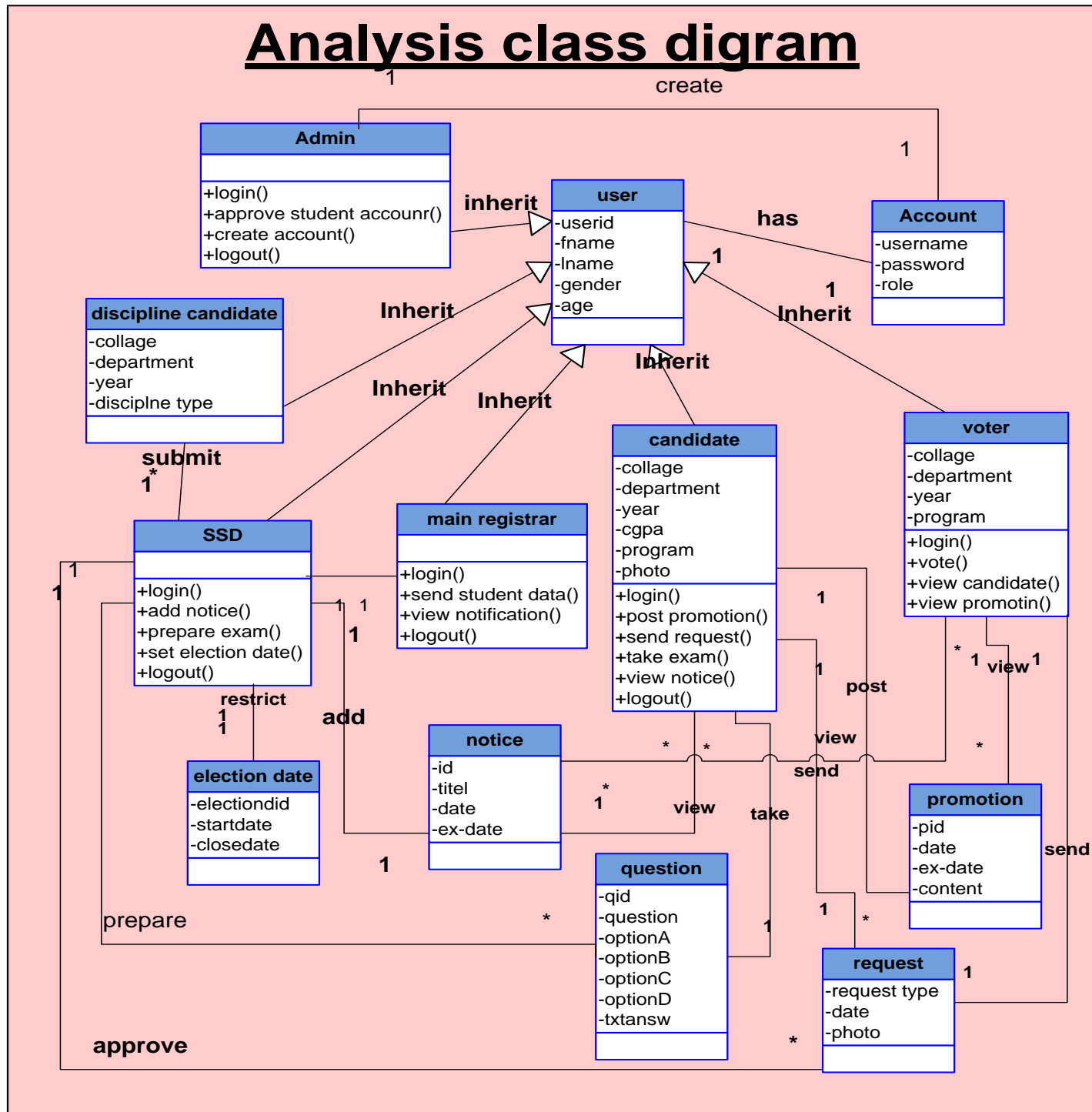


Figure 5: analysis class diagram

2.5.7 Activity diagram invalid

Activity diagrams are used to document the logic of a single operation/method, a single use case, or the flow of a business process. Activity diagrams essentially a flowchart showing flow of control from activity to activity. It includes modeling the sequential process. It also includes modeling the flow of an object as object as it moves from one state to another state at different points in the flow of control. [4]

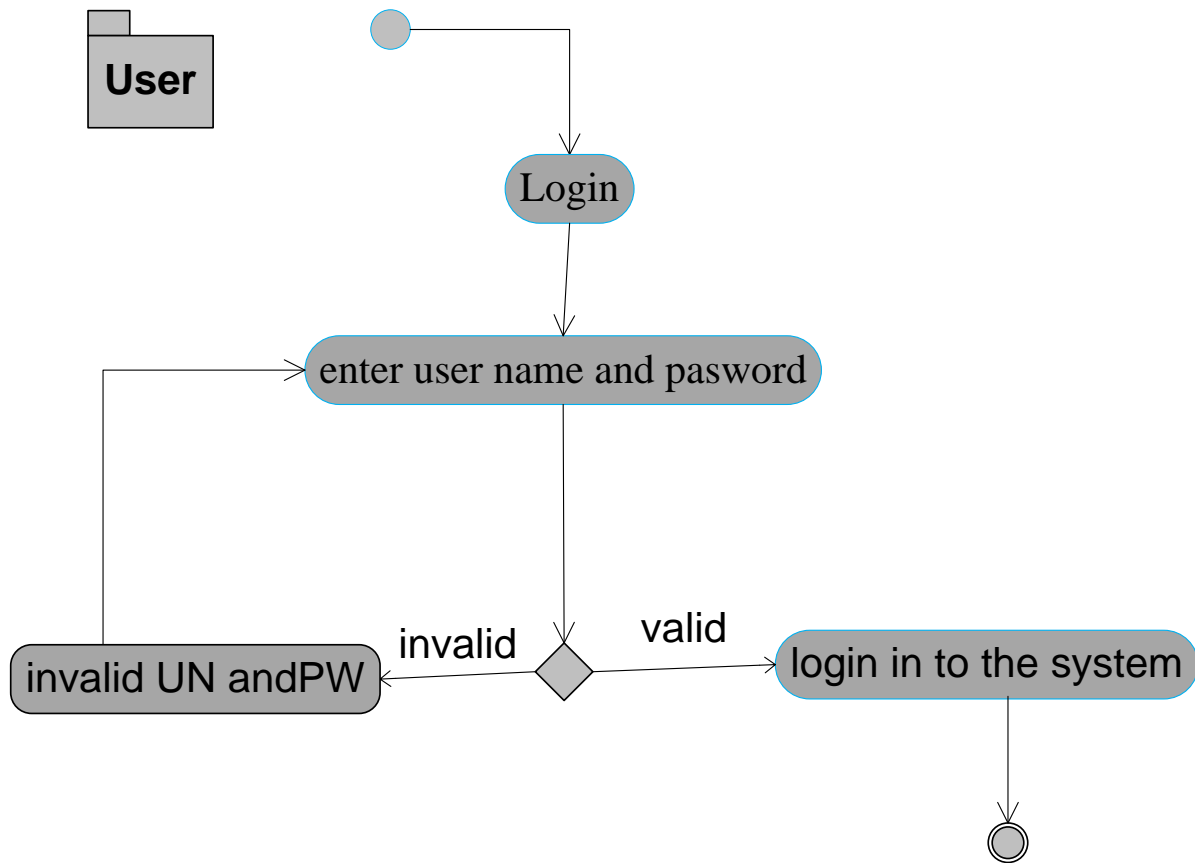


Figure 6: Activity Diagram of login

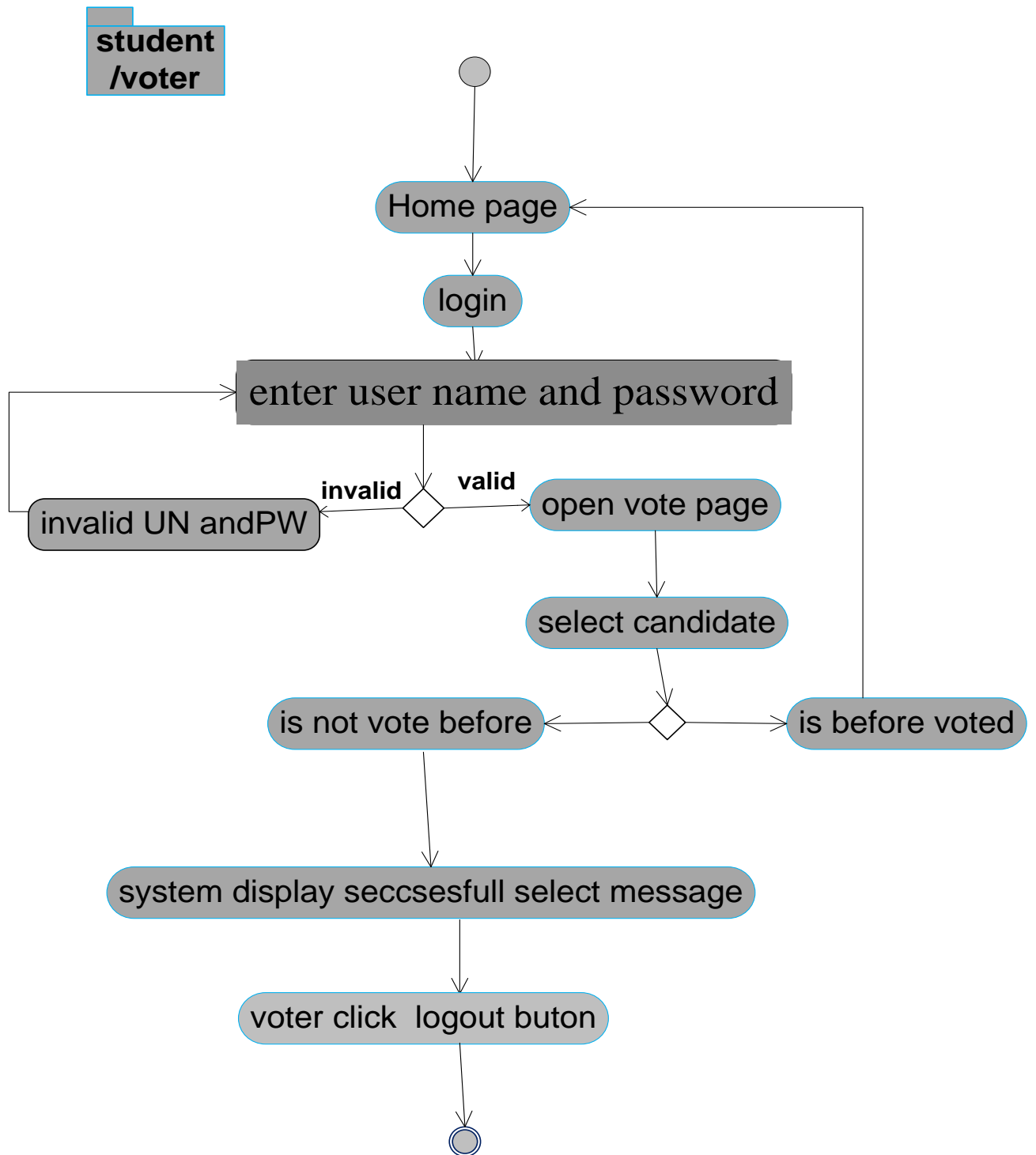


Figure 7: Activity diagram of vote

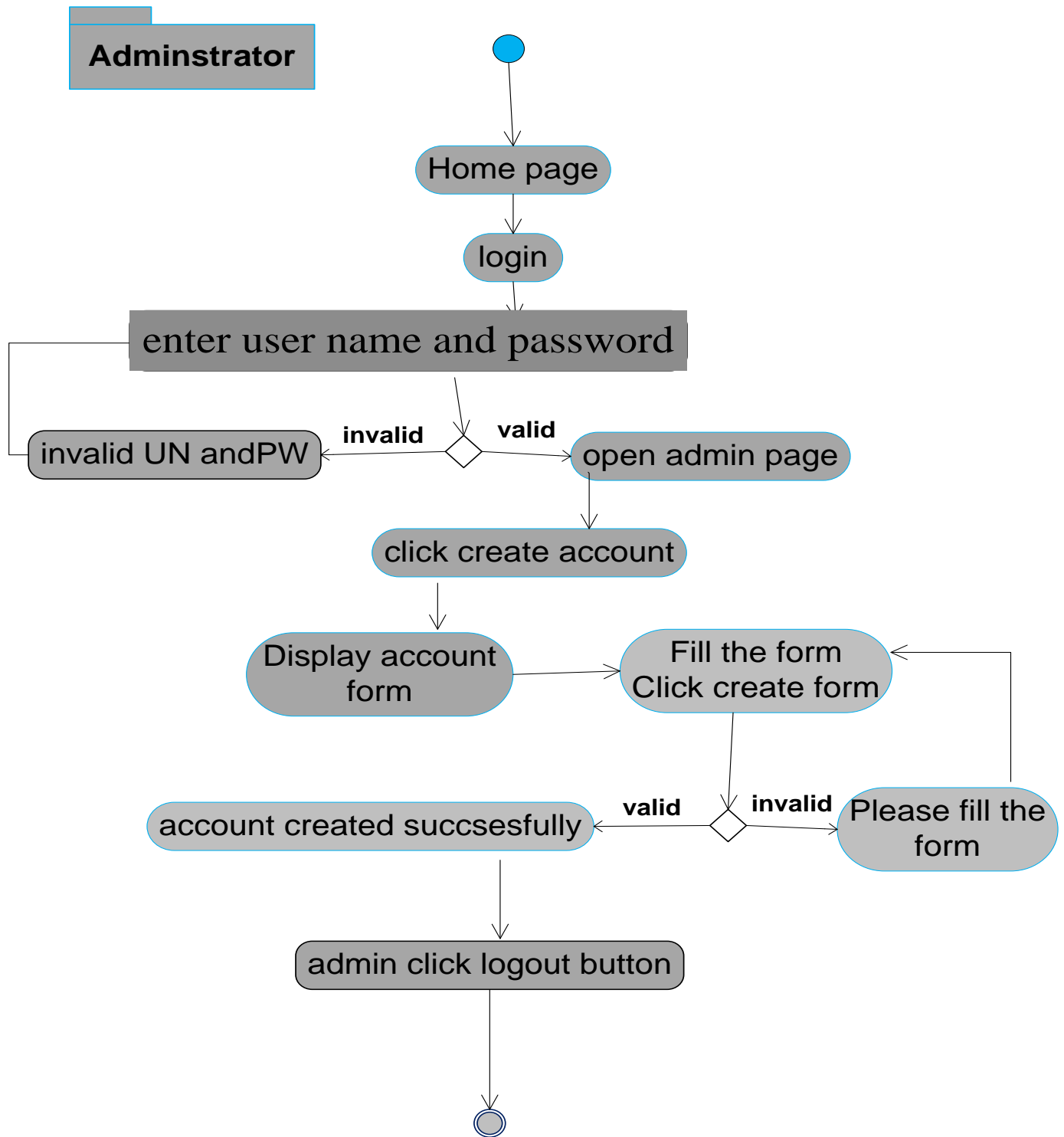


Figure 8: Activity diagram of create account

CHAPTER THREE

3 System design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. The purpose of design is to determine how the system is going to build and to obtain the information needed to drive the actual implementation of the system. It focuses on understanding the model how the software will be built. System design is the detail investigation of system elements from logical view. [5]

3.1 Design class diagram

Class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes:

- their attributes
- operations (or methods)
- And the relationships among the classes.

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). [5]

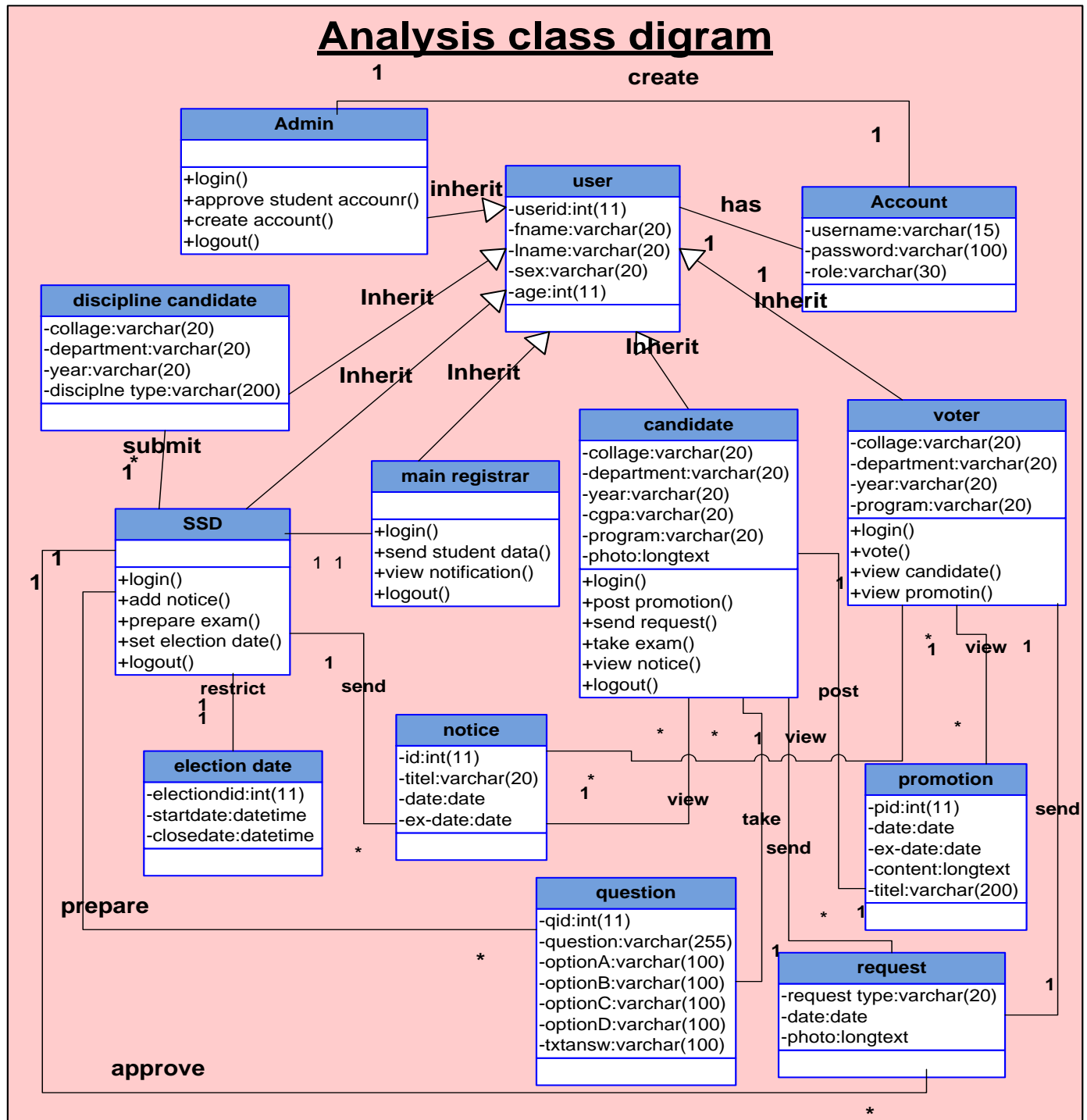


Figure 9: design class diagram

3.2 Class diagram description

Documenting class model is used to document the detail about a class and method. The following are the documentation of the class model for the student information system.

Table 8: class description of student service directorate

Attribute	Description	Data type
Sid	Sid is the unique identifier of student service director.	Int(11)
Fname	The first name of student service director.	Varchar(20)
Lname	The second name of student service director	Varchar(20)
sex	Represent the gender of student service director.	Varchar(10)
age	Represent the age of student service directorate.	Int(11)

Table9: method for student service directorate

Operation	Description
login	Describes how the users log into the system by using username and password.
Add notice	Used to post the election date can start
Prepare exam	Used to prepare exam that the candidate can measure.
Set election date	Used to set when election can be started at specified date.
Approve request	Used to approve candidate where candidate can send request.
View result	Used to view the result in election.
View student data	View student data send by registrar.
logout	User can be logout in the system.

Table 10: class description of candidate

Attribute	Description	Data type
Cid	Represent the unique identification of candidate	Int(11)
Fname	The first name of candidate	Varchar(20)
Lname	The second name of candidate	Varchar(20)
sex	Represent the gender of candidate.	Varchar(10)
collage	The collage of what the candidate can occur.	Varchar(20)
department	Represent the department of candidate.	Varchar(20)
Year	Represent the year of candidate.	Varchar(20)
program	Represent the candidate must be regular student.	Varchar(20)
cgpa	Represent the cumulative of candidate	Varchar(20)
photo	The photo that the candidate express	longtext

Table 11: method description of candidate

operation	description
Send request	Candidate can send account request to adminthat participate in election.
Take	Candidate can take exam to participate in election.
View exam result	Candidate can view exam result and election result.
login	Describes how the users log into the system by using username and password
logout	User can be logout in the system.
Post promotion	Candidate can post advert or promotion that can be elected.

Table 12: class description of voter

vid	Represent the unique identification of voter	Int(11)
Fname	The first name of voter	Varchar(20)
Lname	The second name of voter	Varchar(20)
sex	Represent the gender of voter.	Varchar(10)
collage	The collage of what the voter can occur.	Varchar(20)
department	Represent the department of voter.	Varchar(20)
Year	Represent the year of voter.	Varchar(20)
program	Represent the voter must be regular student.	Varchar(100)
photo	The photo that the candidate express	longtext

Table 13: method description of voter

operation	description
login	Describes how the users log into the system by using username and password.
vote	Used to vote candidate.
View result	Used to view the result in election.
View promotion	Voter can view candidate promotion.
View candidate	Voter can view candidate to elect their representative.
logout	Voter can be logout the system after voting.

Table 14: class description of main registrar

Attribute	Description	Data type
Rid	Represent the unique identification of main registrar.	Int(11)
Fname	The first name of registrar.	Varchar(20)
Lname	The second name of registrar	Varchar(20)
sex	The gender of registrar.	Varchar(10)
age	The age of main registrar	Int(11)

Table 15: method description of main registrar

operation	description
login	Describes how the users log into the system by using username and password.
Send student data	Used to send student information to student service directorate.

3.3 Database design\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. [8]

Table 16: student service director table

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<u>Sid</u>	varchar(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 2	Fname	varchar(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 3	Lname	varchar(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 4	age	int(11)			No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 5	sex	varchar(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index

Table 17: candidate registration table

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<u>cid</u>	varchar(20)	latin1_swedish_ci		No			Change Drop Primary Unique Index
<input type="checkbox"/> 2	fname	varchar(20)	latin1_swedish_ci	Yes	NULL			Change Drop Primary Unique Index
<input type="checkbox"/> 3	lname	varchar(20)	latin1_swedish_ci	Yes	NULL			Change Drop Primary Unique Index
<input type="checkbox"/> 4	gender	varchar(20)	latin1_swedish_ci	Yes	NULL			Change Drop Primary Unique Index
<input type="checkbox"/> 5	age	int(11)		Yes	NULL			Change Drop Primary Unique Index
<input type="checkbox"/> 6	collage	varchar(20)	latin1_swedish_ci	Yes	NULL			Change Drop Primary Unique Index
<input type="checkbox"/> 7	department	varchar(20)	latin1_swedish_ci	Yes	NULL			Change Drop Primary Unique Index
<input type="checkbox"/> 8	year	varchar(20)	latin1_swedish_ci	Yes	NULL			Change Drop Primary Unique Index
<input type="checkbox"/> 9	cgpa	varchar(20)	latin1_swedish_ci	Yes	NULL			Change Drop Primary Unique Index
<input type="checkbox"/> 10	program	varchar(100)	latin1_swedish_ci	No	None			Change Drop Primary Unique Index

Table 18: promotion table

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<u>pid</u>	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index
<input type="checkbox"/> 2	Dates	date			Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 3	Ex_Dates	date			Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 4	Title	varchar(200)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 5	Content	longtext	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 6	userid	varchar(20)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index

Table 19: voter registration table

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<u>vid</u>	varchar(20)	latin1_swedish_ci		No			Change Drop Primary Unique Index
<input type="checkbox"/> 2	fname	varchar(20)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 3	lname	varchar(20)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 4	gender	varchar(20)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 5	age	int(11)			Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 6	collage	varchar(20)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 7	department	varchar(20)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 8	year	varchar(20)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 9	program	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index

Table 20: voter promotion table

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	<u>PVid</u>	varchar(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
2	Pid	varchar(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
3	Vid	varchar(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index

Table 21: request table

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<u>Student ID</u>	varchar(30)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 2	fname	varchar(15)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 3	lname	varchar(15)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 4	sex	varchar(10)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 5	age	int(11)			No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 6	collage	varchar(30)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 7	department	varchar(30)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 8	year	varchar(10)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 9	cgpa	float			No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 10	username	varchar(30)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 11	password	longtext	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 12	requesttype	varchar(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 13	photo	longtext	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 14	approved	varchar(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 15	date	datetime			No	None		Change Drop Primary Unique Index

Table 22: account table

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	<u>userid</u>	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index
<input type="checkbox"/> 2	fname	varchar(20)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 3	lname	varchar(20)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 4	sex	varchar(10)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 5	age	int(11)			Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 6	username	varchar(15)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 7	password	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index
<input type="checkbox"/> 8	role	varchar(30)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 9	status	int(11)			Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 10	photo	longtext	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/> 11	votestatus	varchar(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index

3.4 User interface design

User interface design is the design of computers, appliances, machines, mobile communication devices, software applications, and websites with the focus on the user's experience and interaction. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals what is often called user centered design. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design may be utilized to support its usability. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs. [6]

**Online Student Union Voting System
For
Debre Markos University**

Home Contactus Feedback Mission Vission Help Login

Just Login Here

User Name:

Password:

 ☐ Show Password

[Forget Password](#)

Copyright © 2018 Online Student Union Voting System For Debre Markos University [Designed By](#)

Figure 10: login page

This interface describes login page user interface when unauthorized user cannot login into the system but only authorized user that can register into the database can login into the system by entering user name and password that perform their activity. so security can be perform.

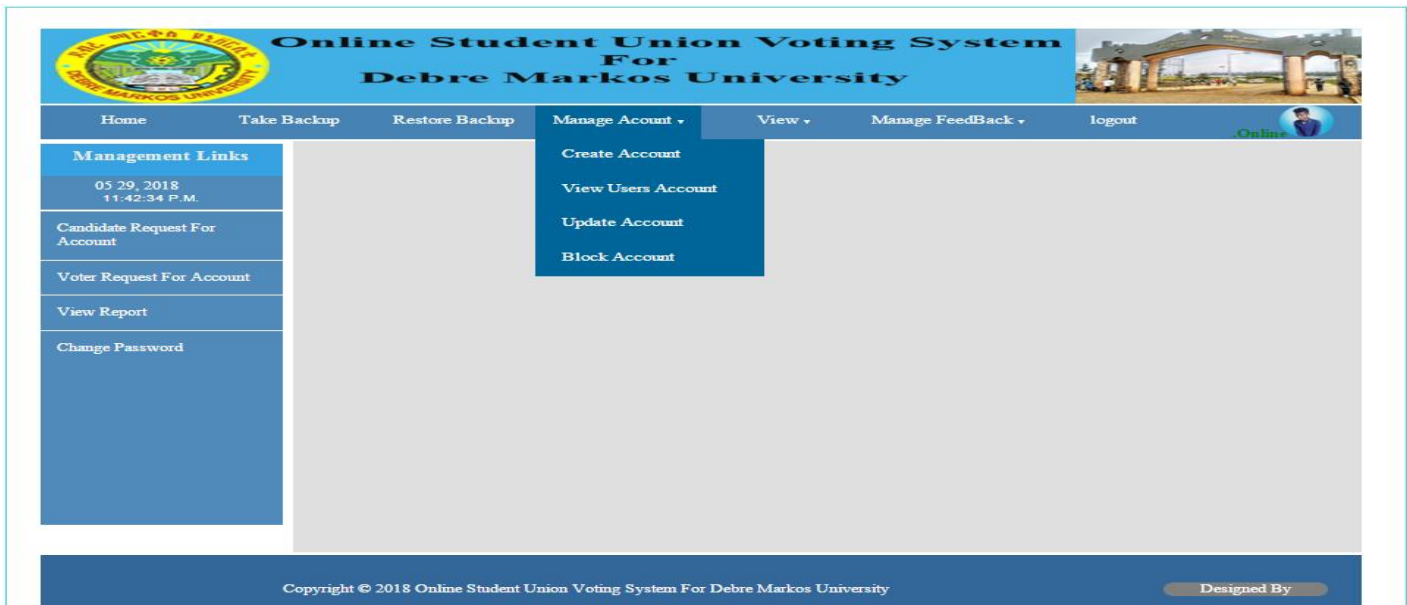


Figure 11: admin user interface

This page describes admin user interface. In admin page, the actor admin performs different activities. These are: create new account for user, update account, approve request for account that is sent by the candidate and voter, block account, take backup and restore backup.

3.5 System architecture

3.5.1 Deployment diagram

Deployment diagram is a structure diagram which shows architecture of the system as deployment or distribution of software artifacts to deployment targets. Deployment diagrams model the physical architecture of a system. It also shows the relationship between the software and hardware. A deployment diagram shows how and where the system is to be deployed; that is, its execution architecture. [7]

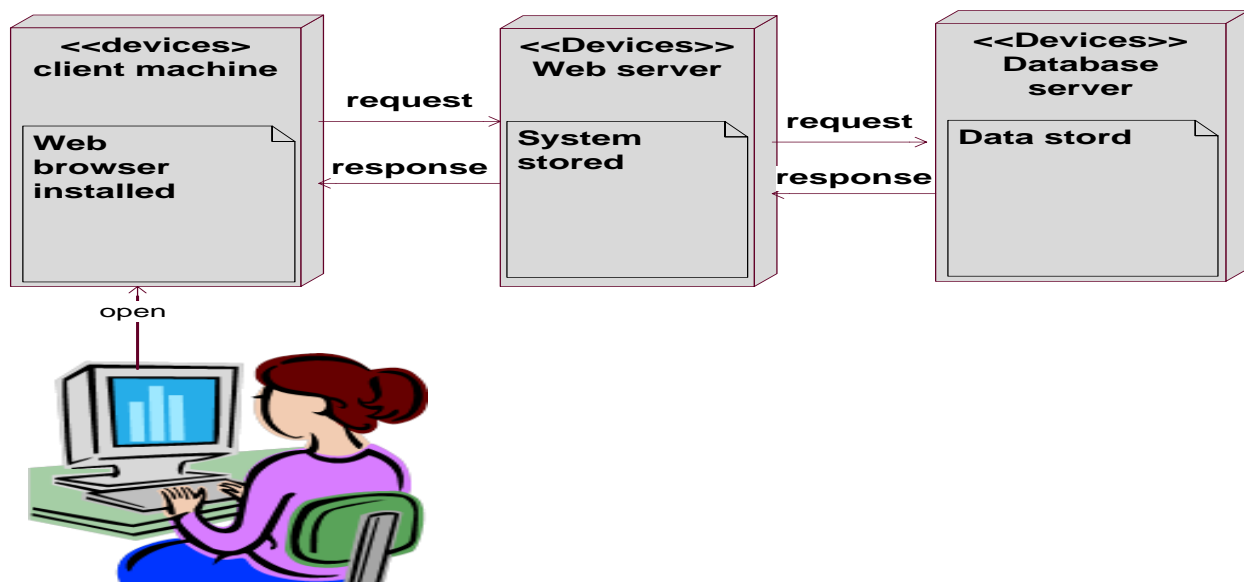


Figure 12: deployment diagram

CHAPTER FOUR

4 Implementation

The implementation document helps users on how to work with the system. It acts as a user manual and it helps users not to be the system confused with. It includes sample forms and some selected fragment code. It gives the users a brief overview of the system. This document describes the project implementation for developing online student union voting system. The project implements PHP, MySQL and standard HTML. The project will be capable of running on standard internet web browsers. The interface for the project will provide a good navigation to user of the system with nice look and feel.

4.1 Overview of programing language

This project used php server side programming technology integrated MySQL database with a programming language. Because:

- ✱ The code and its syntax are simple to understand.
- ✱ It's quick to develop in PHP.
- ✱ It's available for free.
- ✱ It runs on many different operating systems.
- ✱ Easy to access other web-based tools through PHP (i.e. Google).

In general PHP is a widely used open source general purpose scripting language that is especially suited for web development and can be embedded into HTML. Hypertext Preprocessor (PHP) is a server side scripting language designed for web development but also used as a general purpose programming language.

4.2 Algorithm used

❖ Register user

Click on register user link

Registration form is displayed

Fill user profile data

Click on register button

If (valid)

Display successful message.

Else display invalid input message

❖ Update account

Click on update account link

Form is displayed

Search for user ID

Fill new information for the user

Click on update button

If (valid)

Display successful message.

Else display invalid input message

4.3 Sample code

```
<?php
session_start();
if(isset($_session['counter']))
    $_session['counter']+=1;
else
    $_session['counter']=1;
?>

<html>
<link href="css/allcss.css" rel="stylesheet" type="text/css"/>
<head>
<title>Online Student Union Voting System</title>
<style>
#login
{
margin-top: 0px;
    background-color: #718076;
    height: 220px;
    width: auto;
}
#content
{background-color: #dfdfff;
    width: 740px;
    height: 550px;
    margin-left: 230px;
    margin-top: 7px;
    overflow-x: hidden;
    overflow-y: hidden;
```

```

}

#rightside
{background-color: #508abb;

    width: 220px;

    height: 546px;

    float: right;

    margin-top: 5px;

    margin-right: 40px;

}#leftside
{background-color: #508abb;

width: 220px;

height: 540px;

float: left;

margin-top: 10px;

}

#calander
{margin-top: 1px;

    padding-top: 0px;

    width: auto;

    height: 190px;

}#myBtn {

    width: 60px;

    display: none;

    position: fixed;

    bottom: 13px;

    right: 80px;

    font-size: 18px;

    border-radius: 150px;;

```

```

        outline: none;

        color: #fff;

        transition: 0.8s ease;

        opacity: 0.8;

        background-color: #96a0a0;

        cursor: pointer;

        padding: 2px;

        height: 50px;}

#myBtn:hover {

    background-color: black;
}input[type=text], input[type=password] {

    width: 100%;

padding: 12px 20px;

    margin: 8px 0;

    display: inline-block;

    border: 1px solid #dedede;

    box-sizing: border-box;
}hr.style-five {

    border: 0;

    height: 0; /* Firefox... */

    box-shadow: 0 0 10px 1px black;
}hr.style-five:after { /* Not really supposed to work, but does */

    content: "\00a0"; /* Prevent margin collapse */
}</style>

<script>

function show_password() {

    var x = document.getElementById("password");

    if (x.type === "password") {

```

```

        x.type = "text"; }

    else { x.type = "password";}}

// Get the modal

var modal = document.getElementById('id01');

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {

    if (event.target == modal) {

        modal.style.display = "none"; }}

// When the user scrolls down 20px from the top of the document, show the button
window.onscroll = function() {scrollFunction()};

function scrollFunction() {

    if (document.body.scrollTop > 50 || document.documentElement.scrollTop > 50) {

        document.getElementById("myBtn").style.display = "block";

    } else {

        document.getElementById("myBtn").style.display = "none"; }}

// When the user clicks on the button, scroll to the top of the document
function topFunction() {

    document.body.scrollTop = 0;

    document.documentElement.scrollTop = 0;

}</script></head><body>

<div id="container">

    <div id="header">

        <?php

            include("headerhome.php"); ?>

        </div>

        <div id="menu">

            <?php

                include("menu.php");?></div>

```

```

<div id="leftside">

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

</div>

<div id="rightside">

    <div id="login">



</div>

<div id="calander">

<b style="color: #12d7ed;">Calander</b>

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

    </div> </div>

    <div id="content"><?php include("popuplogin.php");?>

</div><div id="footer">

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

    <button onclick="topFunction()" id="myBtn" title="Go to top"></button>

</body>

</html>

```

CHAPTER FIVE

5 Testing

Testing is the main phase when we develop a system. It is the final process that deals how much the system is accurate, correct and functional. The requirements that are tested by the project team during the implementation are correctness, performance, accuracy, security and others.

- ✱ Testing correctness:-correctness determines how users can interact with the system and how the system should behave when it is used correctly. The project team checks the system accuracy by inserting valid as well as invalid data.
- ✱ Testing Performance:-it is performed to determine the system parameters in terms of responsiveness and stability under various workloads. Performance testing measures the quality attributes of the system. The project team verifies system Performance in terms of its response time.
- ✱ Testing security: To insure the security of the system, the team tries to use invalid user name and password but the system allows only authorized users to login.

5.1 Sample test

5.1.1 Sample test for login page

To test for login page when tries to use invalid user name and password four times then the system can inactive to wait for a few second but the system allows only authorized users to login.

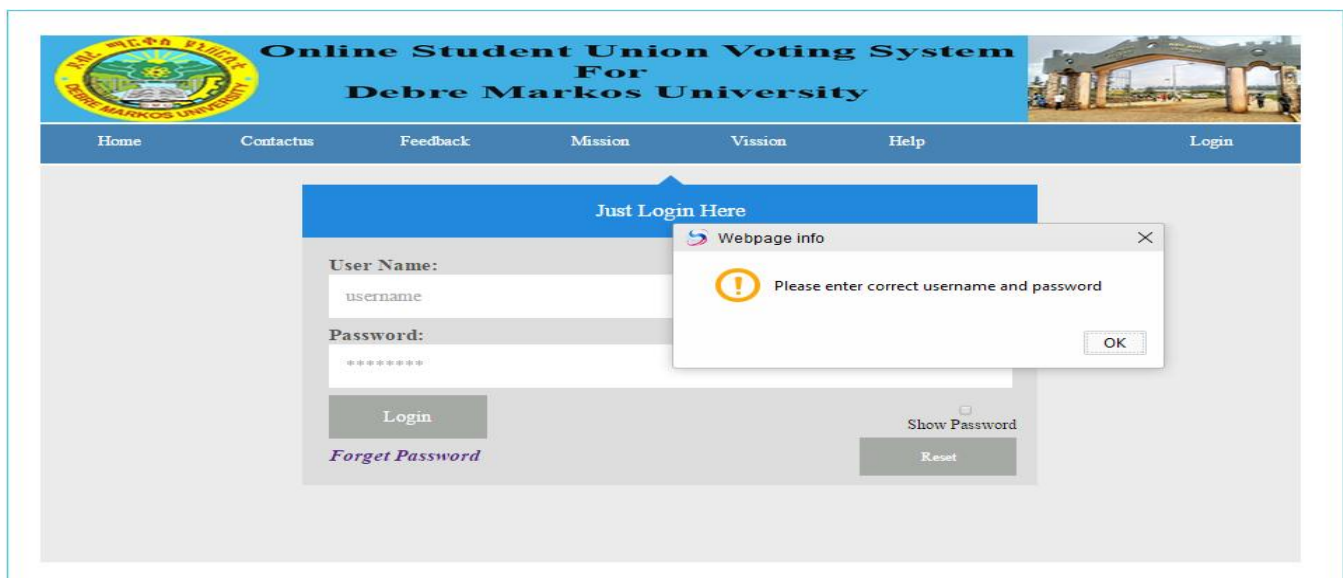


Figure 13: sample test for login page

5.1.2 Sample test for double voting

When one voter vote two or more candidate the system can control this condition that are not vote two times but one voter vote only one candidate.

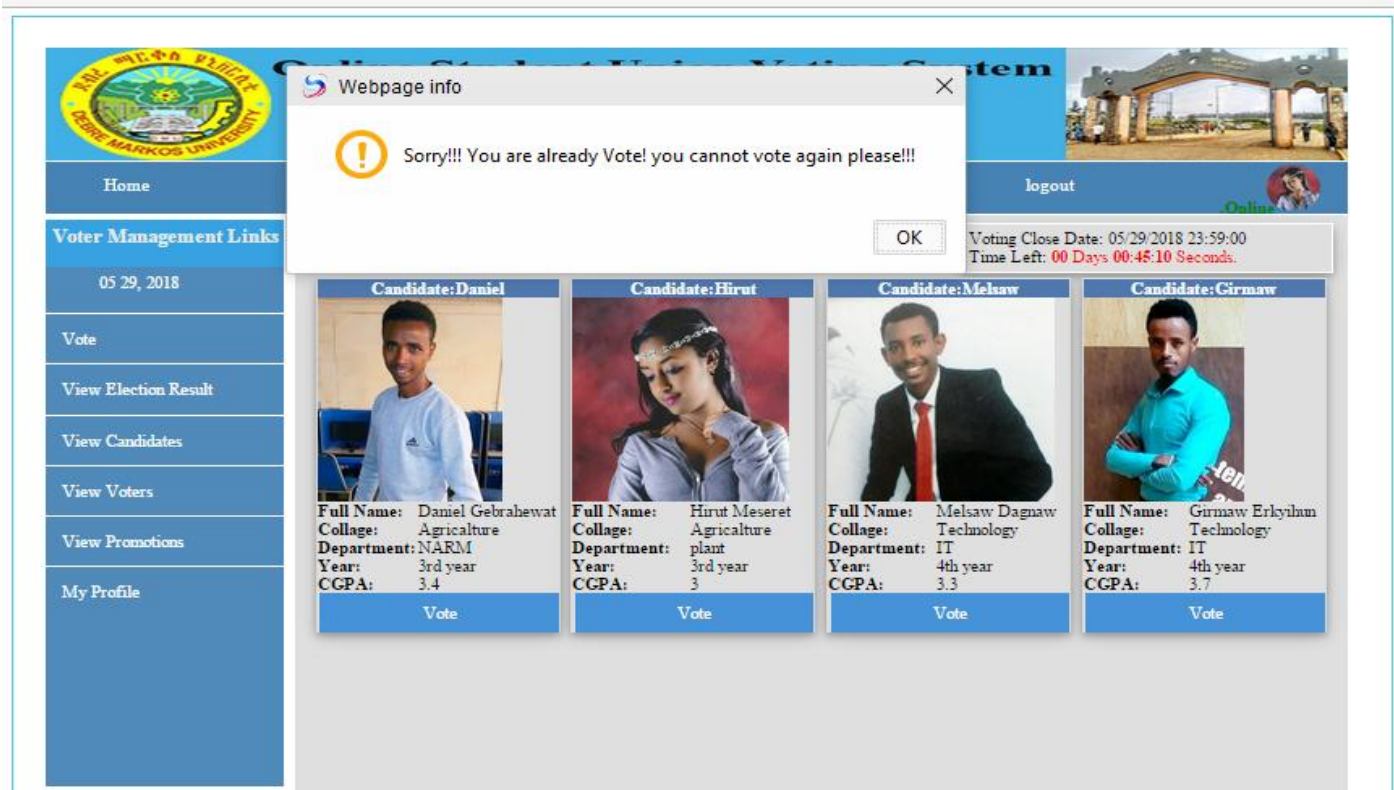


Figure 14: sample test for double voting

6 CHAPTER SEX

6.1 Conclusion and recommendation

6.1.1 Conclusion

Online voting system is a web application that allows voter can vote their representative easily in voting systems. In online student union voting system election can be conducted in free and fair manner in every two years in secret ballots. The aim of our project is to develop an interactive, electronic, voting system for Debre Markos student union members' election with which students can vote using a web browser. It automates the existing manual activities like voter registration, candidate & election process announcements, voting and vote counting.

6.1.2 Recommendation and Future enhancement

According to scope of the project the team develops web based application. Because of the time constraint, the system may have limitation but in the future the team believes that this system should be fully operational by adding some functionality that are not included in the proposed system. The team also wants to recommend this project can be expanded and more automated with additional functionalities by integrating with many new technologies. Generally the team recommends being included the following functionality:

- ✱ System can be responsive.
- ✱ Develop to Amharic language.
- ✱ Develop security camera when candidate take the exam.

References

- [1] Modern system analysis and design third edition by JEFFERY A.HOFFER, JOEY F.GEORGE, JOSEPH S.VALACLCH (system analysis and system requirement actor and use case) on December 12/04/2010 EC.
- [2] Modern system analysis and design third edition by JEFFERY A.HOFFER, JOEY F.GEORGE, and JOSEPH S.VALACLCH (functional and nonfunctional requirement.) on December 12/04/2010 EC.
- [3] Modern system analysis and design third edition by JEFFERY A.HOFFER, JOEY F.GEORGE, and JOSEPH S.VALACLCH (design sequence diagram.) on December 25/04/2010 EC.
- [4] Modern system analysis and design third edition by JEFFERY A.HOFFER, JOEY F.GEORGE, JOSEPH S.VALACLCH (about how to design activity diagram.) on Decembers 27/04/2010
- [5] Modern system analysis and design third edition by JEFFERY A.HOFFER, JOEY F.GEORGE, and JOSEPH S.VALACLCH (for the analysis and design class diagram detail operation, attribute, inheritance and association.) on December 26/04/2010 EC and On January 1/05/2010 EC.
- From internet
- [6].UserInterface <https://www.google.com/search?q=user%20interface&cad=h>. On January 7/05/2010 EC.
- [7].deployment diagram <https://www.google.com/search?q=deployment%20diagram&cad=h> On January 10/05/2010 EC.
- [8] Database design https://en.wikipedia.org/wiki/Database_design on December 26/04/2010 EC and On January 1/05/2010 EC.