

UNIVERSITY OF GONDAR

FACULTY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCES

**WEB BASED ONLINE STUDENT UNION VOTING SYSTEM FOR UNIVERSTY OF GONGAR**

INDUSTRIAL PROJECT

BY

PROJECT PARTCIPANTS ID NUMBER

1. Ebistie Mengistie …………………….…..GUR/3771/08
2. Mulatu Mekonnen………..………………GUR/3908/08
3. Esubalew Marnew…………………..……..GUR/3789/08
4. Abdulhamid Mohammed ...……….………GUR/3634/08
5. Tinsae Mathewos………………….……...GUR/3996/08

PROJECT ADVISOR: Mr.BIRHANU

DATE JUNE 18/2011 E.C

# 

# DECLARATION

This is to declare that the project works which is done under the supervision of instructor Birhanu having the title of online student union for university of Gondar is the sole contribution of:

Ebistie Mengistie

Mulatu Mekonnen

Esubalew Marnew

Abdulhamid Mohamm

Tinsae Mathewos

No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and cited properly. We will be responsible and liable for any consequence if violation of this declaration occurs. Date: 18/2011 E.C

Group members:

FULL NAME SIGNITUR

Ebistie Mengistie \_\_\_\_\_\_\_\_\_\_

Mulatu Mekonnen \_\_\_\_\_\_\_\_\_\_

Esubalew Marnew \_\_\_\_\_\_\_\_\_\_\_

Abdulhamid Mohamm \_\_\_\_\_\_\_\_\_\_\_

Tinsae Mathewos \_\_\_\_\_\_\_\_\_\_\_

# CRETIFICATE

This is to certify that this BSc industrial project entitled online student union voting system for university of Gondar written by:

Ebistie Mengistie

Mulatu Mekonnen

Esubalew Marnew

Abdulhamid Mohamm

Tinsae Mathewos

is approved by me for submission. I certify further, to the best of my knowledge, the project represents work carried out by the students.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Date that Name and Signature of Supervisor

# 

# ACKNOLEDGEMMENT

This project is finalized through the priceless effort of many individuals. So first and foremost we would like to thank you our God the owner and giver of wisdom who helps us to start this project. Secondly our heart full gratitude goes to several persons who helped us directly or indirectly to conduct this project work. The completion of this undertaking could not have been possible without the participation and assistance of UOG student union, UOG student service dean (SSD) and some individuals those give us all required information that we want So, we have no word to express our deepest respect to them whom helped us in different aspect. The last but our greatest respect and grateful thanks goes to our project adviser **Mr. Birhanu** **Ewnetu** for his sincere guidance and inspiration of the completion of this project. He has priceless contribution in this project by organizing and rearranging the document.

Contents

[DECLARATION i](#_Toc12345960)

[CRETIFICATE ii](#_Toc12345961)

[ACKNOLEDGEMMENT iv](#_Toc12345962)

[LIST OF TABLE x](#_Toc12345963)

[LIST OF FIGURE xi](#_Toc12345964)

[ACRONYMS AND ABBRIVATIONS xii](#_Toc12345965)

[ABSTRACT xiii](#_Toc12345966)

[CHAPTER ONE 1](#_Toc12345967)

[1. PROJECT PROPOSAL 1](#_Toc12345968)

[1.1 Introduction 1](#_Toc12345969)

[1.2 Statement of the Problem and Justification 2](#_Toc12345970)

[1.3 Objective of the Project 3](#_Toc12345971)

[1.3.1 General Objective 3](#_Toc12345972)

[1.3.2 Specific Objective 4](#_Toc12345973)

[1.4 Scope and Limitation of the Project 4](#_Toc12345974)

[1.4 .1 Scope of the Project 4](#_Toc12345975)

[1.4.2 Limitation of the Project 4](#_Toc12345976)

[1.5 System Development Methodology 4](#_Toc12345977)

[1.5.1 Investigation (Fact-Finding) Method 6](#_Toc12345978)

[1.5.2 System Development Tools 6](#_Toc12345979)

[1.6 Significant of the Project 7](#_Toc12345980)

[1.7 Beneficiary 7](#_Toc12345981)

[1.8 Time Schedule 8](#_Toc12345982)

[CHAPTER TWO 9](#_Toc12345983)

[2. REQUIREMENT ANALYSIS 9](#_Toc12345984)

[2.1 Introduction 9](#_Toc12345985)

[2.2 Current System 9](#_Toc12345986)

[2.2.1 Major Function of Current System Description 9](#_Toc12345987)

[2.2.2 Problem of the Current System 10](#_Toc12345988)

[2.3 Requirement Gathering 11](#_Toc12345989)

[2.3.1 Requirement Gathering Methodology 11](#_Toc12345990)

[2.3.1.1. Observation 11](#_Toc12345991)

[2.3.1.2. Interviewing 11](#_Toc12345992)

[2.3.1.2.1 Open-Ended Questions 11](#_Toc12345993)

[2.3.1.2.2 Close-Ended Questions 12](#_Toc12345994)

[2.3.2 Results Found 12](#_Toc12345995)

[2.4 Business Rule 12](#_Toc12345996)

[2.5 Proposed System 13](#_Toc12345997)

[2.5.1 Overview 13](#_Toc12345998)

[2.5.2Requirement Specification 14](#_Toc12345999)

[2.5.2.1 Functional Requirements 14](#_Toc12346000)

[2.5.3 Non-functional Requirement 16](#_Toc12346001)

[2.3.3.1 User Interface and Human Factors 17](#_Toc12346002)

[2.5.3.2 Documentation 17](#_Toc12346003)

[2.5.3.3 Hardware Consideration 17](#_Toc12346004)

[2.5.3.4 Performance Characteristics 17](#_Toc12346005)

[2.5.3.5 Error Handling and Extreme Conditions 17](#_Toc12346006)

[2.5.3.6 Quality Issues 18](#_Toc12346007)

[2.5.3.7 System Modifications 18](#_Toc12346008)

[2.5.3.8 Physical Environment 18](#_Toc12346009)

[2.5.3.9 Security Issues 18](#_Toc12346010)

[2.5.3.10 Resource Issues 18](#_Toc12346011)

[3.6 System Model 19](#_Toc12346012)

[3.6.1 Scenario 19](#_Toc12346013)

[2.6.2 Use Case Modeling 23](#_Toc12346014)

[2.6.2.1 Use case Diagram 23](#_Toc12346015)

[2.6.2.2 Description of Use Case Model 25](#_Toc12346016)

[2.6.3 Activity Diagram 38](#_Toc12346017)

[2.6.4 Object Model 42](#_Toc12346018)

[2.6.4.1 Data Dictionary 42](#_Toc12346019)

[2.6.4.2 Class Modeling 45](#_Toc12346020)

[2.6.4.2.1 Class Diagram 45](#_Toc12346021)

[2.6.4.3 Dynamic Modeling 46](#_Toc12346022)

[2.6.4.3.1 Sequence Diagram 47](#_Toc12346023)

[2.6.4.3.2 State Chart Diagram 53](#_Toc12346024)

[2.7 User Interface 53](#_Toc12346025)

[CHAPTER THREE 57](#_Toc12346026)

[3. SYSTEM DESIGN 57](#_Toc12346027)

[3.1 Introduction 57](#_Toc12346028)

[3.2 Design Goal 57](#_Toc12346029)

[3.3 Current Software Architecture 58](#_Toc12346030)

[3.4 Proposed Software Architecture 58](#_Toc12346031)

[3.4.1 Overview 58](#_Toc12346032)

[3.4.2 Subsystem Decomposition 60](#_Toc12346033)

[3.4.3 Hardware and Software Mapping 61](#_Toc12346034)

[3.4.4 Persistent Data Management 62](#_Toc12346035)

[3.4.5 Access Control and Security 63](#_Toc12346036)

[3.4.6 Subsystem Service 65](#_Toc12346037)

[3.4.7 Detailed Class Diagram 65](#_Toc12346038)

[3.4.8 Package Diagram 67](#_Toc12346039)

[CHAPTER 4 69](#_Toc12346040)

[4. IMPLEMENTATION 69](#_Toc12346041)

[4.1 Introduction 69](#_Toc12346042)

[4.2 Mapping model to code 69](#_Toc12346043)

[4.2.1 Mapping association 69](#_Toc12346044)

[4.2.2 Mapping the class model to a storage schema is performed 72](#_Toc12346045)

[Primary key 72](#_Toc12346046)

[4.3 source code of major class packages and interface 72](#_Toc12346047)

[CONCLUSION 79](#_Toc12346048)

[RCOMMANDATION 80](#_Toc12346049)

[REFERENCE 81](#_Toc12346050)

# LIST OF TABLE

[Table 1: Table of Login 25](#_Toc12346051)

[Table 2: Table of Send Account Request for Candidate 26](#_Toc12346052)

[Table 3: Table Send Account Request for Voter 27](#_Toc12346053)

[Table 4: Table of View Election Result 28](#_Toc12346054)

[Table 5: Table of Create Account 29](#_Toc12346055)

[Table 6: Table of View Candidate 30](#_Toc12346056)

[Table 7: Table of Generate Report 31](#_Toc12346057)

[Table 8: Table of Give Feedback 32](#_Toc12346058)

[Table 9: Table of Approve Account Request 33](#_Toc12346059)

[Table 10: Table of Set Election Date 34](#_Toc12346060)

[Table 11: Table of Set Student Apply Date 35](#_Toc12346061)

[Table 12: Table of Send Notification 36](#_Toc12346062)

[Table 13: Table of Post Notice 37](#_Toc12346063)

[Table 14: Table of Vote 38](#_Toc12346064)

[Table 15: Table of Data Dictionaries 45](#_Toc12346065)

[Table 16: Table of Access Control and Security 64](#_Toc12346066)

[Table 17 User table 72](#_Toc12346067)

[able 18 Notice table 72](#_Toc12346068)

# LIST OF FIGURE

[Figure 1: Time Schedule 8](#_Toc12346075)

[Figure 2: Use Case Diagram for Online Student Union Voting System. 24](#_Toc12346076)

[Figure 3: Activity Diagram for Login 39](#_Toc12346077)

[Figure 4: Activity Diagram for Post Notice 40](#_Toc12346078)

[Figure 5: Activity Diagram for Approve Request 41](#_Toc12346079)

[Figure 6: Activity Diagram for Generate Report 41](#_Toc12346080)

[Figure 7: Activity Diagram for Vote 42](#_Toc12346081)

[Figure 8: Analysis Class Diagram 46](#_Toc12346082)

[Figure 9: Login Sequence Diagram 48](#_Toc12346083)

[Figure 10: Sequence Diagram for View Election Result. 49](#_Toc12346084)

[Figure 11: Sequence Diagram for Post Notice 50](#_Toc12346085)

[Figure 12: Account Request Sequence 51](#_Toc12346086)

[Figure 13: Sequence Diagram for View Candidate 52](#_Toc12346087)

[Figure 14: Sequence Diagram for Vote 52](#_Toc12346088)

[Figure 15: State Chart Diagram for Post Notice 53](#_Toc12346089)

[Figure 16: User Interface for Login Page 54](#_Toc12346090)

[Figure17: User Interface for Admin Page 55](#_Toc12346091)

[Figure 18: User Interface for Create Account Page 56](#_Toc12346092)

[Figure 19: System Architecture Model for Proposed System 59](#_Toc12346093)

[Figure 20: Component Diagram of Student Union Voting System 61](#_Toc12346094)

[Figure 21: Deployment Diagram 62](#_Toc12346095)

[Figure 22: Persistent Data Management 63](#_Toc12346096)

[Figure23: Component Diagram Subsystem Service 65](#_Toc12346097)

[Figure 24: Detail Class Diagram 66](#_Toc12346098)

[Figure 25: Package Diagram 68](#_Toc12346099)

# ACRONYMS AND ABBRIVATIONS

BR: Business Rule

CGPA: Cumulative Grade Point Average

CPU**:** Central Processing Unit

CSS: Cascading Style Sheet

GB: Gigabyte

GHZ: Giga Hertz

HIV: Human Immunodeficiency Virus

HTML: Hypertext Markup Language

HW: Hardware

OSUVS: Online Student Union Voting System

PHP: Hypertext Preprocessor

Q: Questions

RAM: Random Access Memory

RDBMS: Relational Database Management System

SDD: System Design Document

SPMP: Software Project Management Plan

SQL**:** Structured Query Language

SRS: Software Requirement Specification

SSD: Student Service Dean

STD: Software Test Documentation

SW: Software

UML: Unified Modeling Language

UOG: university of Gondar.

WAMP: Windows Apache MySQL PHP

# ABSTRACT

This documentation explains about online student union voting system which we would develop for University of Gondar. The current voting process is manually and faced many problems. So to overcome these problems, we are going develop 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 University of Gondar students since it is applicable anywhere in the same functionality. Online voting system used to register Candidate, 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 proposal the project team used different data collection methods such as interview, document analysis and practical observation. In order to analyze and design the system we are going to use object oriented approach for both analyzing and designing the new system.

# CHAPTER ONE

# PROJECT PROPOSAL

# 1.1 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.

University of Gondar is founded in 1954, University of Gondar is a non-profit public higher education institution located in the urban setting of the large town of Gondar. This institution has also branch campuses in the following location(s): Maraki. Officially accredited and/or recognized by the Ministry of Education, Ethiopia, University of Gondar (UOG) is a large (uniRank enrollment range: 10,000-14,999 students) coeducational higher education institution. University of Gondar (UOG) offers courses and programs leading to officially recognized higher education degrees in several areas of study. See the uniRank degree levels and areas of study matrix below for further details. This 65 years old higher-education institution has a selective admission policy based on entrance examinations and students' past academic record and grades. The admission rate range is 10-20% making this Ethiopian higher education organization a most selective institution. International students are welcome to apply for enrollment [6].

Student union is most popular in all university. It is first established in Addis Ababa University. University of Gondar is also establishing this union for different purpose. Student union is performing many activities in university. They solve many problems. Such as academic related, food related, finance related, discipline related and other problem is solved by university student union.

In our project Online voting system is a web application that allows Voter can vote their representative easily. In online student union voting system election can be conducted in free and fair manner in every two years.

The president, the vice president, secretary, and the finance committee come to power through direct election for university of Gondar.

s For each Campus the president, vice president, secretary, discipline committee, sport committee, club help committee, gender and HIV committee, finance committee and public communication committee also come to power through direct election.

The Candidate with the highest vote becomes the president, the vice president, secretary, and the finance committee for UOG respectively.

For each campus the Candidate who gets high score will be the president, vice president, secretary, discipline committee, sport committee, club help committee, gender and HIV committee, finance committee and public communication committee respectively. The aim of our project is to develop an interactive and electronic voting system for University

of Gondar student union election by which students can vote using a web browser. It automates the existing manual activities like Voters and Candidates’ registration, voting process 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 consumption in voting process, extravagant resource, geographical limited voting, and undocumented and unstructured information capture. The system makes not only the voting process is easy but also assist students by providing them with information which associated with the student union.

## Statement of the Problem and Justification

Since university of Gondar has many students, it is difficult to directly participate all students in election of student union. That means some students participate in election. These students not keep the whole student’s interest. Register Candidate and Voter on the paper and this paper based or manual system leads to different problems. One of the problem is caused by this manual system is loss of file. This implies that the information is simply put on papers manually. Because of these papers are easily moveable, other may intentionally loss all recorded data or files. Moreover data is not secured from an unauthorized body or it will be damaged as well as it is not well organized, because of this the efficiency of the working system is not good. The student service dean (SSD) requires significant volume of paper which contains student`s information and other related data to manage the system.

Generally the main and observed problems are:-

* **Time consuming**: The process of entering the gathered data into the existing system consumes or takes too much time to perform. Example to collect Candidate data from each department requires much time.
* **Too much paper work**: Since the voting system is manual, the process involves much paper work and paper to storage, which is bulky as the paper becomes much with the students’ number increase. Example to register Candidate from each department requires much paper work and paper act as storage material.
* **Reworking:** As it is manual, the files may be lost and it requires registering the candidate again.
* **Difficult to keep The student’s interest:** Because the system is manual, the Candidate`s information is not fairly verified to all student. Example Candidate`s information can verified for two student those are selected from each class to elect or to be elected in student union but two students cannot satisfy the whole class interest.
* **Geographical limitation**: The election is take place in specific area students who are in remote area or campus will not participate on election.
* **Error prone manual working**: When counter count the vote missing of vote or miscount the number of vote of election.
* **Lack of security**: The existing system leads Poor security because one can get easily the document and can change whatever they want. That means There is manual unauthorized person can update, delete The Candidate`s information.
* **Lack of information:** Notice about student union elections may not be fairly distributed for all the students. Because of The manual information distribution system that makes most of the students could not have the habit of reading notice or accessing information from certain board.

## Objective of the Project

### 1.3.1 General Objective

The general objective of our project is to develop web based online student union voting system for university of Gondar.

### 1.3.2 Specific Objective

In order to achieve main objective, we have the following the specific objectives

* Study and understand the existing system.
* Identify and analysis the problem with the existing system.
* Gathering required information for proposed system by using interview, observation, and document analysis.
* Analyzing the gathered information using SRS document
* Compare and contrast the proposed system with existing system
* Design a new proposed system to solve the problems on existing system.
* Identify functional and non-functional requirements of the proposed System
* Design the proposed system including system architecture using UML diagram.
* Design a user interface for the proposed system.
* Select the appropriate development tools for the system.
* Implement and test the system.

## 1.4 Scope and Limitation of the Project

### 1.4 .1 Scope of the Project

The scope of the project refers to the area where the ongoing system is applied and the overall features of what the new system is capable of doing. The scope of this project is developing web based student union voting system for university of Gondar. This system is cable of registering Candidates and Voters, online vote, calculate votes for each Candidate, posting final result and at the end of election generate report.

### 1.4.2 Limitation of the Project

Generally our proposed system is limited to do the following activities**.**

* The proposed system uses only English language

## 1.5 System Development Methodology

The methodology we used for developing the system using iterative [5]. Because of:

* Requirements that we already used can be modified.
* It is more flexible, less costly to change the scope and requirement without altering the entry requirement.
* It is not exposed for risk because risk pieces are identified and handled during its iteration

### 1.5.1 Investigation (Fact-Finding) Method

The data collection instruments will be used to gather accurate information about the existing system and the requirements for the new system.

In order to get a precise data, the developer used the following data collection techniques.

Those are:-

**Direct Observation**

Observation is best method of investigation. Due to this, the developer physically observed the existing system to get information.

**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 like rules and procedures for processing data.

**Interview**

The additional data would be gathered through the technique of interview. Through this technique we get different information from president of student union, SSD, executive members of the union and student who have been participating on voting before.

### 1.5.2 System Development Tools

**Hardware tools:**

* Laptop: - to develop our system we use laptop which has the following properties.
* The capacity of RAM is 4GH
* The capacity of hard disk is 400GH
* Has processor Intel® Core TMi3.2.GHZ.
* Flush disk (8GB and above).

**Software tools:**

* Microsoft word 2007: it is software that we use to write our system documentation
* Microsoft PowerPoint 2007: it is software that we use for presentation.
* Edraw: is used for draw UML diagram.
* Web browsers: it is used to display or view or access the website.
* WAMP sever: it is used store information permanently.
* Notepad++: we used this software to edit html, PHP, JavaScript and CSS codes.

**Languages**:

* PHP: Server-side scripting language.
* JavaScript: Client-side scripting language.
* CSS: Style and layout.
* Html: Client side coding.

## 1.6 Significant of the Project

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

* The system eliminates vote repetition, while the vote is taking place.
* The system eradicates the geographical limitation by making the vote is accessible anywhere by using browser.
* The system enables the Administrator to update, delete and edit information about the Voter and Candidate simply.
* Resource and finance expenditure in terms of meeting oriented costs will be saved and the student union can use that for another essential works.
* Error of counting of votes will not occur, because the system assures that counting is done automatically by the system so that there are no voting frauds at all.
* Increases accuracy, availability, quality of the voting process and number of Voters as individual will find it easier and more convenient to vote.
* It provides equally distributed information through online for all Voters and Candidates and also Information is available on time and security for the data is also available.

## 1.7 Beneficiary

**Student/Voter**: this system is used for Voter as they can elect their representative without killing their time or short period of time. So, improve voting service to the Voters through fast, timely and convenient voting.

**Admin:** This system is important for admin can manage users easily.

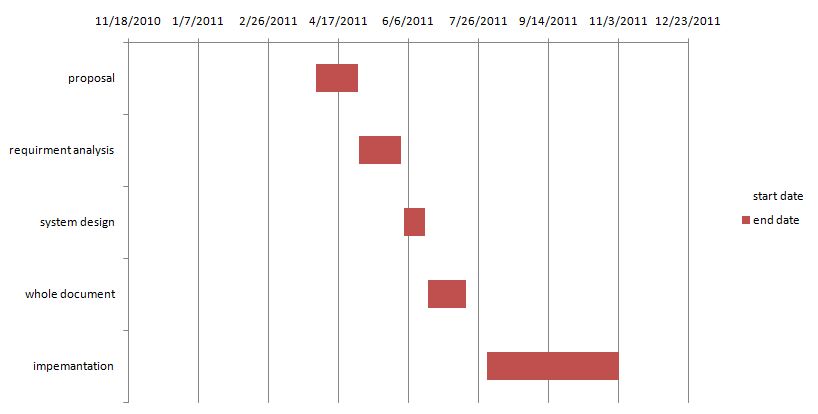
**Candidate and SDD**: they participate in voting in a short period of time.

**University:** this system is for university by reduces the extravagant resource expenditure of the university.

**Developers**: - Developers are beneficiary from the system because we are going to know detail information and understanding about the system .we can also gain knowledge how to develop new system.

**Other people:-**Other people may also benefited by referring the document of our project to retrieve information related to voting.

## 1.8 Time Schedule



**Figure 1: Time Schedule**

# CHAPTER TWO

# 2. REQUIREMENT ANALYSIS

## 2.1 Introduction

Requirement analysis is the initial step of constructing the whole software system. It is a description of the project requirements that we have been gathered and analyzed. Without requirements, the goal is not clear and understandable. In this chapter, requirements of the software system are defined through human natural language and use case. The requirement analysis document is provided in order to ensure that the software we produce will be consistent with the need of the system’s user.

During this phase, the analysis and system modeling must become full aware of functional requirements, nonfunctional requirement of the proposed system, boundaries of the system using use case description, Use-case diagram, showing the behavior of the system and the communication among objects using sequence diagram and describing the structure of the system using class diagram.

The aim of this document is to gather and analyze and give an Interactive system by defining the current system problem statement in detail.

## 2.2 Current System

### 2.2.1 Major Function of Current System Description

The current system of the university of Gondar student union voting system works manually .It registers Candidates and Voters on paper. In each campus first , two students from each class of each department male and female are elected directly by getting high number of vote from their class then those students who are elected join the student union as members of parliament for the student union. Both students who are elected should have more than 2.75 CGPA and should be disciplined in their conduct. And then 15 students are selected from Member of Parliament .After that, those 15 students introduce themselves ,their future plans and sated goal. If the Voter elects them. In each campus 15 students’ puts in order by number of vote and assign 9 students for president, vice president, secretary, discipline issue committee, sport committee, club help committee, gender and HIV committee, finance committee and people communication committee. The remaining will be reserved .Then all campus parliament members are mixed together in order to select the president, vice president, secretary and finance committee of the student union members for the whole campus. From each campus 9 students having higher vote are passed and again introduce themselves, their future plans and sated goal and will have “elect me” session to all campus parliament members for president, vice president, secretary and finance committee. Then the parliament members will choose the Candidate based on Candidates criteria of promotion. The one who gets the higher number of vote will be a president, the second higher number of voice will be vice president and the third higher number of voice will be secretary and the fourth higher number of voice will be finance committee of student union for UOG.

### 2.2.2 Problem of the Current System

Since it is manual based the current system has faced many problems. Among Those problems which includes poor performance, prone to error and poor productivity are listed out. The other problem is that data management related problems this implies the data is not easily accessible, not well organized, not secure from damage and the data is stored redundantly in the form of multiple copies of files. Since it is manual based the system is not efficient, do not give better quality service as user’s needs .Voter loss their time until vote their representative. Increasing the number of Candidate, it takes time and difficult to organize them. Generally the above voting process hierarchy is bulky and do not full fill the students interest.

Performance related problems

* Poor performance like service delay is usual
* The productivity of the voting is not effective
* The work procedure is prone to error

Information related problems.

* Data is not easily accessible.
* Data is not well organized.
* Data is not secure from accident or damage

Economic Related Problem

* Manual handling of data is expensive.
* Cost in terms of time is high.
* In addition that it needs more stationary materials due to data duplication, errors, and storage number of Candidate increase.

Service Related Problem

* The users do not get better service as their need**.**

## 2.3 Requirement Gathering

This section describes the data collection methods that we use to solve the problem. How information is gathered.

### 2.3.1 Requirement Gathering Methodology

#### 2.3.1.1. Observation

We observed that in the student union office and student dean office there are a lot of papers put on the shelf that contains the Candidate information and business rule of voting student union. During voting day we also saw that the student dean office register Candidates on paper by collecting from each class. The Candidate that registers on the paper by student dean director is not come on time. So it takes much time. Count vote manually and count repeatedly. The Candidate introduces themselves individually and takes time. Some Candidate comes and find their name from the paper, the SSD said that your name is not found. This indicates that the system is not reliable.

#### 2.3.1.2. Interviewing

Help us to collect information about the current situation from the user face-to-face by asking questions prepared by the team. The team prepares two types of questions:-

##### 2.3.1.2.1 Open-Ended Questions

The team prepares questions that require explanation or justification. The answers come in the form of a list, a few sentences or something longer such as paragraph or essay.

Generally we are asked the following questions.

Q.1 how many number of student union member are required?

Q.2 what is the criteria of Candidate to selected for student union member?

Q.3 who is the responsibility of collecting the Candidate and how collect them?

Q.4 how the process of voting is takes place?

Q.5 what are the problems during voting student union members?

Q.6 who are the actors participate in the voting process?

##### 2.3.1.2.2 Close-Ended Questions

The team prepares questions which are simple, clear, to the point and can be answered by "yes" or "no" response.

### 2.3.2 Results Found

These Observations had given us an extraordinary data towards developing our problem statement and objectives. As we investigated we saw a lot of files stored on shelf. We also saw every activity is done manually. From the requirement gathering we get what the current system look like, how the current system works, the drawback of current system and things needs to be done for the organization from the fact finding methods .we conclude the current system follow manual system and works are done on a paper form.

Generally the following results are found while gathering requirements regarding the problem of the current student union voting system.

The SSD said that “collecting Candidates is difficult and takes much time and the students complain the process of voting is not fair. Because not all students participate and vote their representatives. The Candidates introduce themselves and give session in front of students that are selected from each class. Those two selected students represent the whole class student but not fulfill the interest of all students. Counting the number of votes is done manually and counting error occurs during counting process. Information is not distributed to all students because the dates of election post by notice board. So, not all students have the habit of visiting notice board. At the end the SSD also said that” we want to improve this system”.

## 2.4 Business Rule

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

BR-1: The Candidate and Voter should be member of university achieved by Candidate id and Voter id.

BR-2: The Candidate and Voter should have only regular student to participate in election.

BR-3: each voter can elect only one candidate.

BR-4: In university of Gondar student union rule the Candidate should have CGPA greater than 2.75 and free from disciple record.

## **2.5 Proposed System**

To reduce problems existing in the current system and make the operation and activities more accurate, the system needs to computerized and should be support in online. The aim of proposed system is to develop improved facilities that can overcome major problems of the existing system. The system provides proper security and reduces a wide range of manual work. This system also saves resources by doing all things used in election system and counts the result for each Candidate correctly and report with exact value electronically.

Our proposed system is better from the existing system because of the following advantages.

* Improved accessibility of user data.
* Reduce data redundancy.
* Maintain quality data.
* Avoid inconsistency.
* Centralized information control.
* Reduce the time and task required to perform the operation within the election area.
* It will provide speed, efficient, Flexibility, reliability, and security for the system users.
* And it improved the moral (motivation) of the users to use the new technology.

### **2.5.1 Overview**

Online student union voting system is design to provide fast and easy way of controlling all activities of voting process. It is also communicate users with each other using webpage, keep the data for longest time in the database.

Our proposed system provides an easy way for users participate in voting process. Administrator sends notification for SSD in order to address students for election. SSD can post the notice for students whose CGPA is 2.75 and more in addition to this they shouldn’t have discipline record. After that, those students can send request to the admin and act as a Candidate. Then the Candidate fills their information including user name, password, promotion or information and send request to admin. The admin can approve the request and create account. The Voter also fills account form such as user name and password and can send their request to admin. But they must be regular student .Then the admin can be approve the request .After that the Voter login within their account and view Candidate with their promotion or information. After that the Voter can vote the Candidate. Nine students that have highest vote for their own campus become President, Vice President, Secretary, Discipline committee, Sport committee, Club Help committee, Gender and HIV committee, Finance and Public Communication committee respectively. The admin can see the result of all campus and select from the result and Approve Candidate for president, vice president, secretary and finance committee for all campus. After that all campus Voters can vote those Candidates for President, Vice President, Secretary and Finance committee.

The major solutions to address the problems of the existing system are as follows. Better utilization of resources, performance, security, reliability, accuracy and in general better service and the new system is aimed to perform basic and crucial tasks of the organization. It contains a well-organized database server which makes data to retrieve, update easily. Since the computer is capable of performing and processing many and huge tasks too faster, efficient and more correct it is preferred to apply it on the system.

Generally, online student union voting system is concerned on providing easy and understandable way of controlling the activities of voting process, and making easy and fast of communication held between admin, Voter, Candidates and Registrar .

### 2.5.2Requirement Specification

Requirement specification for a software system is a complete description of the behavior of a system to be developed.

In requirement specification there are two main ideas defined or explained

* Non-functional (what are system aspects).
* Functional requirement (what things are performed by the system that we developing).

#### **2.5.2.1 Functional Requirements**

The functional requirements of the system describe the necessary functions for which the system is expected to fulfill [1]. 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.

**Administrator**

* The system allows the Administrator to login before accessing system.
* The system allows the Administrator to manage user account.
* The system permits the Administrator to approve account request.
* The system allows the Administrator to send notification for SSD.
* The system permits the Administrator to view Candidate, student and Voter, election result.
* The system enables the Administrator to generate report.
* The system enables the Administrator to view and delete feedback send from Voter and Candidate.

**SSD**

* The system allows SSD to login before accessing system.
* The systems permits the SSD to view notification send from admin.
* The system enables the SSD to send notification for Registrar.
* The system allows the SSD to set student apply date and election date.
* The system allows the SSD to post notice for student in order to participate voting process.
* The system allows the SSD to view the Candidate, the Voter and election result.
* The system allows the SSD to view the report generate by admin.
* The system allows the SSD to change password.

**Registrar**

* The system permits the registrar to login before accessing system.
* The system enables the registrar to view notification.
* The system allows the Registrar to send student data.
* The system permits the registrar to change password.

**Candidate**

* The system allows the Candidate to view notice.
* The system allows the Candidate to send account request.
* The system permits the Candidate to login before accessing system.
* The system enables the Candidate to view election result.
* The system allows the Candidate to give feedback for Admin.
* The system allows the Candidate to change password.
* The system allows the Candidate to view the report generate by Admin.

**Voter**

* The system allows the Voter to view notice.
* The system allows the Voter to send account request.
* The system permits the Voter to login before accessing the system.
* The system permits the Voter to view the Candidate and vote.
* The system allows the Voter to view the result of election.
* The system allows the Voter to give feedback for Admin.
* The system permits the Voter to change password.
* The system allows the Voter to view the report generate by admin.

**Discipline committee**

* The system permits the Discipline committee to login before accessing system.
* The system permits the Discipline committee to send discipline record student.

### **2.5.3 Non-functional Requirement**

In the system which the team develops, these are requirements which are not the functionalities of a system but features, characteristics and constraints of a system.

* **Security:-**Security is how we protect the system from unauthorized access. The system authenticates users by using username and password and also the system is secured by using encryption type md5.
* **Availability: -** The system should be available for access at restriction day and time at election set by SSD.
* **Error handling**: - This system allows preventing or eliminating of error by displaying the message box or the system warns the users who make errors.
* **Usability**: - Usability means the system is user-friendly which is easy to learn, use input data interpret outputs of system or component and operate. The User interface for this system will be simple and clear. This system allows all students to participate in election easily with any place in the university.
* **Performance**:-Performance is the ability of system to response quickly, it is how fast the operation of a system. That means:-how quickly the system reacts to a user input.
* **Portability:**-The system is machine and software system independent. It can run to different target platforms. It not affected by type or required hardware or software.
* **Reliability:-**the system performs its intended functions and operations in a system's environment.

#### 2.3.3.1 User Interface and Human Factors

The system provides web based application user interfaces that are compatible with any Platforms. The knowledge required from the user to access the system is: - computer usage and website interface access. The interface of the proposed system is very easy for users.

#### 2.5.3.2 Documentation

The system has a guide book describes how to use the system for users .We prepare those documents i.e. SPMP, SRS, SDD, STD document with full implementation.

#### 2.5.3.3 Hardware Consideration

The hardware required to use the system are a computer device. The computer device can be a laptop computer or desktop computer or a tablet with capability of connecting to the internet and provide connection to the system.

#### 2.5.3.4 Performance Characteristics

* **Response Time:** The proposed system must have fast response time.
* **Resource Utilization:** The application should utilize minimum amount resource like time, CPU and power needed to perform the task.
* **Concurrent Processing:** Because the proposed system is web based application,it can handle multiple users simultaneously.

#### 2.5.3.5 Error Handling and Extreme Conditions

If the user enters incorrect input like inserting empty string, inserting an incorrect username in to the database or login with unregistered username and password the system should display an appropriate message for each error.

The system should handle exceptions listed below

* Login error (unauthorized login): the system shall handle an attempt to login with incorrect username and password and display appropriate message.
* Submitting Empty String.
* Inserting alphabetic value in integer text field.
* Inserting numeric or integer value in the alphabetic text field.

#### 2.5.3.6 Quality Issues

* **Efficiency**: Since we will use web based system the proposed system is efficient regarding to response time.
* **Reliability:** The performance of the system is better which will increase the reliability of the system.
* **Usability**: The system that we develop should be easy to learn and use.Thisallows 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.

#### 2.5.3.7 System Modifications

As technology is capable of change from time to time there will be future change to the system as a result of new technology invention. Therefore the system can be upgrade to the new technology by the systems developers.

#### 2.5.3.8 Physical Environment

This OSUVS is affected by weather condition when the hardware and software available for our system may be crash by weather condition. Weather conditions like earthquake or the available resources or materials are damaged by fire. And also server or other available software and hardware are affected by virus.

#### 2.5.3.9 Security Issues

Security objectives will be done by using login username and password for each users and members. User information like password should be stored in the database in encrypted form. So it can prevent unauthorized login.

#### 2.5.3.10 Resource Issues

**Server:**

The minimum hardware requirement for MySQL is-

* CPU:32 or 64 bit
* Cores: AMD Dual-core processor C70

**Client:**

* CPU:32 or 64 bit
* RAM: 16GB (or higher is recommended)

## 3.6 System Model

In this particular chapter the team members used an object oriented analysis development methodology. The main activities that are performed in this part includes:-identifying main use cases, scenario, constructing a use case model, and documenting the use case course of events.

### 3.6.1 Scenario

A scenario is “a narrative description of what people do and experience as they try to make use of computer systems and applications”. This describes a particular sequence of activities within a use case.

**Scenario Name: login**

**Participating actor:** Admin/SSD/Candidate/Voter/ Registrar

**Event**: Login

User select Login link from home page and system displays login form. Then user enters username and password .The system checks weather the username and Password is valid. If it is valid login to the system and the system displays all available operations.

If the entered values are incorrect error message is displayed.

**Scenario Name: manage user account**

**Participating actor:** Administrator

**Event** 1: create account

Administrator login to system with password and user name and then select manage account menu. System displays create account, update account, View user account and block account link then select create account link and system displays account create form then Administrator fill the form by user information and then click create link. The system displays successfully create.

**Event** 2: update account

Administrator log in to system with password and user name and then select manage account menu. System displays create account, update account, View user account and block account link. Then select update account link and system displays form that the Administrator search by user name that want to update the user account. The systems display the existing information. The Administrator fills the new information and click update button.

**Event** 3**:** block account

Administrator log in to system with password and user name and then select manage account menu. System displays create account, update account, View user account and block account link .Then select block account link and system displays search form by username. Administrator insert user name that want to block and submit. The system checks the availability of user and displays the search information with block link. The Administrator selects the user that wants to block and select block link. The system displays confirmation message. The Administrator confirms the message. The system displays successfully blocks.

**Event** 4: view user account

Administrator log in to system with password and user name and then select manage account menu. System displays create account, update account, View user account, and block account menu. Then select view users account link and then the system display users.

**Scenario Name: approve account request**

**Participating actor:** Administrator

**Event:** approve account request

Administrator log in to the system with user name and password .Select request for account link. The system displays Candidates well as Voter request. The Administrator approves all requests once or individual request by click approve button.

**Scenario Name: view**

**Participating actor:** Administrator/SSD

**Event1:** view Candidate

Administrator and SSD log in to system with password and user name and then select view menu. System displays View student data, View Candidate, and View Voter link then select View Candidate link system displays Candidate.

**Event2:** view student

Administrator and SSD login to system with password and user name and then select view link system displays View student data, View Candidate, View Voter link then select View student link the system displays student data.

**Event3:** view Voter

Administrator and SSD login to system with password and user name and then select view link system displays View student data, View Candidate, View Voter link then select View Voter link system displays Voter.

**Scenario Name: post notice**

**Participating actor:** SSD

**Even**: post notice

SSD log in to the system with password and user name and then select post notice link. The system display notice form. Then the SSD fills the form and sends notice. The system validates the form. If it is valid, system displays successfully post.

**Scenario Name: send notification**

**Participating actor:** SSD/Administrator

**Event**: send notification

SSD and Administrator log in to the system with password and user name and then select send notification link then the system display notification form. Then the SSD and Administrator fill the form and send notification.

**Scenario Name: view notification**

**Participating actor:** SSD/Registrar

**Event**: view notification

SSD and Registrar login to the system with password and user name and then select send notification link then the system display information.

**Scenario Name: manage feedback**

**Participating actor:** Administrator

**Event1**: view feedback

Administrator login to the system with password and user name and then select mange feedback link .The system display view feedback and delete feedback then select view feedback the system display feedback.

**Event** 2: delete feedback

Administrator login to the system with password and user name and then select mange feedback link .The system display view feedback and delete feedback then select delete feedback link .If there is feedback the Administrator see and delete it.

**Scenario Name: view notice**

**Participating actor:** Voter/Candidate.

**Event**: view notice

Candidate and Voter select notice link and click on home page, then the system displays the content of the notice.

**Scenario name: vote**

**Participating actor:** voter

**Event:** vote

Voters login to the system with username and password. System display voter page. Voters select vote link. Then system display candidate with their own vote button. Then voter select candidate by click vote button. System validates the vote. System displays successfully vote. If the voter selects one candidate more than once, system displays “not allowed more than once”.

**Scenario name: set election date**

**Participating actor:** SSD

**Event:** set election date

SSD login to the system with password and user name and then select set election date link then the system display the form. Then the SSD fill the form and click the set button, then the system validate and display successfully set the time message.

**Scenario name: set student apply date**

**Participating actor:** SSD

**Event:** set student apply date

SSD login to the system with password and user name and then set student apply date link then the system display the form. Then the SSD fill the form and click the set button, then the system validate and display successfully set the time message.

**Scenario Name: Generate Report**

**Participating actor:** Admin

**Event:** Generate report

Admin log in to the system and click on generate report link then system displays the report form Admin writes the report and click submit button. If the values are incorrect system displays please fill with correct value.

**Scenario Name: view election result**

**Participating actor:** admin /candidate/SSD/voter

**Event:** view election result

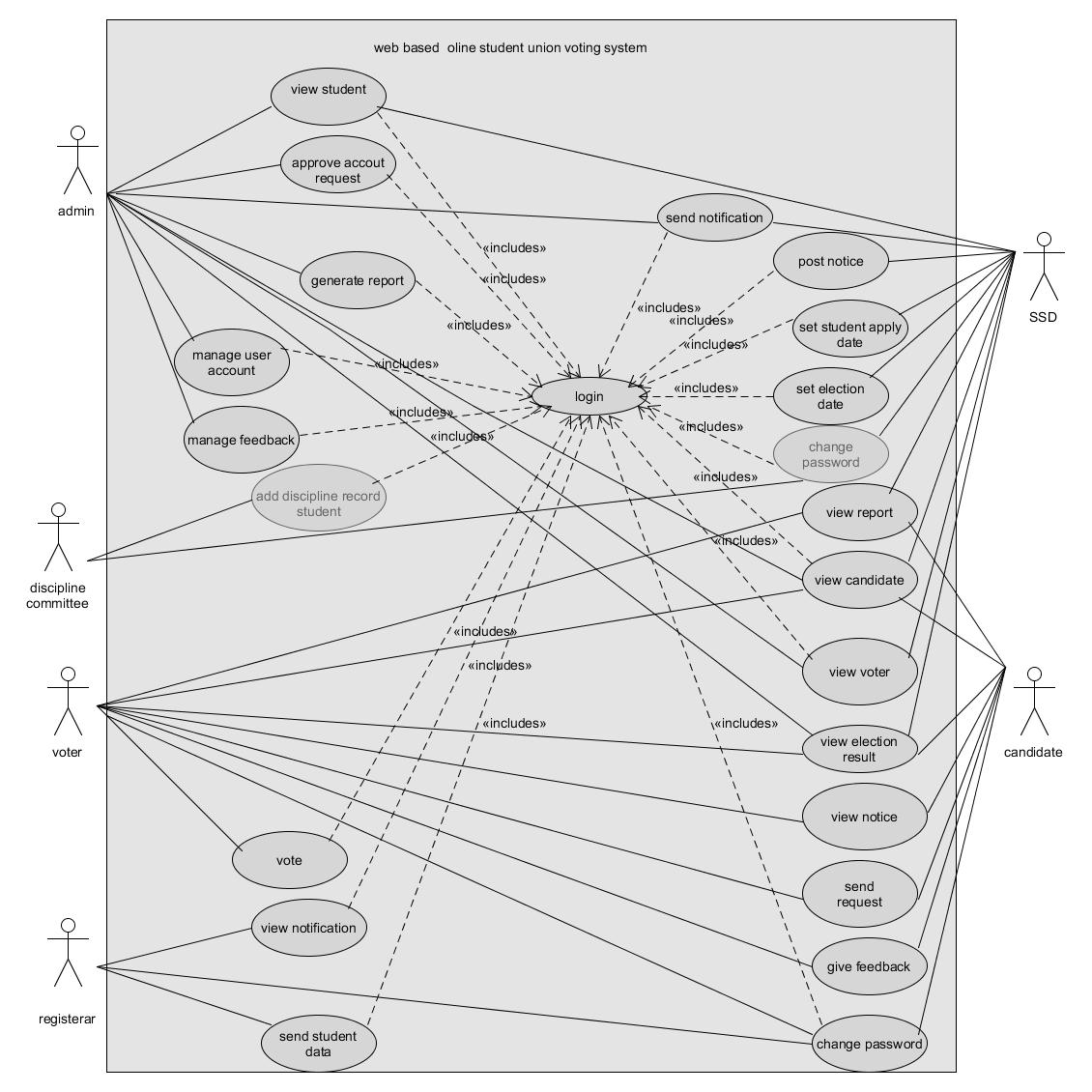
Voter, candidate, Admin and SSD select view election result link on user page then the system display position in option form. They select position from the option. System validates and displays the candidate with their information.

### 2.6.2 Use Case Modeling

In this section we are going to see about use case diagram, description of use case diagram and activity diagram this helps to know the functional interaction between user and the system. Use cases are drawn by examining the actors and defining what the actor will be able to do with the system.

#### 2.6.2.1 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. [3]



**Figure 2**: **Use Case Diagram for Online Student Union Voting System.**

#### 2.6.2.2 Description of Use Case Model

**Table 1: Table of Login**

|  |  |
| --- | --- |
| **Use case name** | **Login** |
| Participate actors | System Administrator, SSD, Registrar, Voter and Candidate |
| Description | System Administrator, SSD, Registrar, Voter and Candidate uses this form to login in the system |
| Pre-condition | System Administrator, SSD, Registrar, Voter and Candidate need to have an account to login. |
| Basic Flow of events | 1. The users enter to the home page. 2. User select login link. 3. System displays login form. 4. Enter user name and password and click login button. 5. The system checks for the validity of the user name and password. 6. The system displays the user page if user name and password is valid.[A] 7. Use case ends. |
| Alternative course of action | A: If the insert user name and password is not valid.  A1: system displays “Please Enter valid user name and password” and back to step3. |
| Post condition | The users (Voter, admin, Candidate Registrar and SSD) logged in to the main page. |

**Table 2: Table of Send Account Request for Candidate**

|  |  |
| --- | --- |
| **Use case name** | **Send account request** |
| Actor | Candidate |
| Description | These use case permits the Candidate create account and send to admin in order to be elected. |
| Pre-condition | Time must be run over the time limit and must be member UOG. |
| Basic Flow of events: | 1. This use case is initiated when the Candidate clicks on Candidate request link in home page. 2. System displays display search form that Candidate searches their information by their id. 3. Candidate enters his or her id and click search. 4. If the insert id valid, System display full information and form for user name, password, photo, promotion, and position.[A] 5. Candidate fill form and click send button. 6. System validates the form. 7. If it is valid , System displays request send successfully.[B] 8. Use case Exit |
| Alternative course of action | A: If the entered id is invalid  A1: the system displays “you are not member of UOG” and “please enter valid id”. Go to flow 2.  B: if form is not valid.  B1:System display” please insert valid form” go to flow 4 |
| Post condition | The Candidate request is successfully apply. |

**Table 3: Table Send Account Request for Voter**

|  |  |
| --- | --- |
| **Use case name** | **Send account request** |
| Actor | Voter |
| Description | These use case permits the Voter create account and send to admin in order to participating voting process. |
| Pre-condition | Time must be run over the limit and must be member UOG. |
| Basic Flow of events | 1. This use case is initiated when the Voter clicks on Candidate request link in home page. 2. System displays display search form that Voter searches their information by their id. 3. Voter enters his or her id and click search. 4. If the insert id valid, System display full information and form for user name, password and photo.[A] 5. Voter fill form and click send button. 6. System validates the form. 7. If it is valid , System displays request send successfully.[B] 8. Use case Exit |
| Alternative course of action | A: If the entered id is invalid  A1: the system displays “you are not member of UOG” and “please enter valid id”. Go to flow 2.  B: if form is not valid.  B1:System display” please insert valid form” go to flow 4 |
| Post condition | B The Voter request is successfully apply. |

**Table 4: Table of View Election Result**

|  |  |
| --- | --- |
| **Use case name** | **View election result** |
| Actor | Voter, Candidate, SSD |
| Description | This describes the process of how the user views the election results by using the system. |
| Basic course action | 1. The user open home page. 2. The user select view election result link. 3. System display search by position form. 4. User select position option. 5. System validates the position. 6. If valid position, the system displays the election result.[A] 7. The users view the necessary information. 8. Use case ends. |
|  | A: If invalid position, system display “select position “go to flow3. |
| Post condition | The user knows the wanted information. |

**Table 5: Table of Create Account**

|  |  |
| --- | --- |
| **Use case name** | **Create account** |
| Actor | Administrator |
| Description | This use case is initiated by the System Administrator. It provides to Create Account. |
| Pre-condition | System Administrator must login in to the system. |
| Basic course action | 1. The Administrator select manage user account menu. 2. System displays create account, view account, update account and block account submenu. 3. The Administrator selects Create account submenu. 4. System displays create new account form. 5. The Administrator inserts fill user name, password and photo for users. 6. Click of create account button. 7. System Validate data entry. 8. New account created message displayed to the Administrator.[A] 9. Use case ends |
| Alternative course of action | A: If the data entry not valid.  A1: system displays invalid Input message. Go to flow 4  A2: If user has been already create account, user create account message displayed. |
| Post condition | The System Administrator created account so users can login the system. |

**Table 6: Table of View Candidate**

|  |  |
| --- | --- |
| **Use case name** | View Candidate |
| Actor | Administrator, SSD, Voter |
| Description | This describes the process of how users view Candidate information by using the system. |
| Pre-condition | Administrator, SSD and Voter have account if they want to see Candidate. |
| Basic course action | 1. This use case is initiated when the Voter, admin and SSD click view Candidate link. 2. The system display available name of Candidate with their   Some information.[A]   1. User view Candidate. 2. Use case ends. |
| Alternative course of action | A: If the users cannot see the Candidates.  A1:system display “try again “go to flow 2 |
| Post condition | The user can see Candidate information. |

**Table 7: Table of Generate Report**

|  |  |
| --- | --- |
| **Use case name** | **Generate report** |
| Actor | Admin |
| Description | Admin prepare a report about the Voters and Candidates. |
| Precondition | Admin must be log in to the system |
| Main course of action | 1. System display menu 2. 2.Admin clicks on Generate report 3. System displays form for generate report. 4. 4.Admin fill report 5. 5. Admin clicks on save button. 6. Systems validate the user input and unfilled input. 7. System display report generate successfully[A] 8. Use case Exit.   5. |
| Post condition | Generate report successfully |
| Alternative flow of event | A:If the fill form is invalid  A1:system display “please enter valid form “go to 3 |

**Table 8: Table of Give Feedback**

|  |  |
| --- | --- |
| **Use case name** | **Give feedback** |
| Actor | Voter, Candidate |
| Description | To enable the users to send the feedback, to comment any Suggestion on text area. |
| Precondition | Voter and Candidate must be having a suggestion about the system  as well as voting process. |
| Flow of event | 1. The Voter and Candidate click feedback button.  2. System display form.  3. The Voter and Candidate write some necessary information and click send button.  4. The system can validate.  5. System display feedback successfully sends.  6. Use case Exit |
| Post condition | Suggestion will be send. |

**Table 9: Table of Approve Account Request**

|  |  |
| --- | --- |
| **Use case name** | **Approve account request** |
| Actor | Admin |
| Description | This use case permits admin create account for Voter and Candidate. |
| Pre-condition | The admin should be login to the system. |
| Basic Flow of events: | 1. The admin click approve account request link in the admin page. 2. System displays the request. 3. The admin click approve button. 4. System displays approve successfully.[A] 5. Use case Exit |
| Alternative course of action | A:if there is no request  A1: system display “there is no request” and go to flow 1. |
| Post condition | The Voter and Candidate is register and create account successfully. |

**Table 10: Table of Set Election Date**

|  |  |
| --- | --- |
| **Use case name** | **Set election date** |
| Actor | SSD |
| Description | This use case permits SSD can restrict the election date |
| Pre-condition | The SSD should be login to the system. |
| Basic Flow of events: | 1. The SSD click set election date link in the SSD page. 2. System displays the form. 3. The SSD fill the form and click the set button. 4. System validates the form. 5. System displays time is set successfully.[A] 6. Use case Exit |
| Alternative course of action | A:if the enter value is not valid  A1: system display “please enter valid value” and go to flow 2. |
| Post condition | The election date successfully set. |

**Table 11: Table of Set Student Apply Date**

|  |  |
| --- | --- |
| **Use case name** | **Set student apply date** |
| Actor | SSD |
| Description | This use case permits SSD can restrict the student send request date, |
| Pre-condition | The SSD should be login to the system. |
| Basic Flow of events: | 1. The SSD click set student apply date link in the SSD page. 2. System displays the form. 3. The SSD fill the form and click the set button. 4. System validates the form. 5. System displays time is set successfully.[A] 6. Use case Exit |
| Alternative course of action | A:if the enter value is not valid  A1: system display “please enter valid value” and go to flow 2. |
| Post condition | The students apply date successfully set. |

**Table 12: Table of Send Notification**

|  |  |
| --- | --- |
| **Use case name** | **Send notification** |
| Actor | SSD, Admin |
| Description | This use case permit SSD and Admin want to send message. |
| Pre-condition | The SDD and Admin should be login to the system. |
| Basic Flow of events | 1. The SSD and Admin click send notification link in the user page. 2. System displays the form. 3. SSD and admin fill the form and click the send button. 4. System validates the form. 5. System displays message is send successfully.[A] 6. Use case Exit |
| Alternative course of action | A:if the enter value is not valid  A1: system display “please enter valid value” and go to flow 2. |
| Post condition | The students apply date successfully set. |

**Table 13: Table of Post Notice**

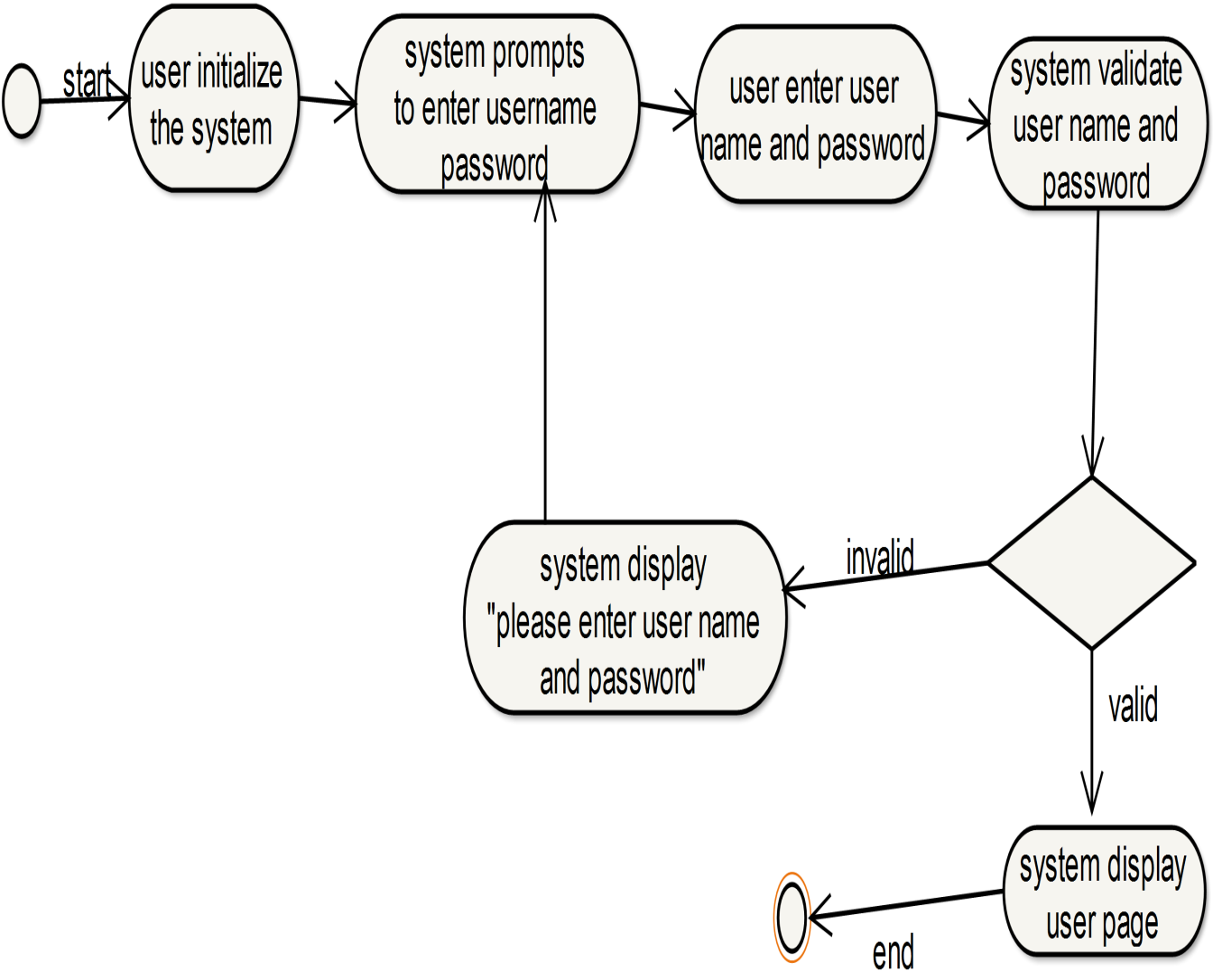
|  |  |
| --- | --- |
| **Use case name** | **Post notice** |
| Actor | SSD |
| Description | This use case permit SSD post notice for to announce information for student. |
| Pre-condition | The SDD and Admin should be login to the system. |
| Basic Flow of events | 1. The SSD click post notice link in the SSD page. 2. System displays the form. 3. SSD fill the form and click the send button. 4. System validates the form. 5. System displays message is post successfully.[A] 6. Use case Exit |
| Alternative course of action | A:if the enter value is not valid  A1: system display “please enter valid value” and go to flow 2. |
| Post condition | The notice post successfully. |

**Table 14: Table of Vote**

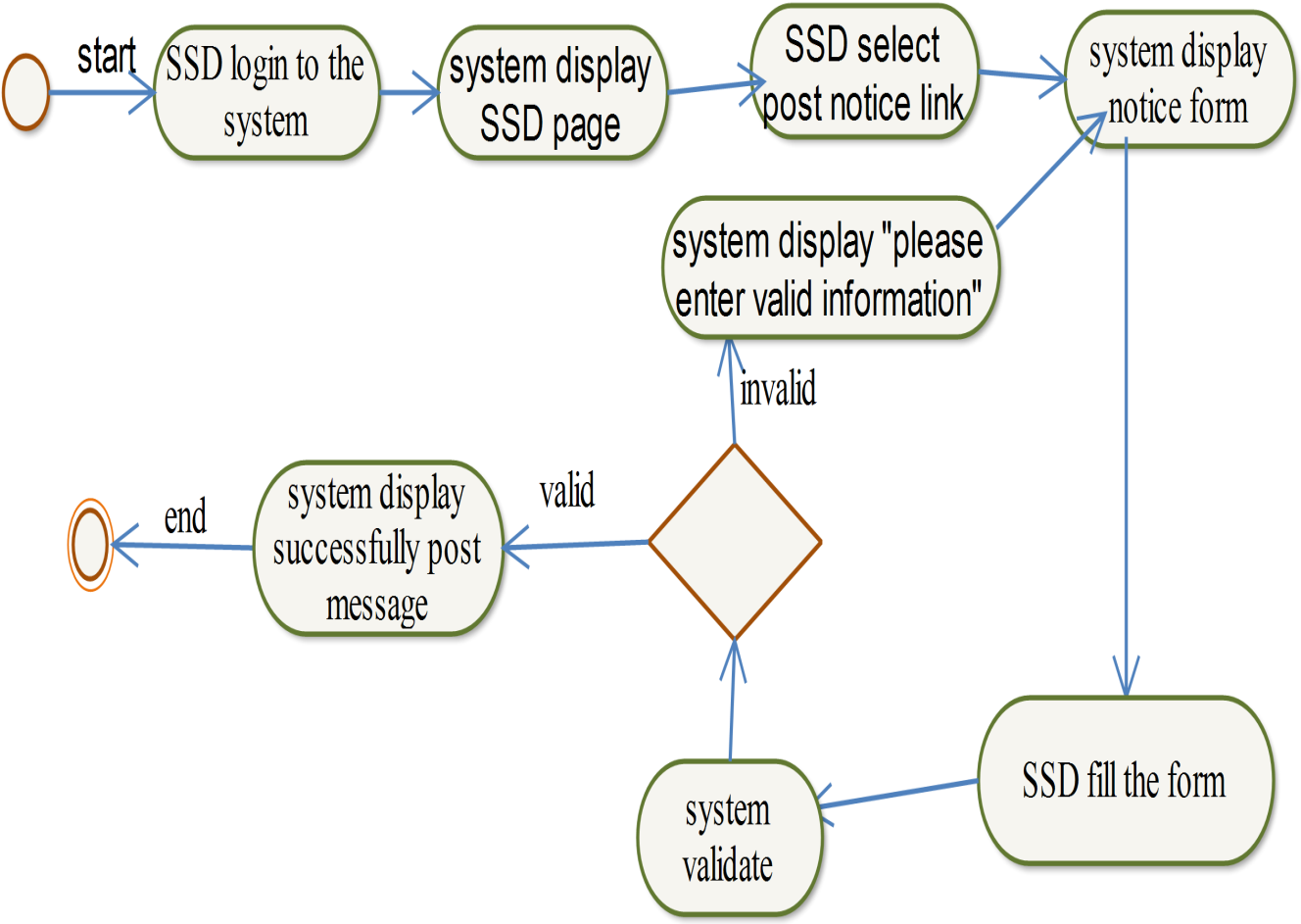
|  |  |
| --- | --- |
| **Use case name** | **Vote** |
| Actor | Voter |
| Description | This use case permit the voter vote their representative |
| Pre-condition | The voter should be login to the system. |
| Basic Flow of events: | 1. The voter select vote link in the voter page. 2. System display candidate with their own vote button. 3. Voter select candidate by click vote button. 4. System validate vote. 5. If not vote before, System displays message is vote successfully.[A] 6. Use case Exit |
| Alternative course of action | A: if the vote is more than once for one candidate.  A1: system display “you already vote before” and go to flow 2. |
| Post condition | The voter vote successfully. |

### 2.6.3 Activity Diagram

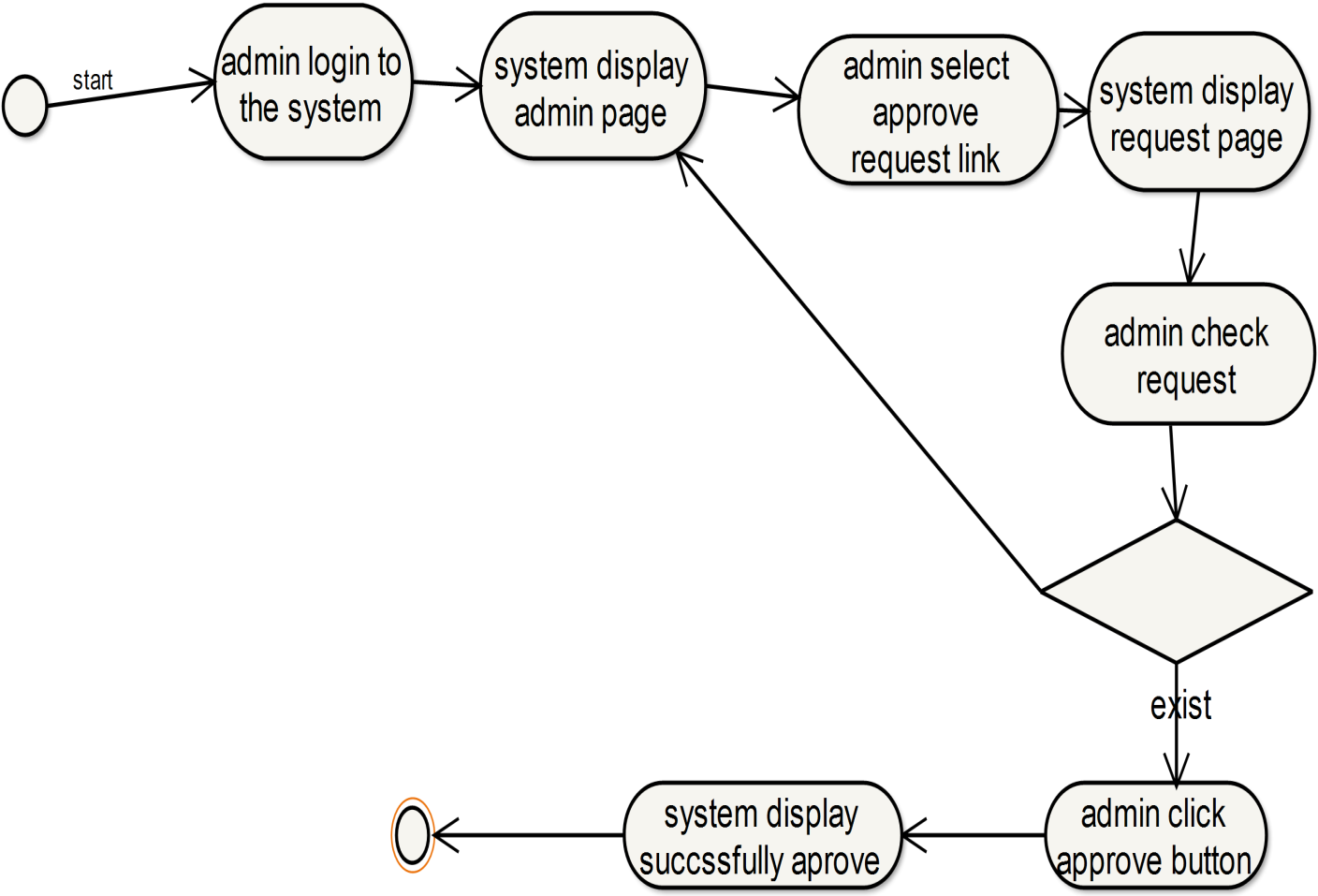
Activity diagrams are used to document the logic of a single operation/method, a single use case, or the flow of a business process. Activity diagrams essentially a flowchart showing flow of control from activity to activity. It includes modeling the sequential process. It also includes modeling the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



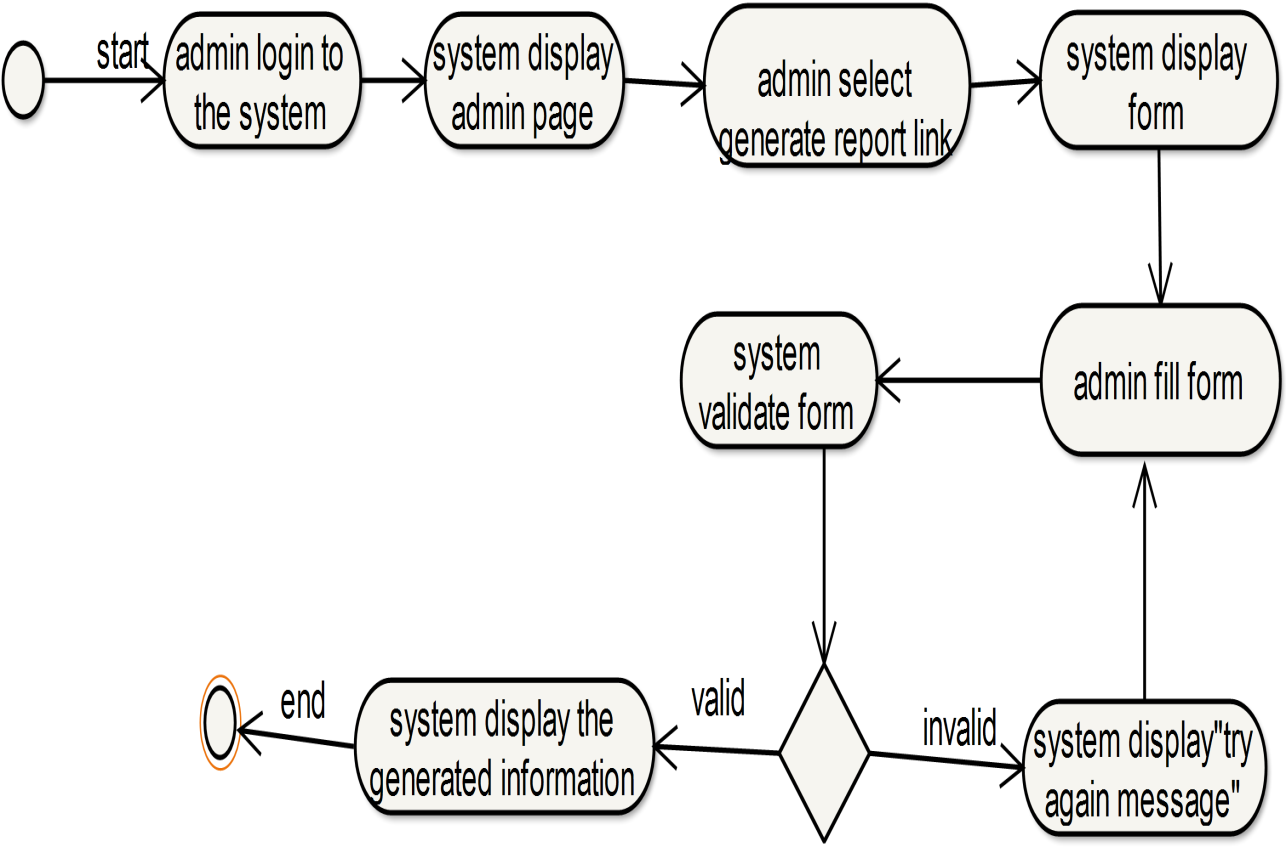
**Figure 3: Activity Diagram for Login**



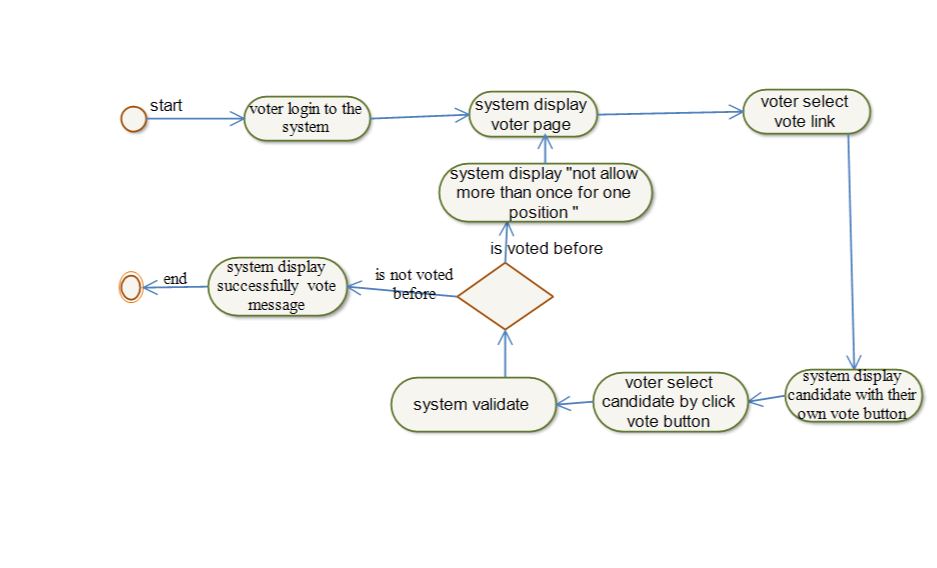
**Figure 4: Activity Diagram for Post Notice**



**Figure 5: Activity Diagram for Approve Request**



**Figure 6: Activity Diagram for Generate Report**



**Figure 7: Activity Diagram for Vote**

### 2.6.4 Object Model

#### 2.6.4.1 Data Dictionary

|  |  |  |
| --- | --- | --- |
| Object | Attribute | Description |
| Admin  SSD  Registrar  Discipline committee | First Name | This describe first name of admin, SSD Registrar and discipline  Committee |
| Last Name | This describe the last name of admin, SSD, Registrar and  discipline committee |
| User name | This describes the user name of the authenticate user |
| Password | This describes the password of the authenticate user |
| Age | Describes the age of an admin , SSD, Registrar and discipline committee |
| Sex | is describes the state of being male or female |
| Role | is describes the type of user |
| Id | This describe the identification number of admin, SSD, Registrar and  Discipline committee |
| Discipline record student | Id | This describe the identification number of Discipline record student |
|  | First Name | This describe first name of Discipline record student |
| Last Name | This describe the last name of Discipline record student |
| Age | Describes the age of Discipline record student |
| Sex | This describes the state of being male or female |
| Faculty | This describe the faculty of Discipline record student |
| Program | This describe the faculty of Discipline record student |
| Discipline type | This describes the discipline of discipline record student |
| Student, Voter | La sid and vid | This describe the identification number of student and voter |
| Us First name | This describe first name of student and voter |
| Last name | This describe first name of student and voter |
| Sex | This describe the state of being male or female and voter |
| P Age | This describes the age of student and voter and voter |
| Campus | Describes the campus of student and voter |
| Faculty | is describes the Faculty of student and voter |
| Department | is describes the department of student |
| Year | This describe the year of student |
| CGPA | These describe the CGPA of student |
| Program | Describe the program of student that follows either regular or extension. |
| Candidate | Cid | Unique identifier for Candidate and Voter |
| First name | This describe first name of Candidate and Voter |
| Last name | This describe last name of Candidate and Voter |
| Sex | is describes the state of being male or female |
| Age | Describe the age of Candidate and Voter |
| User name | This describes the user name of the authenticate user |
| Password | This describes the password of the authenticate user |
| Campus | Describe the campus of Voter and Candidate |
| Faculty | Describe the Faculty of Voter and Candidate |
| Department | Describe the department of Voter and Candidate |
| Year | Describe the year of Voter and Candidate |
| CGPA | Describe the CGPA of Voter and Candidate |
| Program | Describe the program follow Voter and Candidate |
| Promotion | Describe the promotion of Candidate |
| Photo | Describe the image of Voter and Candidate |
| Notice | Posted date | Describe the date of post notice |
| Ex-date | Describe the expire date of notice |
| Title | The title of notice |
| Content | The content of notice |
| Sender | The sender of notice |
| Election | Startdate | Start date of election |
| Closedate | Close date of election |
| Report | Date | Describes the date on which the report is being generated |
| Content | Describe the written report about specific issue |
| Reporter | This describe the information of the person who is being generate the report |
| Request | Request type | This describes who is to identify the requester. |
|  | Date | This describe the date that request is send. |
| Student apply date | Startdate | The start date of account request or registration |
| End Date | The end date of account request or registration |
| Count | cid | This describe the identification number of candidate |
|  | Total vote | The total vote of candidate |
| Ballot store | Vid | This is the voter id. |
|  | cid | This the candidate id, |

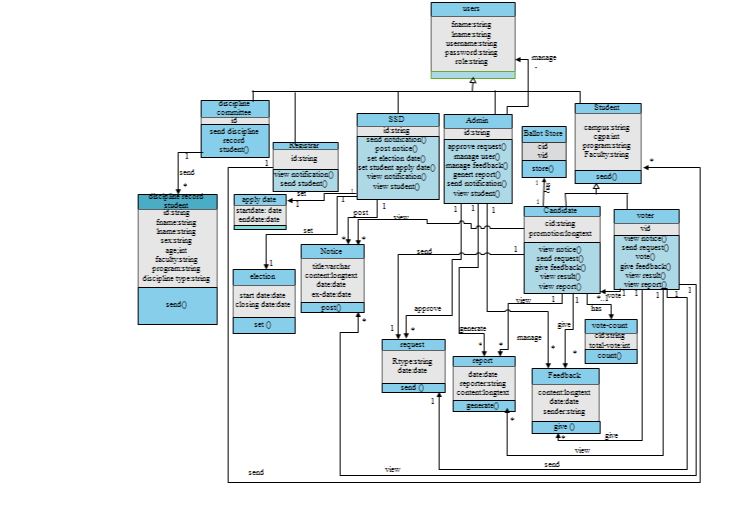
**Table 15: Table of Data Dictionaries**

#### 2.6.4.2 Class Modeling

##### 2.6.4.2.1 Class Diagram

Class diagram is one of the most useful types of diagram in UML as the clearly map out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects.

UML class diagrams show the classes of the system, their interrelationships (including inheritance, aggregation, and association) and the operations and attributes of the classes. Class diagrams are used for a wide variety of purposes, including both conceptual/domain modeling and detailed design modeling. A class model is comprised of one or more class diagrams and the supporting specifications that describe model elements including classes, relationships between classes, and interfaces. [UML class diagrams](http://agilemodeling.com/artifacts/classDiagram.htm) show the classes of the system, their inter-relationships, and the operations and attributes of the classes.

1

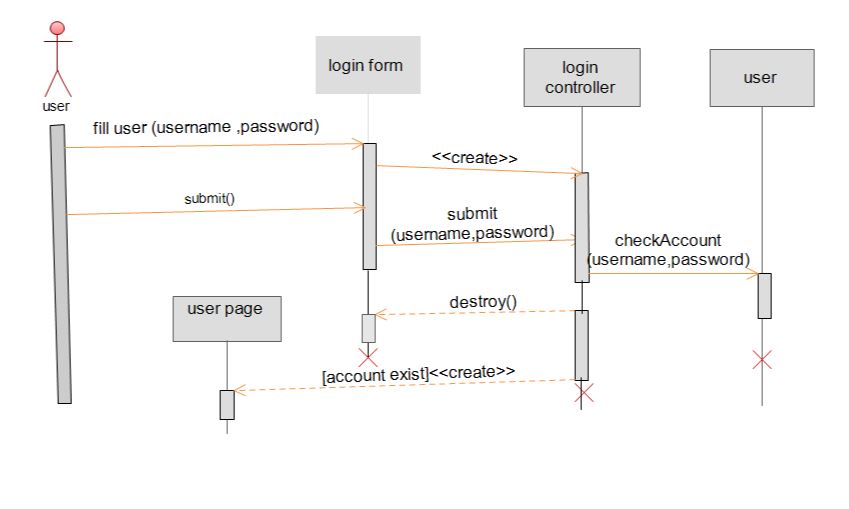
**Figure 8: Analysis Class Diagram**

#### 2.6.4.3 Dynamic Modeling

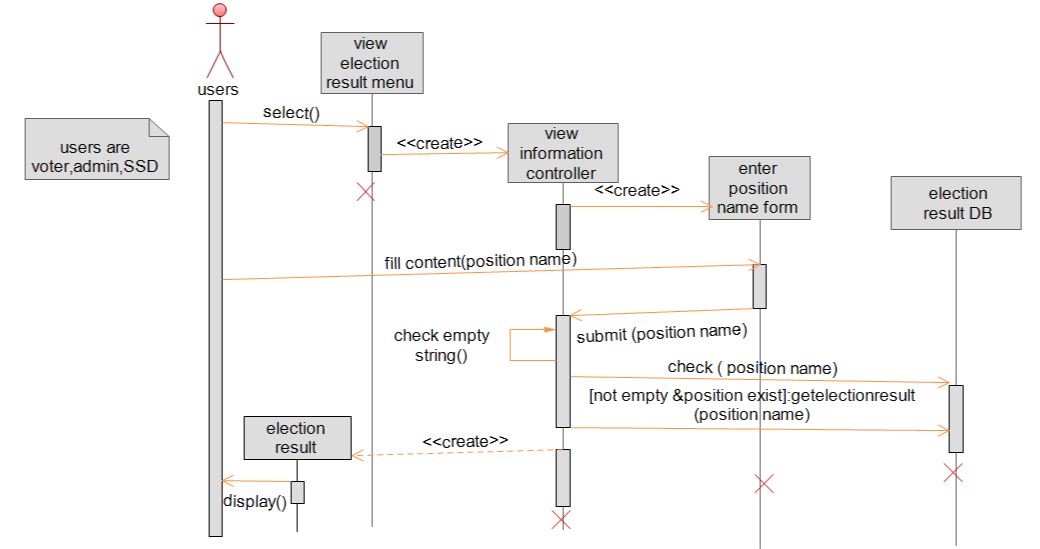
We have identified a number of entity, boundary and control objects, along the way we have also identified some of their attributes and associations, we represent this objects in a sequence diagrams, depicting the interaction that occur during the use case to identify additional associations and attributes.

##### 2.6.4.3.1 Sequence Diagram

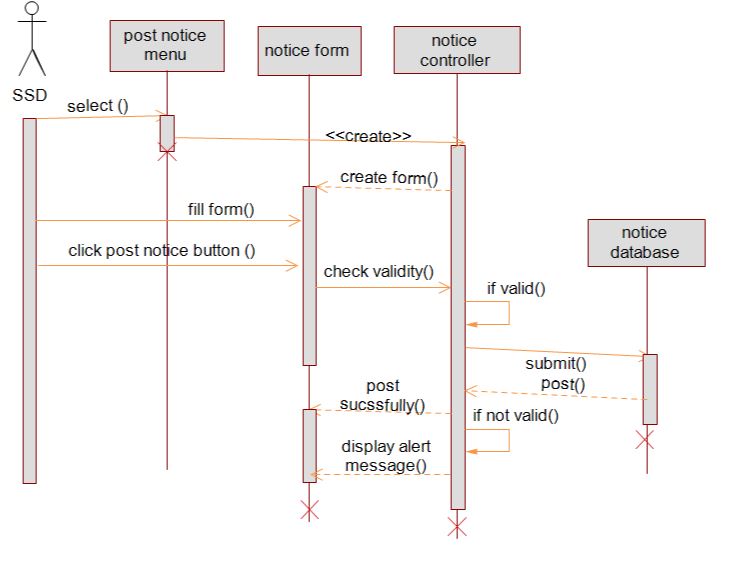
UML Sequence diagram showing the sequence of interactions among objects and used to represent or model the flow of messages, events and actions between the objects or components of a system. Sequence diagrams are also used primarily to design, document and validate the architecture and interfaces of the system by describing the sequence of actions that need to be performed to complete a task or scenario. [1]



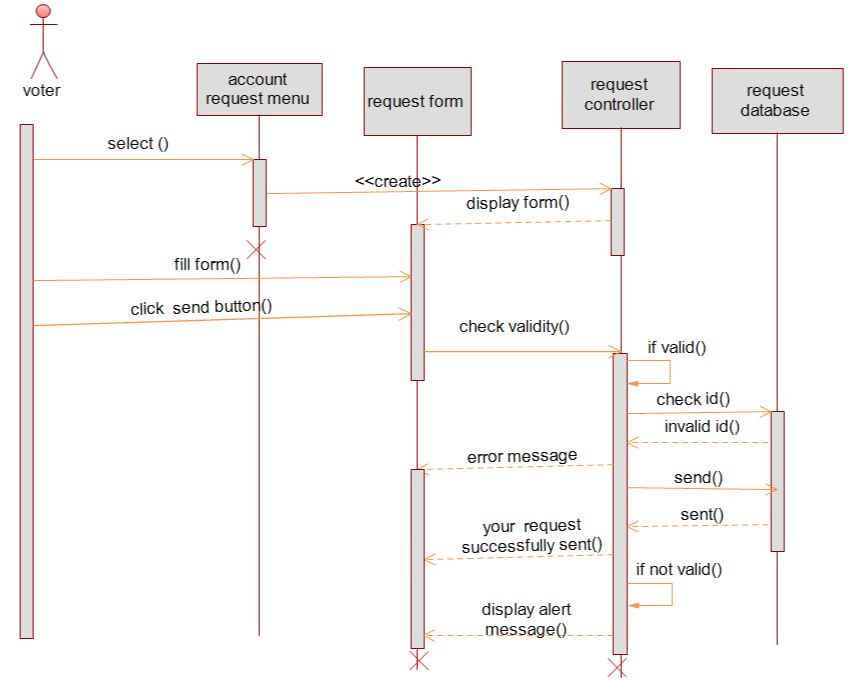
**Figure 9: Login Sequence Diagram**



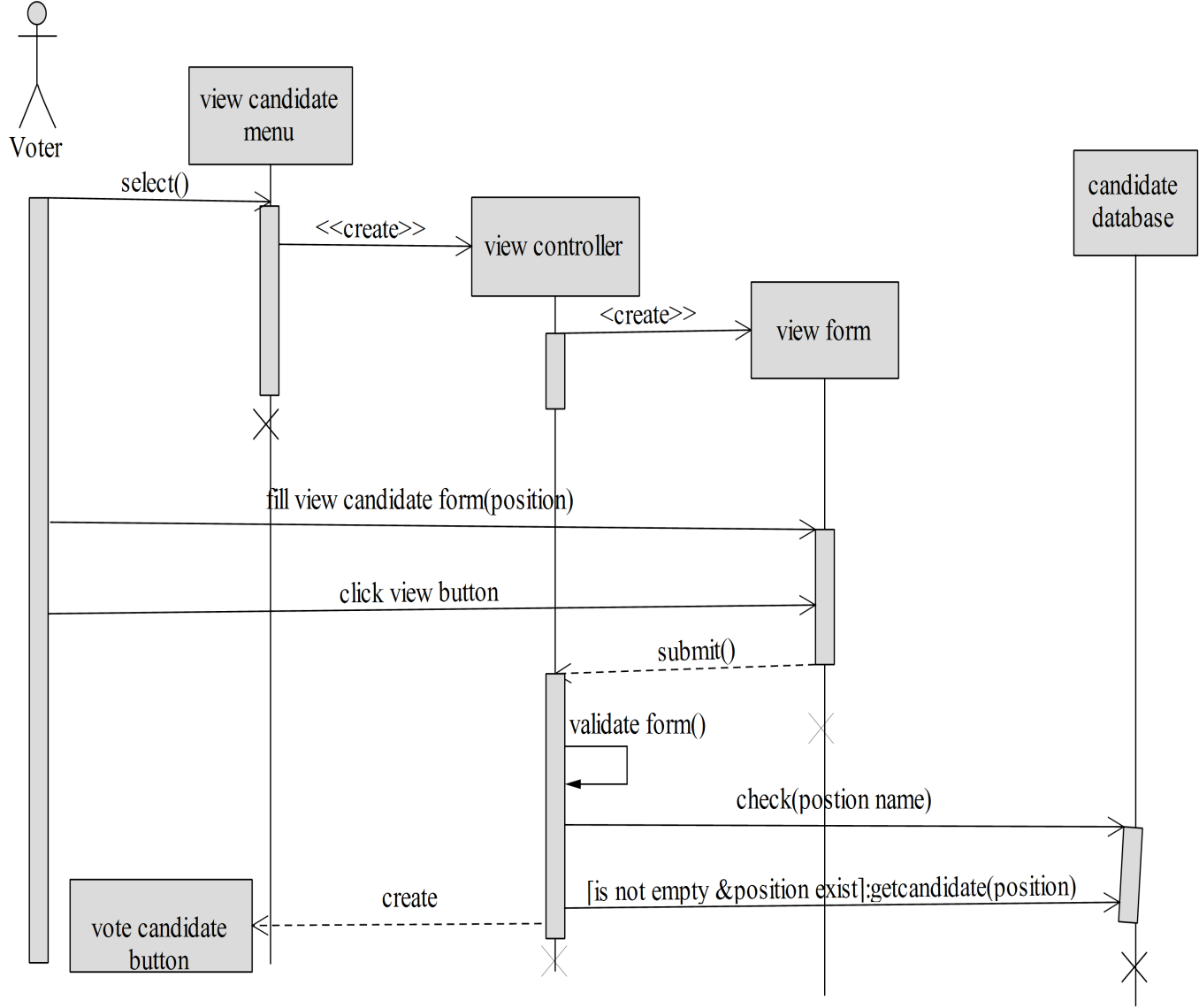
**Figure 10: Sequence Diagram for View Election Result.**



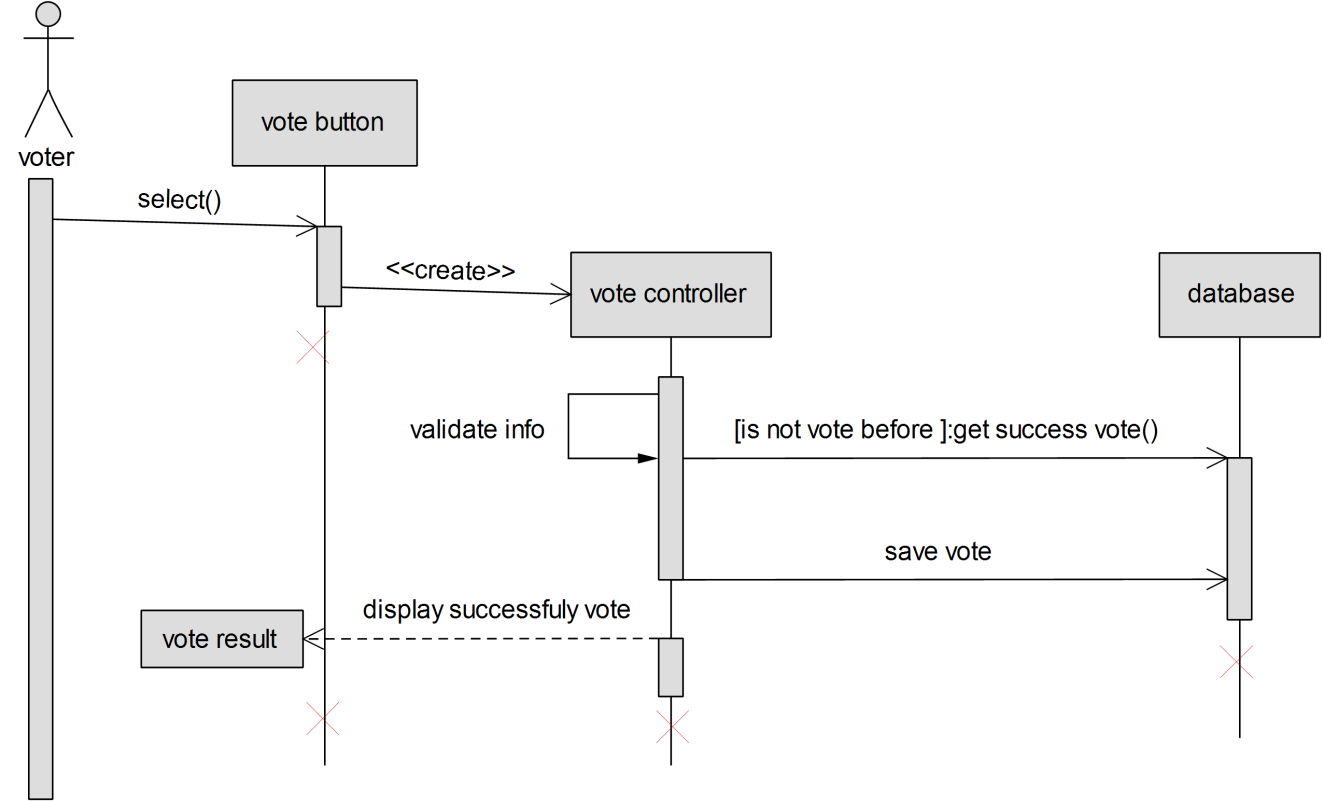
**Figure 11: Sequence Diagram for Post Notice**



**Figure 12: Account Request Sequence**



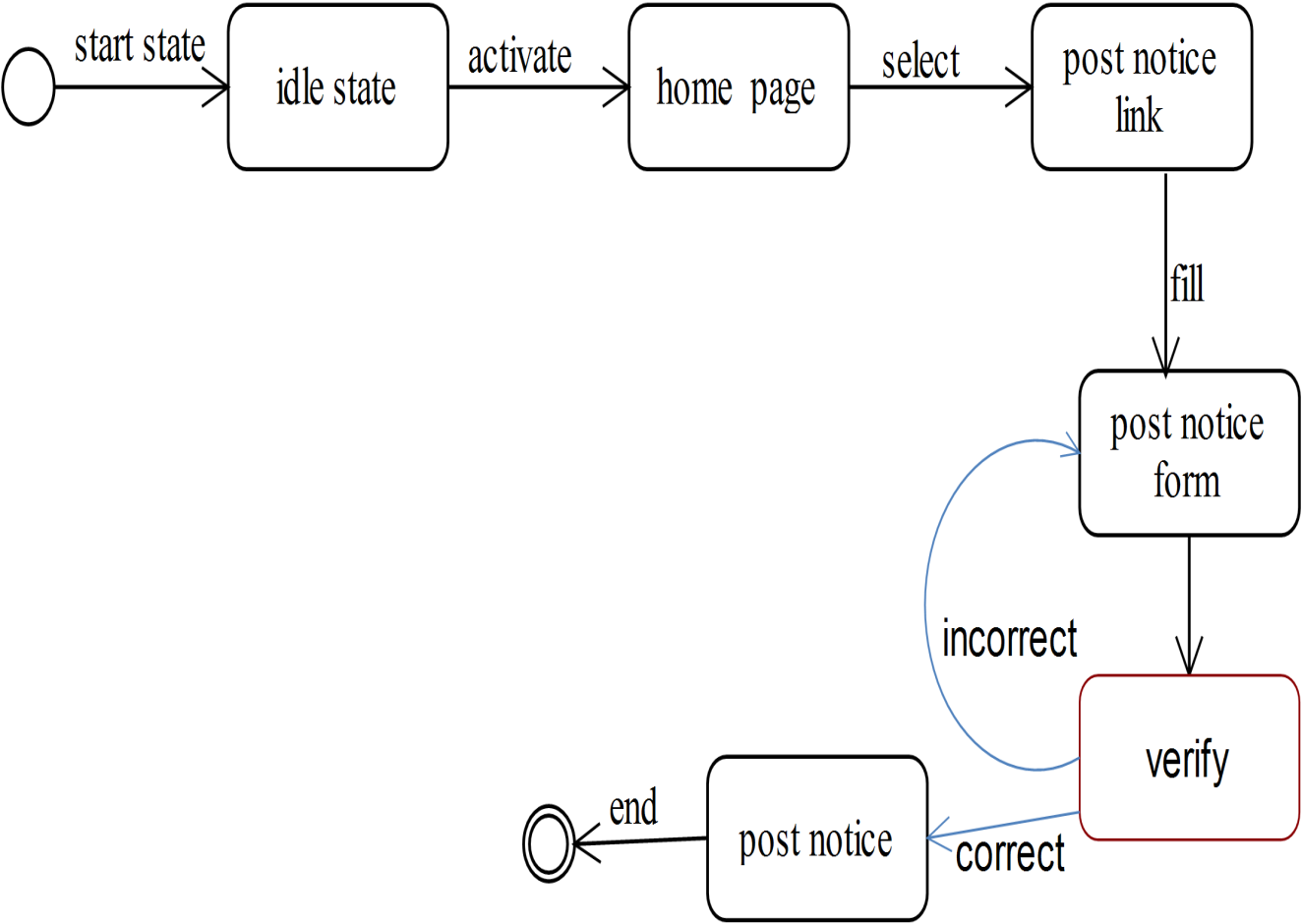
**Figure 13: Sequence Diagram for View Candidate**



**Figure 14: Sequence Diagram for Vote**

##### 2.6.4.3.2 State Chart Diagram

This section highlights the sequence of states that an object goes through the events that cause the transition from one state to the other and the actions that result from a state change.

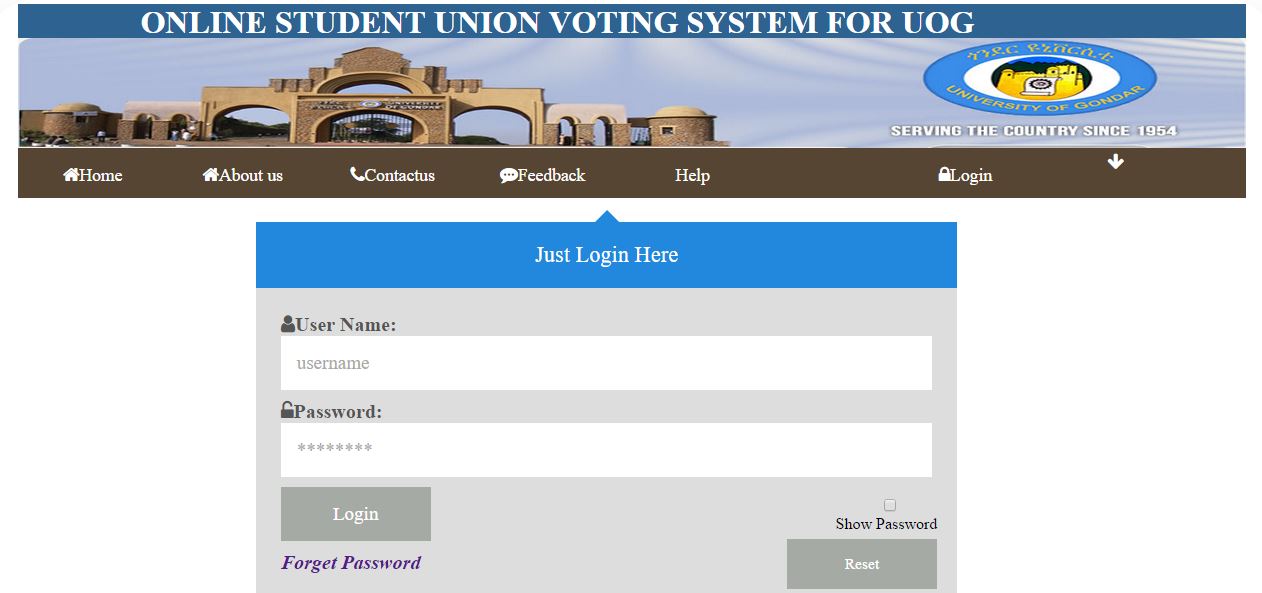


**Figure 15: State Chart Diagram for Post Notice**

## 2.7 User Interface

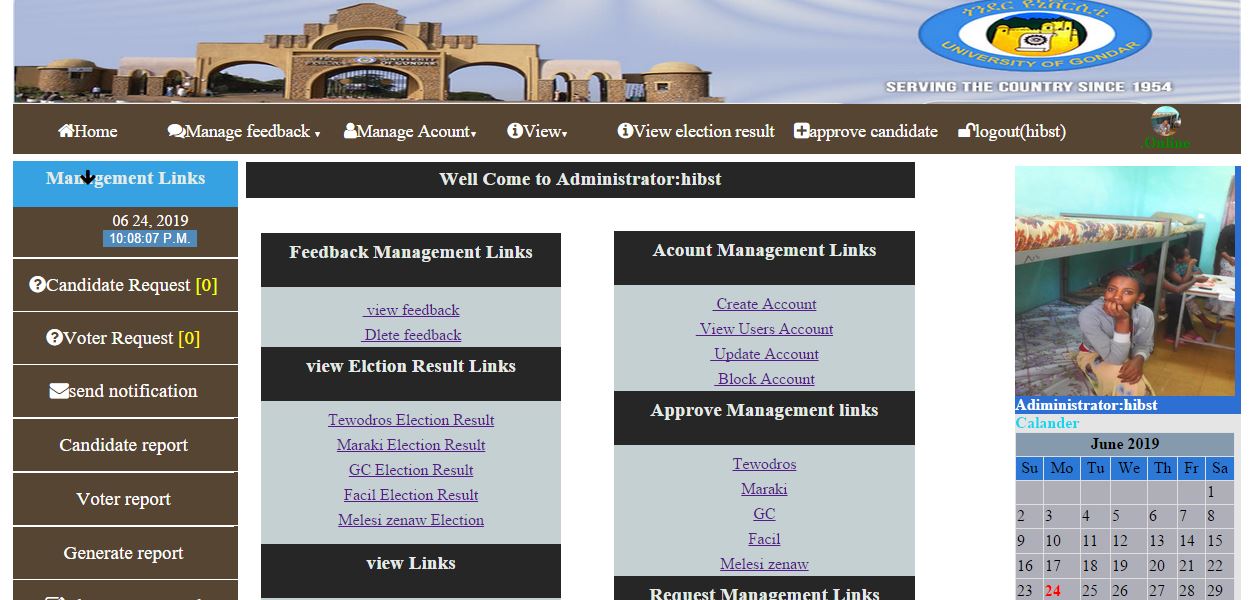
User interface is the front-end application view to which user interacts in order to use the system. User interface is designed such a way that it is expected to provide the user insight of the system.

**Login page**: this page appears as user has signed up .it gives a choice for the role of the user to enter and also the user must enter a password in order to access the next pages of the website .it gives a chance for the user to reset the password if the password is not remembered by him/her.



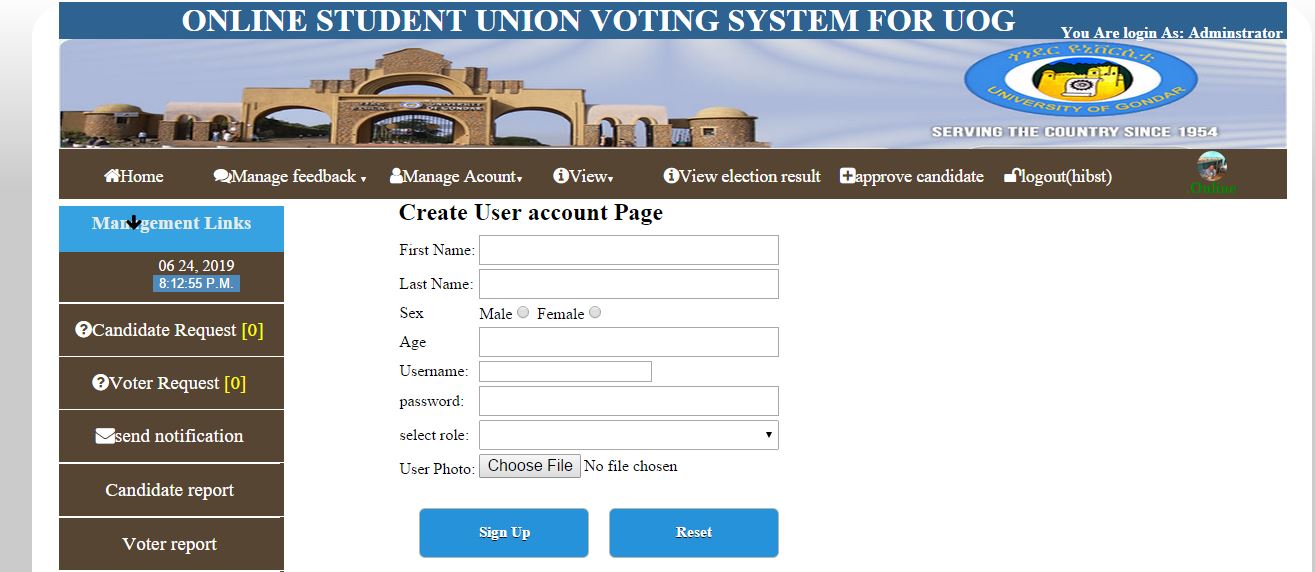
**Figure 16: User Interface for Login Page**

**Admin page:** this page appears after admin login with username and password to do any operation.



**Figure17: User Interface for Admin Page**

**Create Account Page**: this page appears when Administrator login to the system .it has a form that an Administrator fills to create an account for user. Admin must fill username, password, and role.



**Figure 18: User Interface for Create Account Page**

# CHAPTER THREE

# 3. SYSTEM DESIGN

## 3.1 Introduction

In the previous chapter we have discuss the functional and non-functional requirements of the system and produced the analysis model. 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. We provide overview of the current system and the proposed system software architecture and we specify design goals to improve the quality of the system by reducing response time, by increasing the availability and by improving its reliability to function correctly under erroneous conditions. Software architecture is the process of defining a structured solution that meets all of the technical and operational requirements, while optimizing common quality attributes such as performance, security, and manageability. We describe the Hardware/software mapping with deployment diagram, subsystem decomposition, access control and security. Hardware/software mapping is mapping subsystems to processors and components enable us to identify potential concurrency among subsystems and to address performance and reliability goals. We specify the boundary conditions for each subsystem by mentioning how the subsystems initialized and terminates. This document is traceable with the previous documents (Software Project Management Plan and Software Requirement Specification) i.e. sub systems are from our Software Requirement specification previous document and the design goals. We describe each sub topics in detail in the following sub sections.

## 3.2 Design Goal

Design goals describe the important system qualities. Design goals also define the values against which options are evaluated. When designing a new system, a system designer creates a model of the system from requirements made.

**Design Goals:-**

* **Response time:** taking less time for system to respond to the request.
* **Simplicity:** we develop the new system can make user friendly graphical user interface.
* **Reliability:** the system must perform its intended functions and operations in a system's environment. Without experiencing failure or system crash.
* **Security:** the system must be protected from an authorized access, threats, attacks and vulnerabilities.
* **Fault tolerance:** the system must have the ability to satisfy requirements despite failures such as hardware, software or network failures. Determines the quality of the product and future maintainability.
* **Cost:** The system should be developed with minimum cost possible.
* **End Criteria: -** The system should have simple and understandable graphical interface. All the interfaces, forms and buttons are written or designed in a simple language or common language so that everyone can access it without any difficult.

## 3.3 Current Software Architecture

Since it is manual system there is no best current software architecture online student union voting system.

## 3.4 Proposed Software Architecture

Proposed software architecture is the architecture style of future university of Gondar online student union voting management system. The clear description of the software architecture is describing on the next sub topic overview.

### 3.4.1 Overview

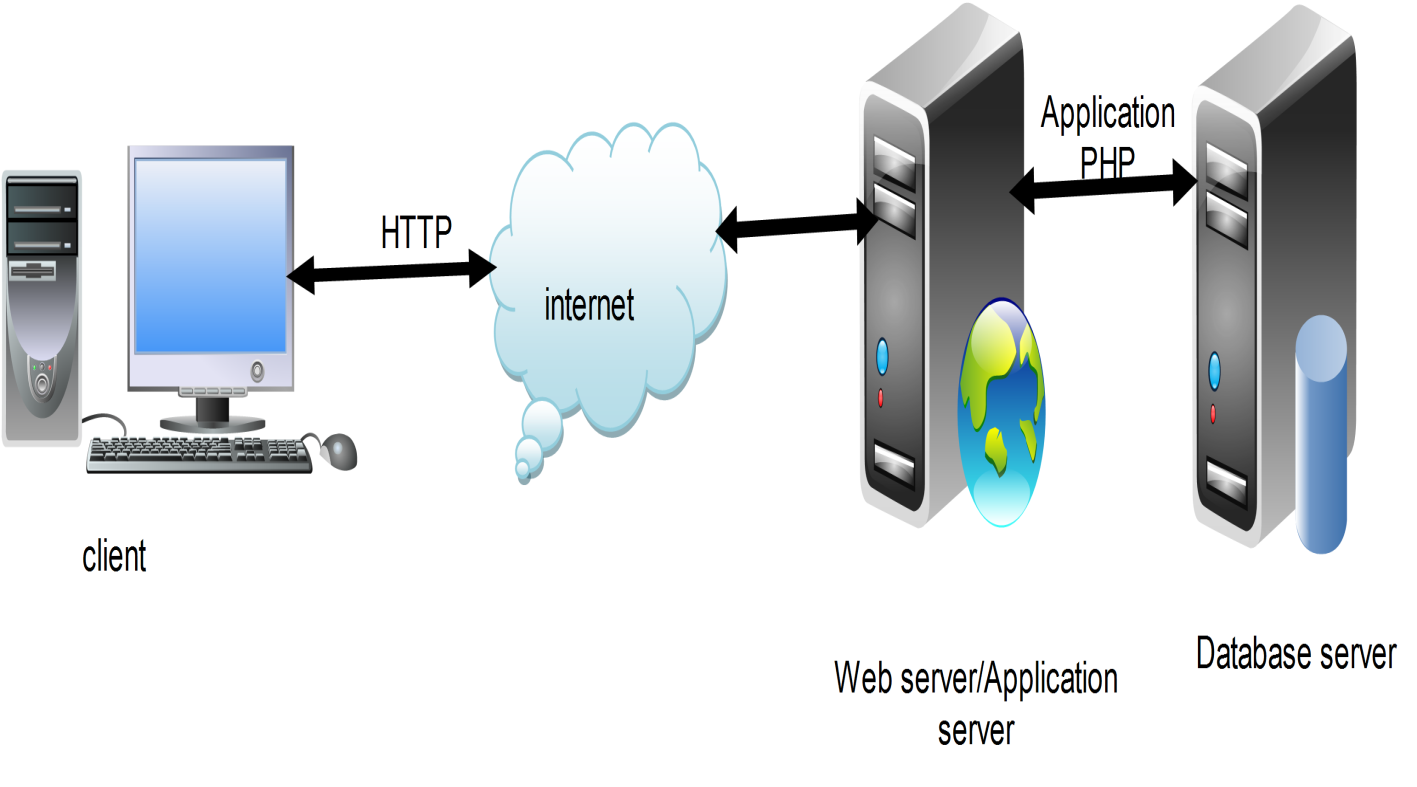
The architecture used for the proposed system is a 3 tier Client/Server Architecture. Such as interface layer providing front end user interface, application logic layer managing system operations, and storage layer storing persistent objects. The client tier is the applications user interface containing data entry forms and client side applications. It displays data to the user. Users interact directly with the application through user interface. The client tier interacts with the web/application server to make requests and to retrieve data from the database. It then displays to the user the data retrieved from the server. A client can use Internet browsers to access information from the server.

The middle tier (web/application server) implements the business logic, controller logic and presentation logic to control the interaction between the application’s clients and data. The controller logic processes client requests such as requests to view user’s request, to record or to retrieve data from the database. Business rules enforced by the business logic dictate how clients can and cannot access application data and how applications process data.

A web server is a program that runs on a network server (computer) to respond to HTTP requests. The most commonly used web servers is Apache. The web server used in this system is Apache. HTTP is used to transfer data across an Intranet or the Internet. It is the standard protocol for moving data across the internet.

The third tier, called the data tier, maintains the applications data such as Candidates etc.

It stores these data in a relational database management system (RDBMS).

****

**Figure 19: System Architecture Model for Proposed System**

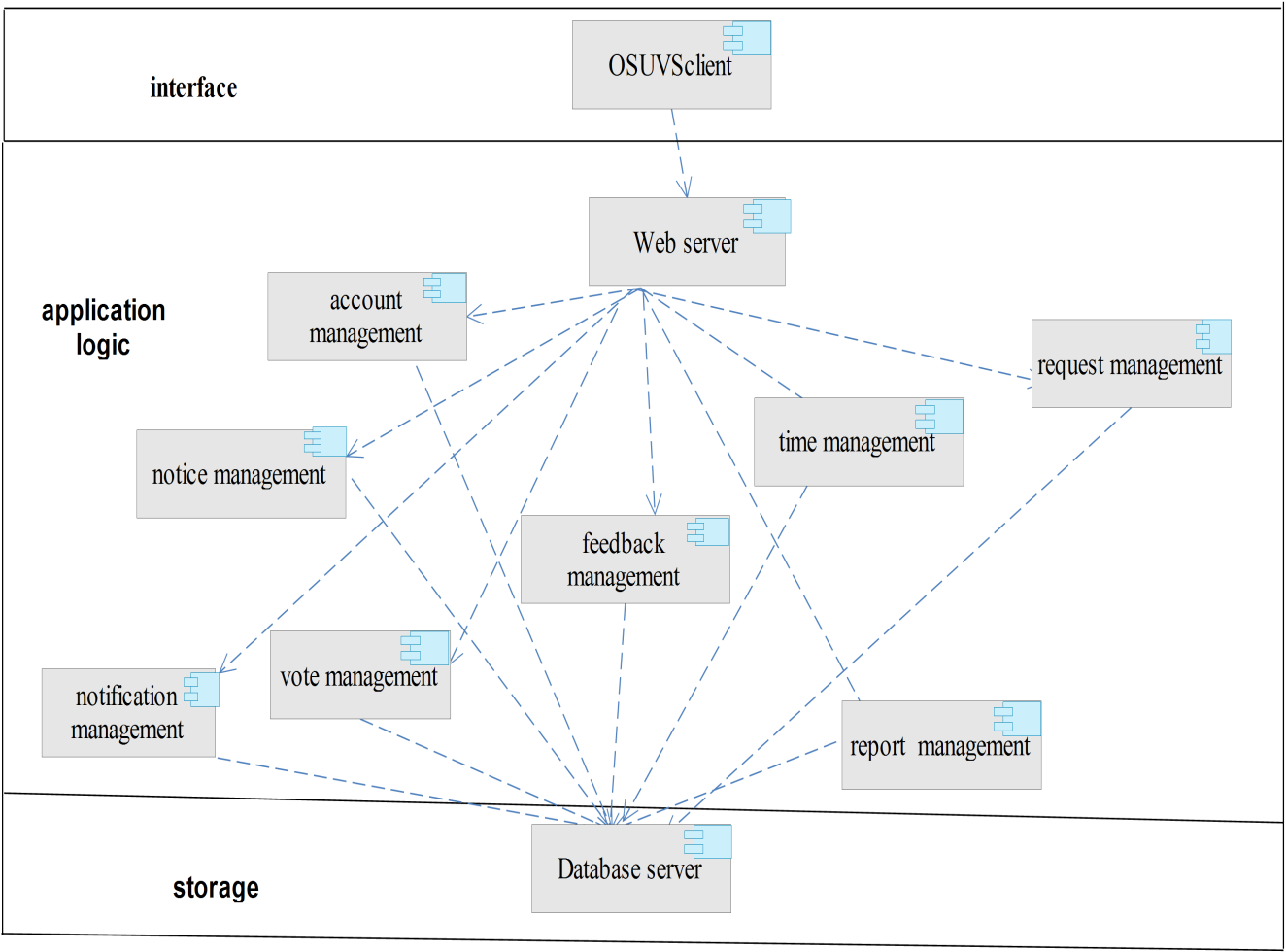
### 3.4.2 Subsystem Decomposition

This section describes the decomposition of system into subsystems and their responsibilities [5]. The System user interface subsystem provides user friendly interface for user.

System decomposition is undertaken to reduce the complexity of the system and gaining insight into the identity of the constituent components. The system is decomposed in to sub-systems which are a collection of classes, associations, operations, events and constraints that are closely interrelated with each other.

The proposed web based student union voting is decomposed in to smaller sub-system as shown in the following figure. These sub-systems are further decomposed in to other sub-systems. The major sub-system identified includes:-

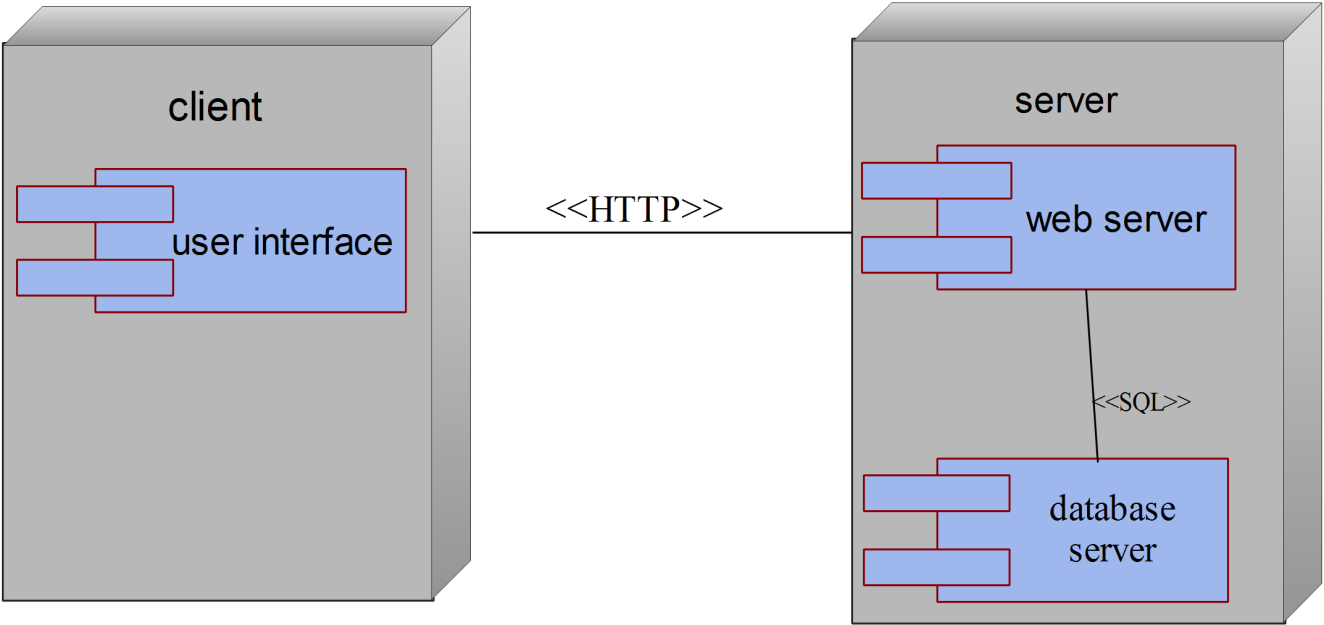
* **User management subsystem**: This subsystem allows the user to access the data or operation within their own given privilege, that means Administrator manage the user by giving their own privilege to access the system.
* **Account Management**:-these subsystem users to create user account before accessing the system as well as allow admin to create, update, search, view and block user account.
* **Request management**: - in this sub system the student can send their request to the Administrator to participate either in a Candidate or a Voter form in voting process as well as Administrator see and approve the request.
* **Notice management:-**in this subsystem the SSD can post notice as well as Candidate and Voter view notice.
* **Feedback management:-** - in this sub system the Voter and Candidate can raise their comments as well as suggestions about the system and Administrator to view and delete feedback.
* **Vote management:-**in this subsystem Voter vote Candidate and users view the result.
* **Notification management:-**in this sub system admin and SSD send notification as well as SDD and registrar see notification.
* **Time management**: - in this subsystem SSD set student apply date and election date.
* **Report Management**: - in this sub system the user view report generated by admin and allows admin to generate report.



**Figure 20: Component Diagram of Student Union Voting System**

### 3.4.3 Hardware and Software Mapping

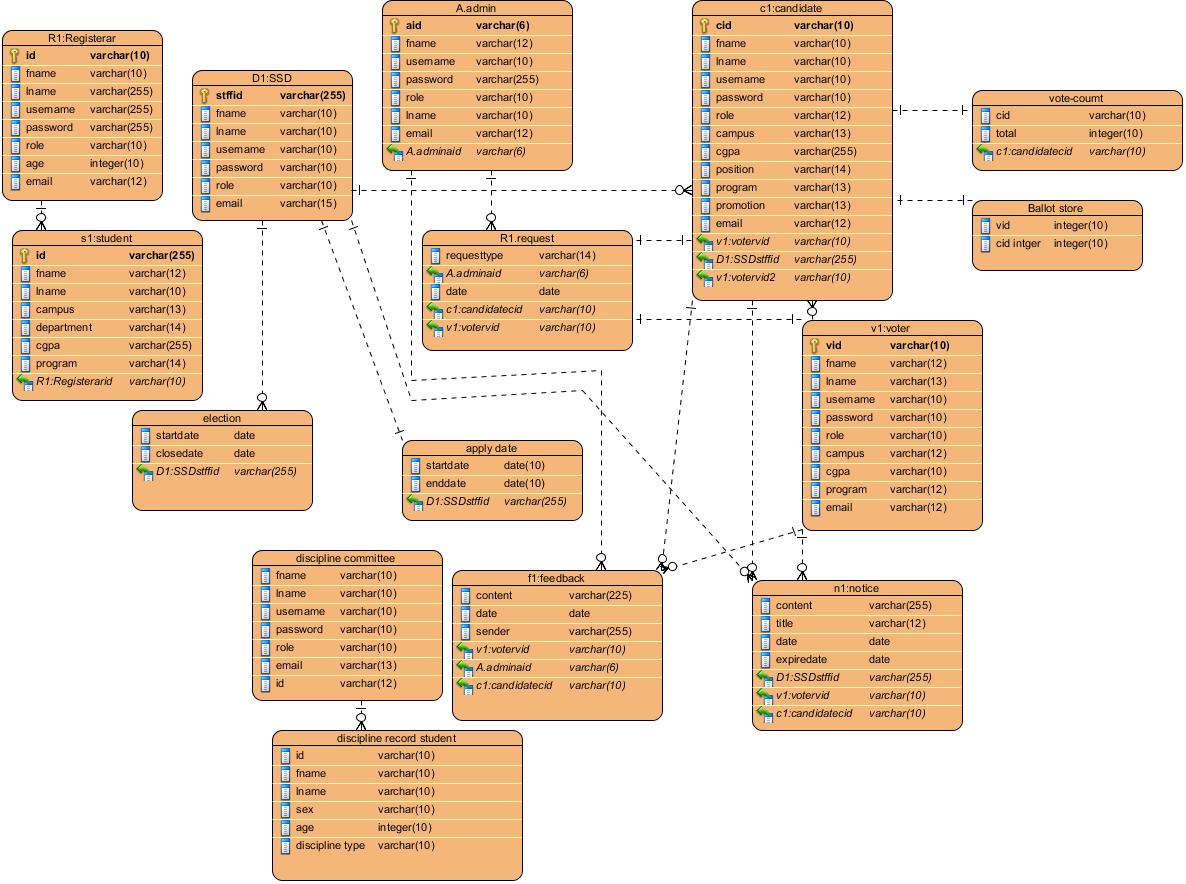
This section describes the HW/SW mapping of the proposed system. To describe this we have use the UML deployment diagram. [5] Deployment diagram is a structure diagram which shows architecture of the system as deployment or distribution of software artifacts to deployment targets. Deployment diagrams 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.



**Figure 21: Deployment Diagram**

### 3.4.4 Persistent Data Management

Persistent data management is basically used to represent the design of the database. Database design is the process of producing a detailed data model of a database. Online student union voting system will have database tables to store, access and maintain information that flows in the system.



**Figure 22: Persistent Data Management**

### 3.4.5 Access Control and Security

In the systems, different actors have access to different functionality and data. Therefore these privileges prevent unauthorized users from accessing data’s which they don’t have granted to access. Authentication: This take place by letting users to insert their user name and password in the displayed login form.

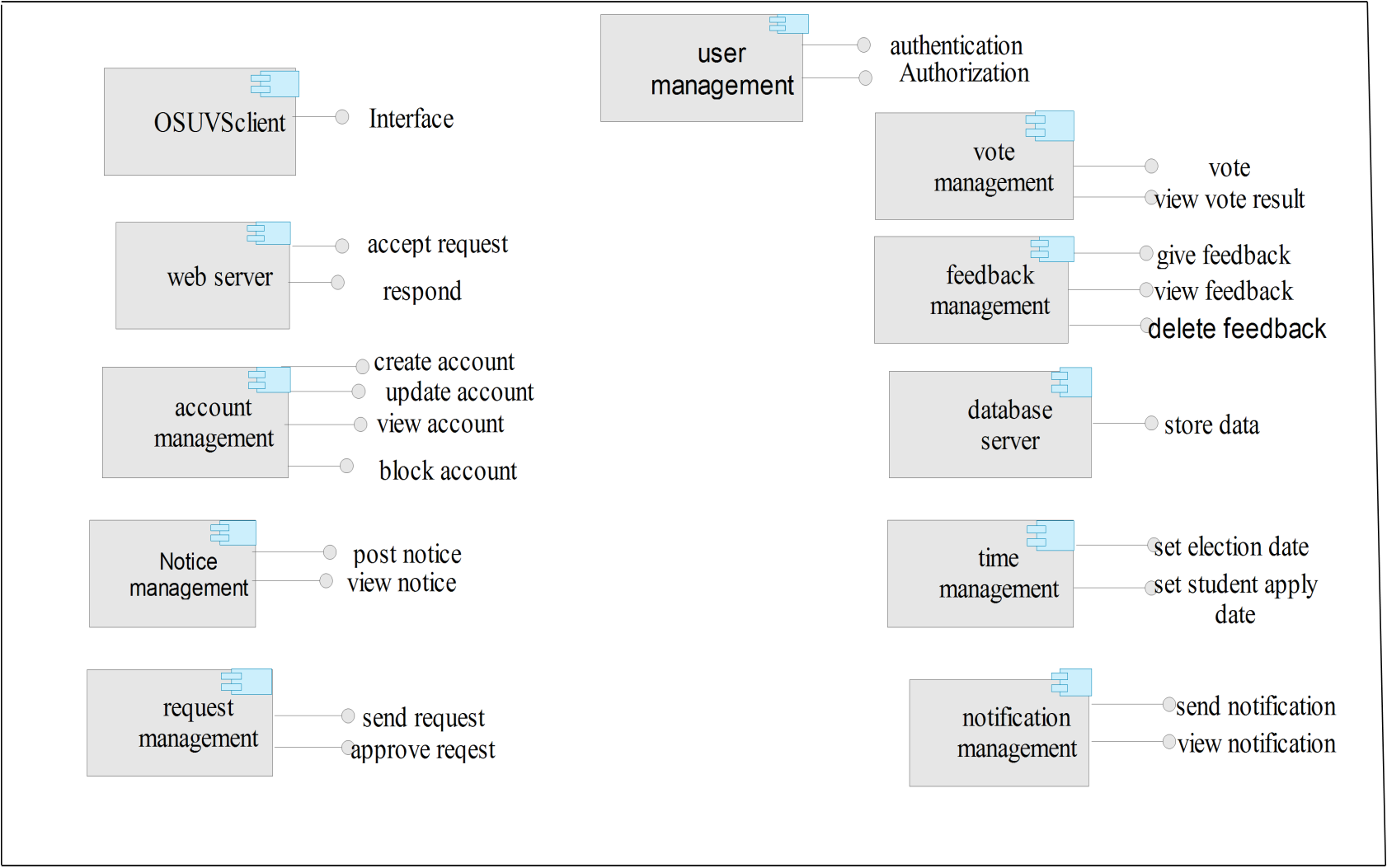
Authorization: This takes place by preventing users from participating in specific tasks on which he/she doesn’t have grant to access.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Actor | | | | |  |
| Function | Admin | SSD | Registrar | Candidate | Voter | Discipline  Committee |
| Post notice |  |  |  |  |  |  |
| Send notification |  |  |  |  |  |  |
| Send request |  |  |  |  |  |  |
| Approve request |  |  |  |  |  |  |
| Vote |  |  |  |  |  |  |
| View notice |  |  |  |  |  |  |
| Give feedback |  |  |  |  |  |  |
| View feedback |  |  |  |  |  |  |
| Delete feedback |  |  |  |  |  |  |
| Set election date |  |  |  |  |  |  |
| View election result |  |  |  |  |  |  |
| View notification |  |  |  |  |  |  |
| Send student data |  |  |  |  |  |  |
| View voter |  |  |  |  |  |  |
| Generate report |  |  |  |  |  |  |
| View report |  |  |  |  |  |  |
| View Candidate |  |  |  |  |  |  |
| Set student apply date |  |  |  |  |  |  |
| Send discipline record student |  |  |  |  |  |  |

**Table 16: Table of Access Control and Security**

### 3.4.6 Subsystem Service

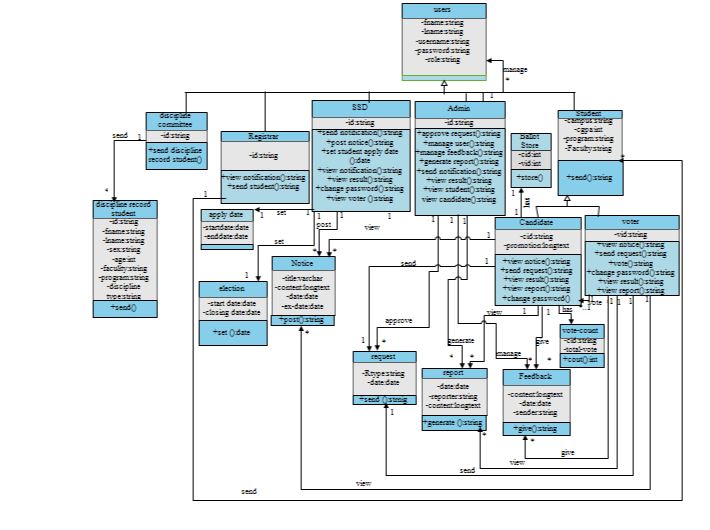
In this section we describe the service provided by each sub system. In this case, we will define the subsystem decomposition by identifying the services provided by each subsystem. Generally this section describes the service provided by each subsystem.



**Figure23: Component Diagram Subsystem Service**

### 3.4.7 Detailed Class Diagram

Class diagram with visibility and signature specified for each attributes and operations. In this class diagram we define which attributes and operations are private to that class, which attributes and operations can be accessed by the class decedents (protected), and which attributes and operations are publicly accessible by the class user (public). We also define the return type of each operations as well as the number and type of parameters of each operations.

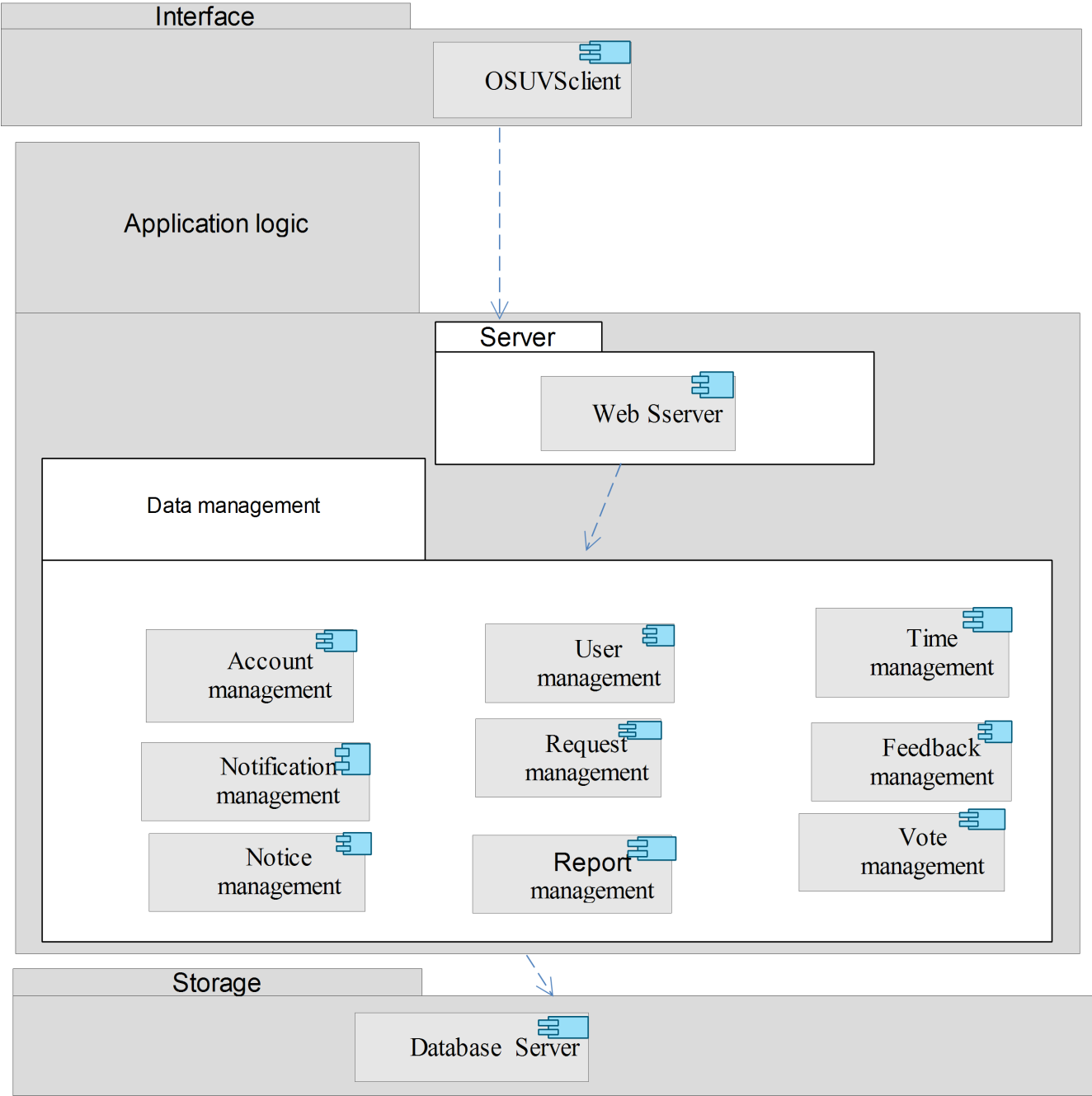


**Figure 24: Detail Class Diagram**

### 3.4.8 Package Diagram

OSUVS takes 3 tier architecture styles. There are sub systems listed above and the dependency between them is described also. In this sub topic we try to draw packages for each sub system. We group the subsystems into three packages.

* Interface package layer is client tier that is user interface. This package is responsible for providing front end user interface for users to initiate use cases.
* Application logic package layer is middle tier that contain sub package server and data management package. Server package holds web Server subsystem, which is responsible for accepting requests from Interface package, process the request and respond back. Server package uses the Data Management package to retrieve data from Storage package.
* Storage package layer is data tier that is database of the system. This package store persistent object or information.



**Figure 25: Package Diagram**

# CHAPTER 4

# 4. IMPLEMENTATION

## 4.1 Introduction

The implementation part used to automate the requirement based document for the organization as a web based applications. The Implementation is the Coding phase of all parts of documents that gathered starting from requirement analysis to Design phase. So now the team is in a position of implement all documents gathered and designed into the code so that the system will be implemented for the user to be used for the purpose to develop to the system.

## 4.2 Mapping model to code

### 4.2.1 Mapping association

Bidirectional one-to-one association

1 1

1 1

Account

Us User

public class user { public class Account {

private Account account; private user owner;

public user() { public Account (user owner) {

account = new Account (this); this.owner = owner;

} }

public Account getAccount() { public user getOwner() {

return account; return owner;

} }

public Account setAccount(Account public user setOwner( user

account) { owner){

this.account=account; this. owner=owner;

} }

}

**One-to-many association**

\*

1 1

\\

NewRequest

Admin

Public class Admin {

public class NewRequest{

Private Set NewReqest;Private Admin admin;

Public Admin () {public void setadmin

NewEmployee = new HashSet();(Admin newAdmin) {

}if (admin != newAdmin) {

Public void ApproveNewRequest(NewRwquest R) {

Admin old = admin;

NewRequest.add(R);

admin= newAdmin;

setRequest(this);if (newAdmin != null)

}ApproveNewRequest(this);}

}}

}}

### 4.2.2 Mapping the class model to a storage schema is performed

A schema is a description of the data, that is, a meta-model for data. In UML, class diagrams are used to describe the set of valid instances that can be created by the source code. Similarly, in relational databases, the database schema describes the valid set of data records that can be stored in the database. Relational databases store both the schema and the data.

Primary key

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| userid | fname | Lname | sex | Age | Password | role | status | photo |
| 6 | Mulatu | Mekonnen | male | 23 | 123 | SSD | 1 | eb.jpg |

**Table 17 User table**

Foreign key

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Title | Date | Ex-Date | content | sender |
| 17 | Election | 2019-6-25 | 2019-6-26 | Election information | Mulatu |

**Table 18 Notice table**

## 4.3 source code of major class packages and interface

**Login sample code for login**

<?php

session\_start();

if(isset($\_session['counter']))

$\_session['counter']+=1;

else

$\_session['counter']=1;

?>

<html>

<head>

function show\_password(){

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

var checkbox = document.getElementById("checkbox");

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

pass.type = "text";

checkbox.checked = true;

}else{

pass.type = "password";

checkbox.checked = false;

} }

function ValidateAlpha(evt)

{

var keyCode = (evt.which) ? evt.which : evt.keyCode

if ((keyCode < 65 || keyCode > 90) && (keyCode < 97 || keyCode > 123) && keyCode != 32 && keyCode != 8 && keyCode != 9)

{

alert("please enter Only letters! ")

return false;

}}

</script>

</head>

<body>

<div id="container">

<div id="header">

<?php

include("headerhome.php");

?>

</div>

<div id="menu">

<?php

include("menu.php");

?>

</div>

<div id="content">

<div class="login" style="margin-left:0%;" >

<div class="login-triangle" style="margin-left;"></div>

<h2 class="login-header">Just Login here</h2>

<form class="login-container" method="post" enctype="multipart/form-data">

<p><i class="fa fa-user"></i>User Name:<input type="text" placeholder="username" name="username" onkeypress="return ValidateAlpha(event)"></p>

<p> <i class="fa fa-unlock-alt"></i>Password:<input type="password" placeholder="\*\*\*\*\*\*\*\*" name="password" id="password" pattern=".{3,}" title="three or more characters"></p>

<script type="text/javascript">

var f1 = new LiveValidation('password');

f1.add(Validate.Presence,{failureMessage: "X"});

</script>

<p><input type="submit" value="Login" name="login"></p>

<label style="margin-left: 100px;float: right;margin-top: -50px;">

<input type="checkbox" name="checkbox" id="chekbox" onClick="show\_password();">Show Password

</label>

<P><a href="insertforgetpasswordinfor.php" style="text-decoration:none;font-style: italic;">Forget Password</a></P>

<span style="width: 10px;"><input type="reset" value="Reset" style="width: 150px;float: right;margin-top: -40px;"/></span>

</form>

<?php

include("Database/connection.php");

if($con)

{

if(isset($\_POST['username'])&&isset($\_POST['password']))

{

if(isset($\_POST['login']))

{

$uname=$\_POST['username'];

$upass=$\_POST['password'];

if(empty($uname)||empty($upass)){

echo "<script type='text/javascript'>alert(' please fill all field')</script>";

}

else{

//$pass=md5($upass);

$result=mysql\_query("Select \* from user where (user.username='$uname' and

user.password='$upass' and user.status=1)");

$row=mysql\_fetch\_array($result);

$username=$row['username'];

$password=$row['password'];

$role=$row['role'];

$photo=$row['photo'];

$userid=$row['userid'];

$campus=$row['campus'];

if($username==$uname&&$password==$upass)

{

$sql=mysql\_query("delete from attempt");

$\_SESSION['username']=$uname;

$\_SESSION['password']=$upass;

$\_SESSION['role']=$role;

$\_SESSION['userid']=$userid;

$\_SESSION['sphoto']=$photo;

$\_SESSION['campus']=$campus;

if($\_SESSION['role']=="Adminstrator")

header("location:Admin/Admin.php");

else if($\_SESSION['role']=="SSD")

header("location:ssd/SSD.php");

else if($\_SESSION['role']=="Candidate"&&$\_SESSION['campus']=="tedy")

header("location:Tedycandidate/Candidate.php");

else if($\_SESSION['role']=="Voter"&&$\_SESSION['campus']=="tedy")

header("location:Tedyvoter/Voter.php");

else if($\_SESSION['role']=="Candidate"&&$\_SESSION['campus']=="marki")

header("location:Marakicandidate/Candidate.php");

else if($\_SESSION['role']=="Voter"&&$\_SESSION['campus']=="marki")

header("location:Marakivoter/Voter.php");

else if($\_SESSION['role']=="Candidate"&&$\_SESSION['campus']=="Facil")

header("location:Facilcandidate/Candidate.php");

else if($\_SESSION['role']=="Voter"&&$\_SESSION['campus']=="Facil")

header("location:Facilvoter/Voter.php");

else if($\_SESSION['role']=="Candidate"&&$\_SESSION['campus']=="Tseda")

header("location:Tsedacandidate/Candidate.php");

else if($\_SESSION['role']=="Voter"&&$\_SESSION['campus']=="Tseda")

header("location:Tsedavoter/Voter.php");

else if($\_SESSION['role']=="Candidate"&&$\_SESSION['campus']=="GC")

header("location:GCcandidate/Candidate.php");

else if($\_SESSION['role']=="Voter"&&$\_SESSION['campus']=="GC")

header("location:GCvoter/Voter.php");

else if($\_SESSION['role']=="Registrar")

header("location:Registrar/Registrar.php");

else if($\_SESSION['role']=="discipline\_committee")

header("location:discipline\_committee/discipline\_committe.php");

else

echo "Unknown Role";

}

else

{

$count="insert";

$sql=mysql\_query("select\*from attempt");

$total=mysql\_num\_rows($sql);

$total++;

if($total>3)

{

header("location:logincounter.php");

}

else

{

echo "<script type='text/javascript'>alert(' Please enter correct username and password')</script>";

echo "<script type='text/javascript'>alert('You have 4 Chance now you tries $total times')</script>";

//echo "<font color=red>Please enter correct username and password</font>";

//echo "<br>You have 4 Chance now you tries $total times";

$insert=mysql\_query("insert into attempt values('$count')");

}}}

}else

echo "Database not connected!";

}}

?> </div>

</div>

<div id="footer">

<?php

include("footer.php");

?>

<a name="bottom"></a>

</div>

</div>

</body>

</html>

# CONCLUSION

The project titled Web Based OSUVS is easy to use and access i.e. no need of detail computer skill to use and access. In online student union voting system election can be conducted in free and fair manner in secret ballots. The aim of our project is to develop an interactive, electronic, voting system for university of Gondar 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. The project achieves the objective listed on the SPMP document, functional and nonfunctional requirements, and the problems listed in the first document part, and done with project scope appropriately. To achieve our goal the project team has spent all of its time on the project by performing the tasks individually and in group based on the schedule available.

# RCOMMANDATION

According to scope of the project the team develops web based application. During doing this system the team members has faced different challenges due to lack of project development resources especially time. And also unexpected problems such as the project attack by virus. Nevertheless by the cooperation of all the group members and the advisor the team is now able to reach to the final result. All the group members strongly fight these challenges and take the turn to the front. Because of the time constraint, the system may have limitation but in the feature 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.

* Develop Amharic language.
* The system allows candidate can attach CV.
* Candidates take exam.

# REFERENCE

[1] Modern system analysis and design third edition by JEFFERY A.HOFFER, JOEY F.GEORGE, and JOSEPH S.VALACLCH (functional and nonfunctional requirement.) on December 12/04/2011 EC.

[2] Modern system analysis and design third edition by JEFFERY A.HOFFER, JOEYF.GSEORGE, and JOSEPH S.VALACLCH (design sequence diagram.) on December 25/04/2011 EC.

[3]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 12/04/2011 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 27/04/2011

[5] Bernd Bruegge& Allen H. Dutoit.**Object-Oriented Software Engineering**

**Using UML, Patterns, and Java™ Third Edition,** Prentice Hall, United States,

1985

[6] https://www.4icu.org/reviews/7655.html