

DECLARATION

I, YIRENKYI EMMANUEL (FMS/1689/13), declared that this research report is my original work. It has not been submitted to any other university or higher institution of learning for any award. Any other author's work has clearly been referenced.

Signature.....

Date.....

YIRENKYI EMMANUEL (FMS/1689/13)

CERTIFICATION

I certify that this project report “**DEPARTMENTAL MANAGEMENT SYSTEM**” was carried out independently by **YIRENKYI EMMANUEL**, in the Department of Computer Science under the supervision of Dr. Gabriel Armah as part of the requirement for the award of Bachelor of Science in Information Technology.

.....
SIGNATURE

DR. EDEM BANKAS
(HEAD OF DEPARTMENT)

DEPARTMENT OF COMPUTER SCIENCE

.....
SIGNATURE

DR. GABRIEL ARMAH
(SUPERVISOR)

DEPARTMENT OF COMPUTER SCIENCE

DEDICATION

I dedicate this work to my family and friends who have been a great source of support and inspiration throughout my education.

ACKNOWLEDGEMENT

My greatest appreciation goes to the Almighty God, for protecting and guiding me throughout my stay on campus and for giving me the best of knowledge to accomplish my project successfully.

I am highly grateful to my supervisor, Dr. Gabriel Armah who provided me with an opportunity to work under his kind supervision.

Also, I would like to thank all friends who encouraged me and shared their ideas with me throughout this project.

Finally, I would also like to thank my family for their tremendous support in all endeavors throughout my education.

God bless you all.

ABSTRACT

This project is an analysis and implementation of a computerized system that is going to be used by departments at the University for Development Studies and any other university in general. This is to speed up processes and activities associated with the departments. This system is a windows based application developed under the .NET platform using Visual Basic 2013.

This system would help in the registration of staff members, students, associations, assignment of courses to programmes and courses to lecturers.

The system would also help to compute students' course grade by computing all available criteria such as assignment, mid-trimester or mid-semester exams and end of trimester or semester exams.

Determining the financial status of the department, the system would help to track all income and expenditure transactions and produce report based on any periodic interval.

Lastly this software includes an inbuilt E-Voting System that can be used to run elections by the department.

ABBREVIATION

UMS-----	University Management System
DMS-----	Department Management System
EVS-----	Electronic Voting System
UDS-----	University for Development Studies
IT-----	Information Technology
MIS-----	Management Information System
UML-----	Unified Modelling Language
SQL-----	Structured Query Language
T-SQL- -----	Transact Structured Query Language
ERP-----	Enterprise Resource Planning

TABLE OF CONTENTS

CONTENTS	PAGE
DECLARATION	i
CERTIFICATION	ii
DEDICATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
ABBREVIATION	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	ix
CHAPTER ONE (1)	1
1.0 INTRODUCTION	1
1.1 BACKGROUND OF STUDY	2
1.2 PROBLEM STATEMENT	3
1.3 STUDY OBJECTIVES	4
Specific Objectives	4
CHAPTER TWO	6
2.0 LITERATURE REVIEW	6
2.1 MANAGEMENT INFORMATION SYSTEMS	6
2.1.1 UNIVERSITY MANAGEMENT SYSTEM (UMS)	8
CHAPTER THREE	10
3.0 METHODOLOGY	10
3.1 HOW THE SOFTWARE WAS DEVELOPED USING THE ITERATIVE WATERFALL MODEL	11
3.1.1 DESCRIPTION OF THE PHASES OF THE ITERATIVE WATERFALL MODEL	12
3.1.2 IMPORTANCE OF THE MODEL	13
3.2 LANGUAGES AND TOOLS USED FOR THE SYSTEM DEVELOPMENT	14
3.2.1 VISUAL BASIC 2013	14
3.2.2 STAR UML	15
3.2.3 BACK END TOOL	16
CHAPTER FOUR (4)	17
4.0 DESIGN AND FUNCTIONS OF THE PROPOSED SYSTEM	17
4.1 SCOPE	17

4.2 SYSTEM REQUIREMENTS	19
4.3 THE PROPOSED SYSTEM.....	20
4.4 USAGE SCENARIO PRESENTATION	21
4.4.1 GENERAL USE CASE DIAGRAM OF THE SOFTWARE	22
4.4.2 USE CASE DIAGRAM OF THE ADMINISTRATOR	23
4.4.3 USE CASE DIAGRAM OF THE LECTURER.....	24
4.4.4 USE CASE DIAGRAM OF THE ACCOUNTANT	25
4.4.5 USE CASE DIAGRAM OF THE ACADEMIC (RESULTS) SECTION	26
4.4.6 USE CASE DIAGRAM OF THE TRANSACTION SECTION	26
4.4.7 USE CASE-DIAGRAM OF THE INBUILT EVS SECTION.....	27
4.5 ACTIVITY DIAGRAMS.....	28
4.5.1 SYSTEM LOGIN ACTIVITY DIAGRAM	28
4.5.2 STUDENT REGISTRATION ACTIVITY DIAGRAM.....	29
4.5.3 PROGRAMME-COURSE ASSIGNMENT ACTIVITY DIAGRAM.....	30
4.5.4 COURSE-LECTURER ASSIGNMENT ACTIVITY DIAGRAM.....	31
4.6 USER INTERFACE DESIGN	32
4.6.1The Splash Screens	32
4.6.2 The Login Interfaces.....	34
4.6.3 The Main Interfaces.....	37
4.6.4 Some Registrations and Assignment Section of the software	38
4.6.5 The Academic Section of the software	44
4.6.6 THE TRANSACTION SECTION INTERFACES	48
4.6.7 THE E-VOTING SYSTEM SECTION.....	49
CHAPTER FIVE	52
5.0 CONCLUSION	52
5.1 RECOMMENDATIONS	53
REFERENCES	54
APPENDIX.....	56
CODING AND IMPLEMENTATION	56

LIST OF FIGURES

Figure 4.0 General Use case diagram for the system.....	22
Figure 4.1 Use case diagram of the administrator.....	23
Figure 4.2 Use case diagram of the lecturer.....	24
Figure 4.3 Use case diagram of the accountant.....	25
Figure 4.4 Use case diagram for the academic section.....	26
Figure 4.5 Use case diagram of transaction section.....	26
Figure 4.6 Use case diagram of the inbuilt Electronic voting system.....	27
Figure 4.7 System Login activity diagram.....	28
Figure 4.8 Student Registration Activity Diagram.....	29
Figure 4.9 Programme-Course assignment activity diagram.....	30
Figure 4.10 Course-Lecturer activity diagram.....	31
Figure 4.11 The main splash screen.....	32
Figure 4.12 A splash screen indicating transition into the E-voting system section.....	33
Figure 4.13 A splash screen indicating transition into a new section after a vote has been cast.....	33
Figure 4.14 A splash screen showing a thank-you message after using the voting section.....	33
Figure 4.15 The main login section.....	34
Figure 4.16 The E-Voting administrative login section.....	35
Figure 4.17 The Entry code login section.....	36
Figure 4.18 The Voter login section.....	36
Figure 4.19 The overall main interface system section.....	37
Figure 4.20 The E-voting main interface section.....	37
Figure 4.21 Setting department Information.....	38
Figure 4.22 Create user section.....	39
Figure 4.23 Modify user section.....	39
Figure 4.24 Association Registration section.....	40
Figure 4.25 Student Registration section.....	40
Figure 4.26 Staff Registration section.....	41

Figure 4.27 Course-Programme Assignment Section.....	42
Figure 4.28 Course-Lecturer Assignment section.....	43
Figure 4.29 University Grade Mark Section.....	44
Figure 4.30 Course-Grading Section.....	45
Figure 4.31 Load student score section.....	46
Figure 4.32 Student Total Score in calculate result section.....	47
Figure 4.33 Students Grade in calculate result section.....	47
Figure 4.34 Registering Billing information.....	48
Figure 4.35 Dues and levies payments section under income.....	48
Figure 4.36 Creating administrator to manage election.....	49
Figure 4.37 Registering election positions.....	49
Figure 4.38 Registering Election Candidates.....	50
Figure 4.39 Uploading voters list.....	50
Figure 4.40 A voter Casting his vote.....	51
Figure 4.41 Election Results after voting.....	51

CHAPTER ONE (1)

1.0 INTRODUCTION

The information management environment has changed radically in the last few years, with the development of new and improved systems potentially outstripping our capacity to manage and exploit them fully, resulting in a situation where the role of information management has undergone dramatic transformation.

Developing software systems has become the fastest and most-perfect way of approaching problems using information technology and deriving the best of all possible solutions to meet customer's needs in this globalized world.

We need to be fast as the world grows, and we must also be fluid and flexible (Treloar, 2015). We need to develop these capabilities to survive. In this manner educational organizations must wish to follow this kind of environmental needs.

The scope of information technology systems needs more to be extended in institutions to enable management and academic boards keep track of records, manage records, assign certain duties and take decisions effectively.

University Management System (UMS), automates all the university transactions, internal workflow procedures of a university, as well as the interaction with the students and the instructors. UMS combines a suite of applications specially designed to bring together people, process, and technology towards improving relevance and quality of higher education.

Departmental Management Software (DMS) is also a powerful personal and academic management software tool designed for administrators of large departments, research institutions

and colleges. These types of software are effective and timesaving solution for managing department and they help to manage special commitments and information unique to the organization and bring flexibility to the data management tasks. This system can assist managers in achieving the aims of the institution, formulating strategic plans, distributing and assigning duties and keeping students and staff records.

This software and documentation helps to organize and administrate department works and also to improve the quality of their routine works.

1.1 BACKGROUND OF STUDY

Very comprehensive requirement analysis was done in order to get a very good understanding of the problem domain. After a vivid background study it was noticed that most of the departments in the university do not have any applications that will aid and assists them in their daily works. Problem arises when information such as student's records and others need to be retrieved from the University Management System which causes enough pressure on it. Imagining a situation where Heads of Departments need to assign courses to lecturers and programmes. What will be their source of help? Managing such situations can be a very tedious task when approached manually. It has to do with knowing all the various lecturers and the courses available and making sure a course assigned to a particular lecturer is not assigned to a different lecturer. In such situation, there is the need to implement fully computer aided management system to assist management in taking certain decisions.

1.2 PROBLEM STATEMENT

Although there are many departments at the University for Development Studies and other universities in Ghana. At present, most of the departments organize and manage activities manually. Head of departments face difficulties in assigning duties and performing other operations. With these manual approach in solving problem, departments on campus face difficulties when they are managing their routine works. Some of these difficulties include:

Lack of data security and reliability: Since student records and employee data are not kept in a central database, there is the vulnerability of losing such data. More especially student academic records such as assignments, presentation, mid-trimester results and final examination results are sometimes misplaced. Apart from the University Management System keeping general records of students, departments must also keep track of certain data.

Time consumption during process: Departmental heads and management can take certain decision in a more effective and timely manner if there is a shift from the manual system of taking decisions to a system where technology assist in making real and accurate decisions. With this, time consumption which is a major factor confronting various departments will be minimized.

Difficulties in preparation of accurate and prompt reports: It becomes difficult in preparing accurate reports as information is difficult to collect from various record books.

Difficulties in running elections: Among the problems that the departments face are difficulties in running their elections. Departments and the various associations within them run elections manually. As a result, elections become difficult to manage since it has to go through tedious and boring counting process.

Difficulties in assigning duties: Certain decisions such as assigning courses to programs and assigning courses to lecturers might seem difficult if they are handled manually. Also computing students total score from assessments such as assignment, presentation, practical, mid trimester exams and final exams might be difficult and time consuming when approach manually. A faster way of approaching such situation is by using a management system that will speed up such processes and duties.

1.3 STUDY OBJECTIVES

The main objective of this study is to develop a quality Department Management System to manage employees and student information as well as to run elections with ease.

Specific Objectives

- The system would help departments keep records of all their students, lecturers and other employees.
- The system would help head of departments assign trimester or semester courses to lecturers.
- The system would keep records of students' marks and compute them with ease ranging from assignments to final examination based on the lecturer grading criteria for the course and to help troubleshoot if a student assessment result is missing.
- The system would help to register various associations within the department
- The system would keep track and determine the financial status of the department by monitoring income and expenditure transactions.

- The system would help the department run elections by using an inbuilt Electronic voting system.
- The system would help register all courses and programs within the departments
- The system would help take minutes and attendance during staff meetings.

This report contains five chapters. The Appendix section of this report contains the code and implementation of the software.

CHAPTER TWO

2.0 LITERATURE REVIEW

“We are living in a knowledge society. Knowing what we know and managing it better to know more is considered the key enabler to sustainable economic growth. Seventy to eighty percent of economic growth is directly attributed to new or better knowledge. With information technologies (IT) readily facilitating the globalization of economic activity, the key competitive advantage of this era is the ability to create, use and disseminate knowledge effectively. Although traditionally creators, guardians and disseminators of knowledge is in a fast race, universities and institutions need to also adapt to the challenges and knowledge that society can bring and must evolve their roles in education, research and service to society.” (Whyte, 2008)

2.1 MANAGEMENT INFORMATION SYSTEMS

“MIS is a relatively new academic discipline and scholarly field of study with its own cumulative tradition and history.” (Culnan & Swason, 2010)

“It is an applied field concentrating on strategic, managerial and operational usage of various types of information technologies at societal, organizational, and individual levels. It draws upon several reference disciplines such as cognitive psychology, computer science, behavioral science, decision science, economics, operation management, organization theory and engineering.” (Weber, 2002)

Information systems contain information about significant people, places and things within the organization or in the environment surrounding it. Data, in contrast, are streams of raw facts

representing events occurring in organizations or the physical environment before they have been organized and arranged into a form that people can understand and use.

To deliver genuine benefits, information systems must be built with a clear understanding of the organization in which they will be used.

According to (Lucey, 2017), the central organization factors to consider when planning a new system are the following:

1. The environment in which the organization must function
2. The structure of the organization: hierarchy, specialization, routines and business process.
3. The organization culture and politics
4. The type of organization and its style of leadership
5. The kind of tasks, decisions and business process that the information system is designed to assist.

2.1.1 UNIVERSITY MANAGEMENT SYSTEM (UMS)

Universities face challenges related to academic affairs, scientific knowledge development, programs quality, market tendencies and among others, which make necessary to maintain a continuous learning cycle to check, assess, modify, redesign and restructure policies and procedures in order to meet the national accreditation standards and at the same time to be financially self-sustained. Decision making always involve some level of risk which is always accompanied by high levels of uncertainty. “Developing research focused on the use of “learning laboratories” where organization members are able to experiment through simulations without facing the risks of real implementation.” (Stermann, 2014)

University Management System (UMS) is a large database system which can be used to manage, maintain and secure a university’s day to day business.

University management system consists of fully independent functional modules each designed to improve the efficiency of the university and to provide information. (Morrissey, 2002)

This system is bundled with a very analytical and reporting engine. It provides extensive built in reports in different categories for different user group with varied objectives.

Departmental management system is a software application used to manage the daily activities undertaken within a department. Departmental and center administrators at large research intensive universities operate in a complex administrative environment. These administrators are forced to use complex management software such as ERP and Microsoft suits which does not really suit their operations.

Today's institutions need to manage more information than ever before. Without a solid internal infrastructure for administrators and departments to share data, critical institution and student information can be lost, or worse leading to a host of problems that can affect an institution image and endurance. To remain competitive, institutions need a simple solution that can run individual function, connect their entire operation and simplify day to day operational responsibilities, giving staff more time with students.

According to (Robert, 2008), “Information has always been an organization’s central resource. Without it the modern organization simply could not function. Managers spend an average of 4 weeks a year searching for a waited or misfiled, mislabeled, untracked, or “lost” information. Large organizations lose a document every 12 seconds. Records and information management brings significant classification benefits to an organization.”

Hence departments within institutions need to have a central database to keep track of all their records in order to facilitate their daily duties.

(Nyamekye & Afoakwah, 2014), developed a Computerized Students’ Registration Submission and Report System with lecturers profile management system for University for Development Studies. The system developed was able to keep the records of registration and submission done by the students, their results and the information of lecturers in the department of computer science. This work aims to extend the functionality of the work done by these two students and those who have previously worked with this kind of project.

CHAPTER THREE

3.0 METHODOLOGY

A software development process is a structure imposed on the development of a software product. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. (WebSphere, 2009)

The development models are the various processes or methodologies that are being selected for the development of the project depending on the project's aims and goals. There are many development life cycle models that have been developed in order to achieve different required objectives. The models specify the various stages of the process and the order in which they are carried out. The selection of model has very high impact on the testing that is carried out. (ISTQB, 2017)

In order to ensure the successful development of the project, the iterative waterfall model was used.

3.1 HOW THE SOFTWARE WAS DEVELOPED USING THE ITERATIVE WATERFALL MODEL

Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities were added.

Firstly the development process of the software started with a simple implementation of a subset of the software requirement and iteratively the evolving versions were developed until the full system was achieved.

The basic idea was to develop a system through repeated cycles and in smaller portions at a time (incremental), thereby taking advantage of what was learned during the development of earlier parts or version of the system.

At each iteration of the system implementation, design modifications were made and new functionalities were added.

Also during each iterative steps, a prototype was made and the result was shown to my supervisor for further improvement and recommendation of the prototype system.

3.1.1 DESCRIPTION OF THE PHASES OF THE ITERATIVE WATERFALL MODEL

Unlike the more traditional waterfall model, which focuses on a stringent step-by-step process of development stages, the iterative model is best thought of as a cyclical process. After an initial planning phase, a small handful of stages were repeated over and over, with each completion of the cycle incrementally improving and iterating on the software. Enhancement was recognized and implemented throughout each iteration, allowing the next iteration to be at least marginally better than the last.

- **Planning and Requirements:** As with most development project, the first step during the development of the software was to go through an initial planning stage. Specification documents, software and hardware requirements were generally prepared for the upcoming stages of the cycle.
- **Analysis and Design:** Once planning was completed, an analysis was performed to nail down the appropriate business logic, database models, and the like that will be required at this stage in the project. The design stage also occurred here. Establishment of technical requirements that will be utilized in order to meet the needs of the analysis stage was also analyzed.
- **Implementation:** With the planning and analysis out of the way, the actual implementation and coding process now begun. All planning, specification, and design documents were coded and implemented.
- **Testing:** The next step of the system implementation was to go through a series of testing procedures to identify and locate any potential bugs or issues that have cropped up. This was done several times to make sure the required and exact result was achieved.

- Evaluation: Once all prior stages were completed, it was time for a thorough evaluation of the development process. This allowed us (my supervisor and I) to examine where the project is at, where it needs to be, what can or should change, and so on thereby iterating through the cycle again.

3.1.2 IMPORTANCE OF THE MODEL

- Some working functionalities were developed quickly and early in the life cycle.
- Results were obtained early and periodically.
- We were able to measure development progress.
- Testing and debugging during smaller iteration was easy.
- With every increment, we were able to deliver results.
- Issues, challenges and risks identified from each increment were utilized or applied to the next increment.
- The model was able to support changing requirements.

3.2 LANGUAGES AND TOOLS USED FOR THE SYSTEM DEVELOPMENT

3.2.1 VISUAL BASIC 2013

Microsoft Visual Studio 2013 is the essential tool for performing basic development tasks. Use it to simplify the creation, debugging and deployment of applications on a variety of platforms, including SharePoint and the Cloud. Visual Studio 2013 Professional comes with integrated support for test-driven development, as well as debugging tools that help ensure high-quality solutions. (Microsoft DreamSpark, 2015).

This tool is a powerful language used in developing both desktop and online applications.

This application was developed using visual basic 2013 with visual studio 2013 ultimate edition. Few components such as SAP Crystal report and Dot Net Bar was installed to add additional functionality to the visual studio. The SAP Crystal report was used for the reporting aspect of the software while the Dot Net Bar helped in designing the interface forms. My choice was based on the reliable programming environment offered by Visual Basic and the highly cleared syntax of the programming language as well. Also the “intellisense” feature of the visual studio helped in making the coding process faster.

3.2.2 STAR UML

StarUML is an open source software modeling tool that supports Unified Modeling Language. It is based on UML version 1.4, provides eleven different types of diagram and it accepts UML 2.0 notation.

StarUML makes a clear conceptual distinction between models, views and diagrams. A Model is an element that contains information for a software model. A View is a visual expression of the information contained in a model, and a Diagram is a collection of view elements that represent the user's specific design thoughts.

StarUML supports the following diagram types

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Collaboration Diagram
- Statechart Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram
- Composite Structure Diagram

All use case diagrams and activity diagrams in this report were developed using the Star UML software.

3.2.3 BACK END TOOL

Transact Structured Query Language (T-SQL) was used as a back end tool during the development process. This helped in building the relational databases and to query data from the database with ease. It was easy to learn and implement. In combination with the visual basic, a high functionality and result was achieved through its simple syntax.

CHAPTER FOUR (4)

4.0 DESIGN AND FUNCTIONS OF THE PROPOSED SYSTEM

4.1 SCOPE

This departmental management system is a software that is used to manage the activities of a department ranging from registration to elections. This software is programmed to run in any institution running semester or trimester system. This software is developed to:

1. Register associations within the department
2. Register all programmes run by the department
3. Register all available courses done by the department
4. Register students in the department
5. Register lecturers in the department
6. Register meetings, taking attendance during meetings and writing minutes for meetings.
7. Assign courses to programmes in the department
8. Assign courses to lecturers
9. Upload student assignment, practical, presentation, mid-trimester or mid-semester, and final exams result based on the lecturer criteria settings and the university grading standard.
10. Compute total score, grade, grade point and remark in a particular course based on the available lecturer criteria settings and the university grading standard.
11. Determine the department income by:
 - Receiving all member billings such as dues and levies of staff members.

- Registering donations received by the departments
 - Registering any other income
12. Determine the department expenditure by:
- Keeping track of all purchases done by the department
 - Keeping track of all donations done by the department
 - Keeping track of any other expenditure
13. Determine the financial status of the department using the income and expenditure report
14. Run and determine elections results with ease by:
- Registering election positions
 - Registering voters
 - Registering candidates
 - Allowing voters to cast their vote based on the registered positions
 - Giving immediate elections results after voting process
15. Determine and print out reports from various activities such as:
- Student registration report
 - Meetings report
 - Course-programme assignment report
 - Course-lecturer assignment report
 - Students score report
 - Income report at periodic interval
 - Expenditure report at periodic interval
 - Income and Expenditure report within a periodic interval

- Voters registration list report

4.2 SYSTEM REQUIREMENTS

This software would run on computers that have the following system specifications:

- ✓ 1GB RAM or More
- ✓ 20GB hard disk space or more
- ✓ Windows 7 or higher version of windows
- ✓ .Net framework 4.5 or higher
- ✓ SQL Server 2012 express
- ✓ SAP Crystal Reports Runtime engine for .Net framework
- ✓ Access Database Engine

4.3 THE PROPOSED SYSTEM

This system has three types of users. The Administrator, the lecturer and the accountant. Users are required to provide their corresponding user name, password and access level before they can log into the system successfully.

All users must be registered by the administrator with their user name and password of which they can modify to their choice after logging into the system. Access level is strictly restricted by making sure users have access to only required components pertaining to their work. The administrative section might be managed by two users depending on the department choice. This involves:

1. The super administrator controlling both department activities and elections or
2. Two separate administrators – One controlling department activities and the other controlling elections.

The system is made up of four main section: The registration section, academic section, transaction section and the election section. The academic, transaction and election section are vividly explained using a use case scenario diagram whiles the registration section is explained using an activity diagram.

4.4 USAGE SCENARIO PRESENTATION

This section of the software describes the user interaction with the software using use cases and activity diagrams.

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems. (Rouse, 2015)

Use cases describe how external entities will use the system. These external entities can be human or other systems (called actors in UML terminology). The description emphasizes the users' view of the system and the interaction between the users and the system. Use cases help to further define system scope and boundaries. They are usually in the form of a diagram, along with a textual description of the interaction taking place. The actor is any outside entity that interacts with the system. An actor could be a human user (for instance, a rental agent), another software system, or an interface. Each interaction that occurs between an actor and the system is modeled as a use case. (Clark, 2013)

The system has three actors. The subsequent Use Case diagrams shows how the users interacts with the various forms and sub forms of the system.

4.4.1 GENERAL USE CASE DIAGRAM OF THE SOFTWARE

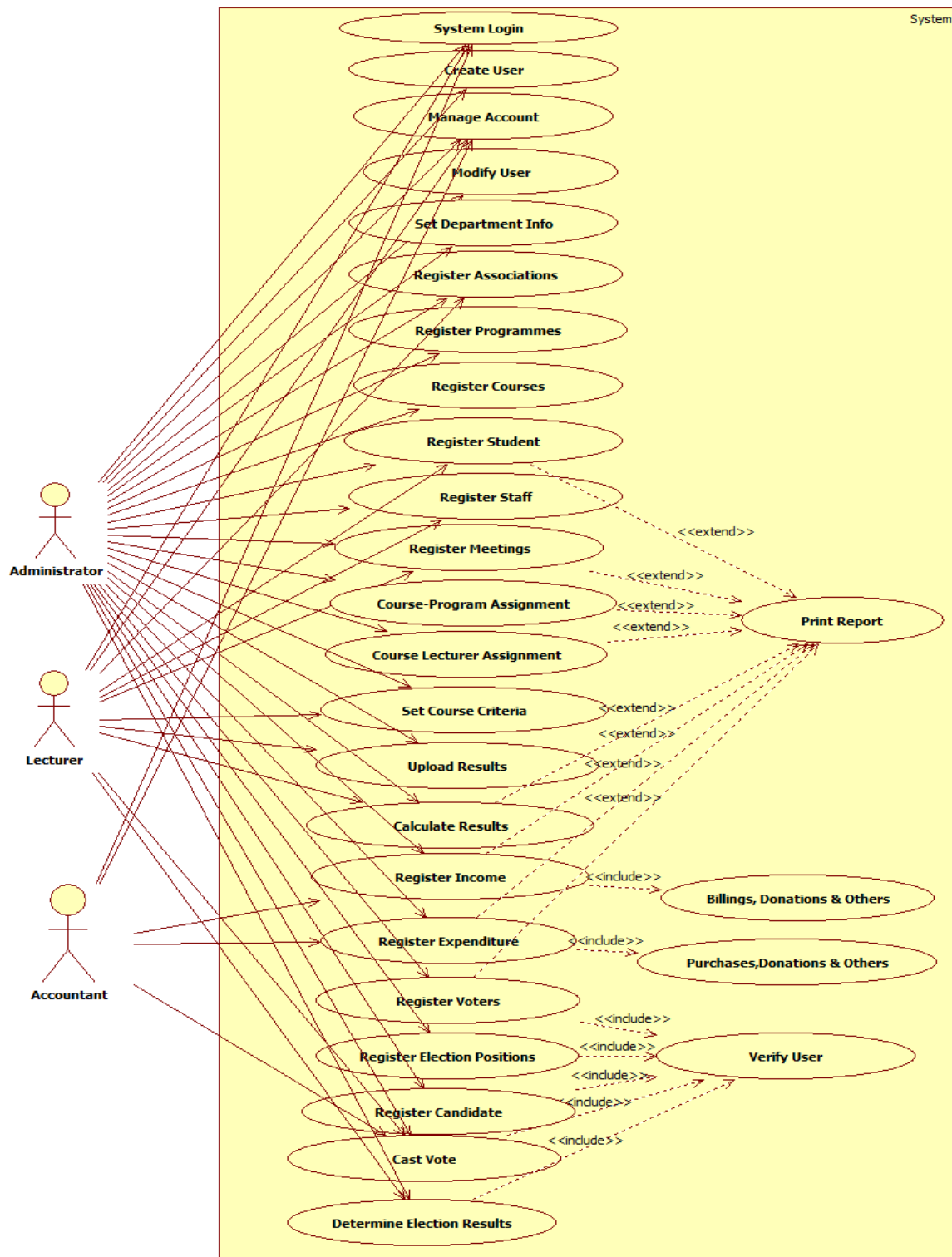


Figure 4.0 General Use case diagram for the system.

4.4.2 USE CASE DIAGRAM OF THE ADMINISTRATOR

The diagram shown below indicates how the administrator will interact with the section of the software allocated to him/her.

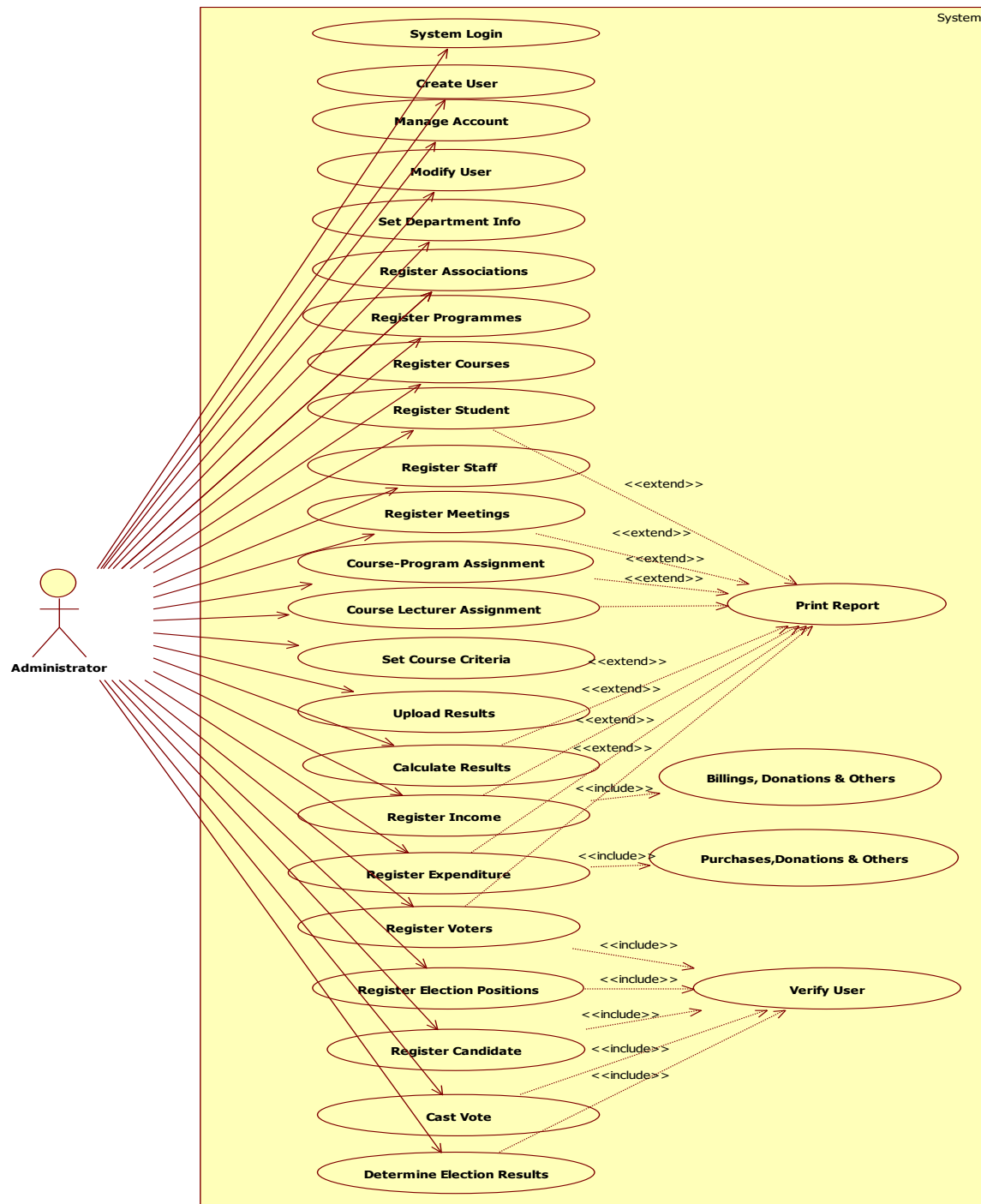


Figure 4.1 Use case diagram of the administrator

4.4.3 USE CASE DIAGRAM OF THE LECTURER

The diagram shown below indicates how the lecturer interacts with the section of the software allocated to him/her.

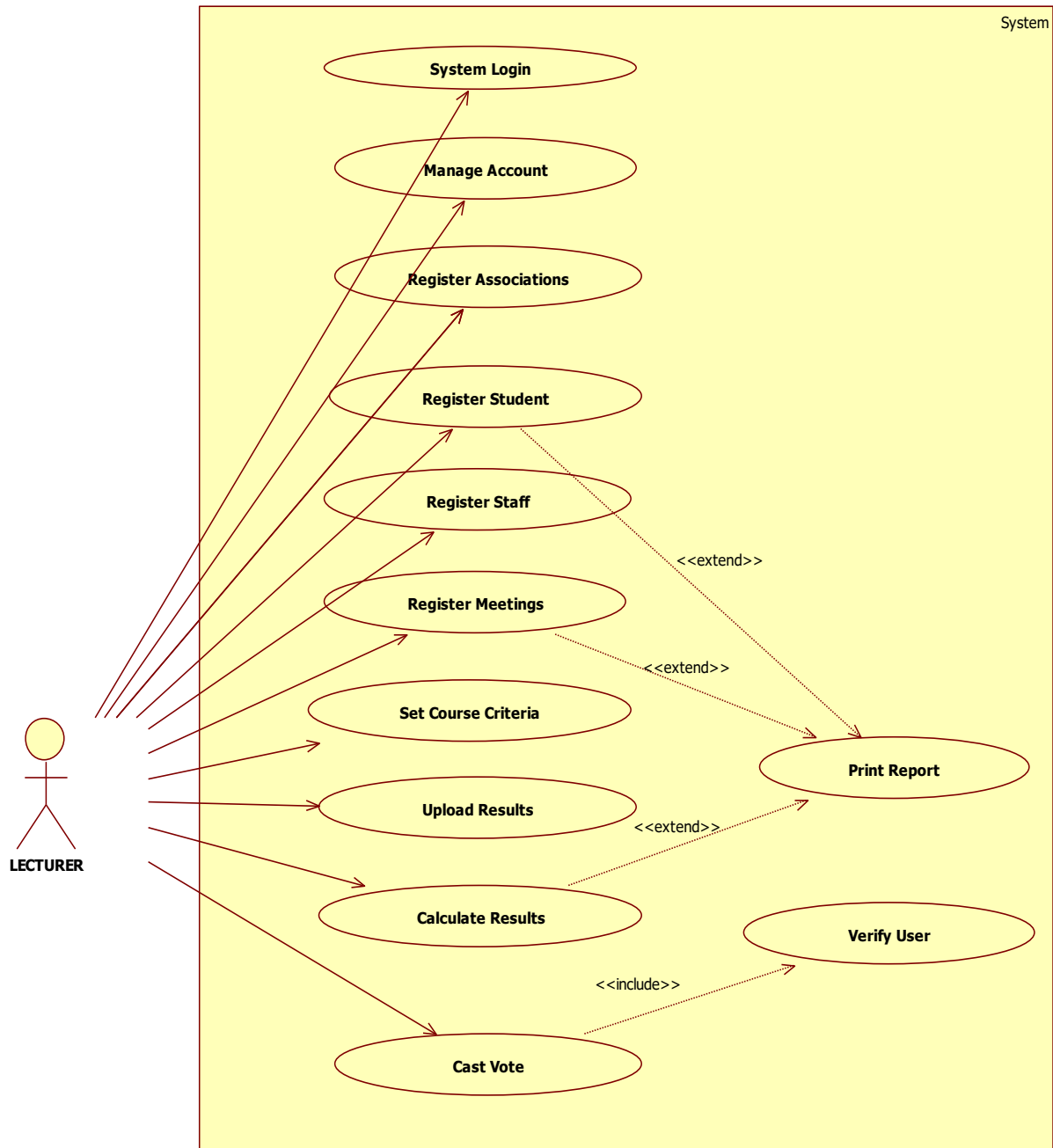


Figure 4.2 Use case diagram of the lecturer

4.4.4 USE CASE DIAGRAM OF THE ACCOUNTANT

The diagram shown below indicates how the accountant interacts with the section of the software allocated to him/her.

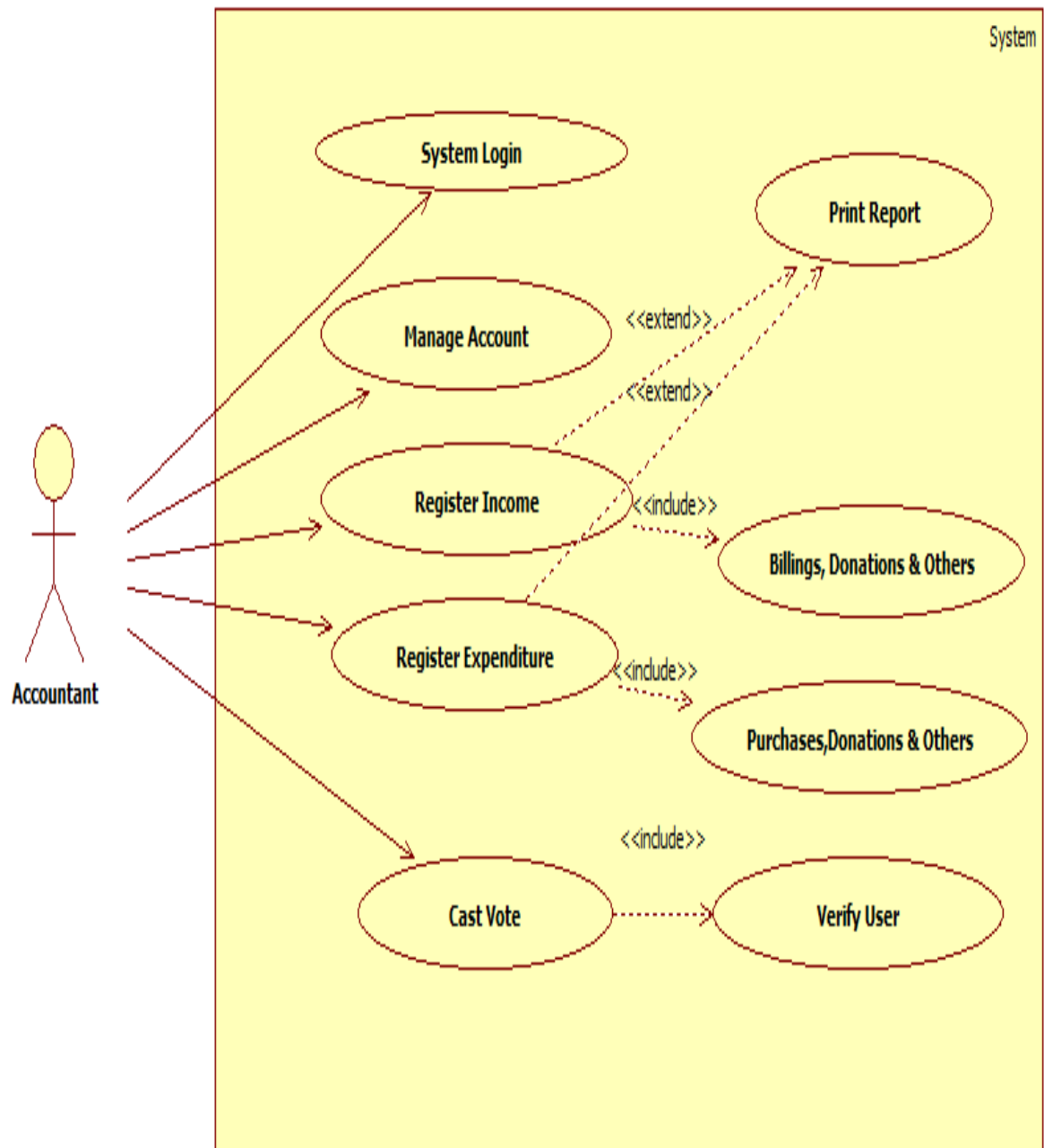


Figure 4.3 Use case diagram of the accountant

4.4.5 USE CASE DIAGRAM OF THE ACADEMIC (RESULTS) SECTION

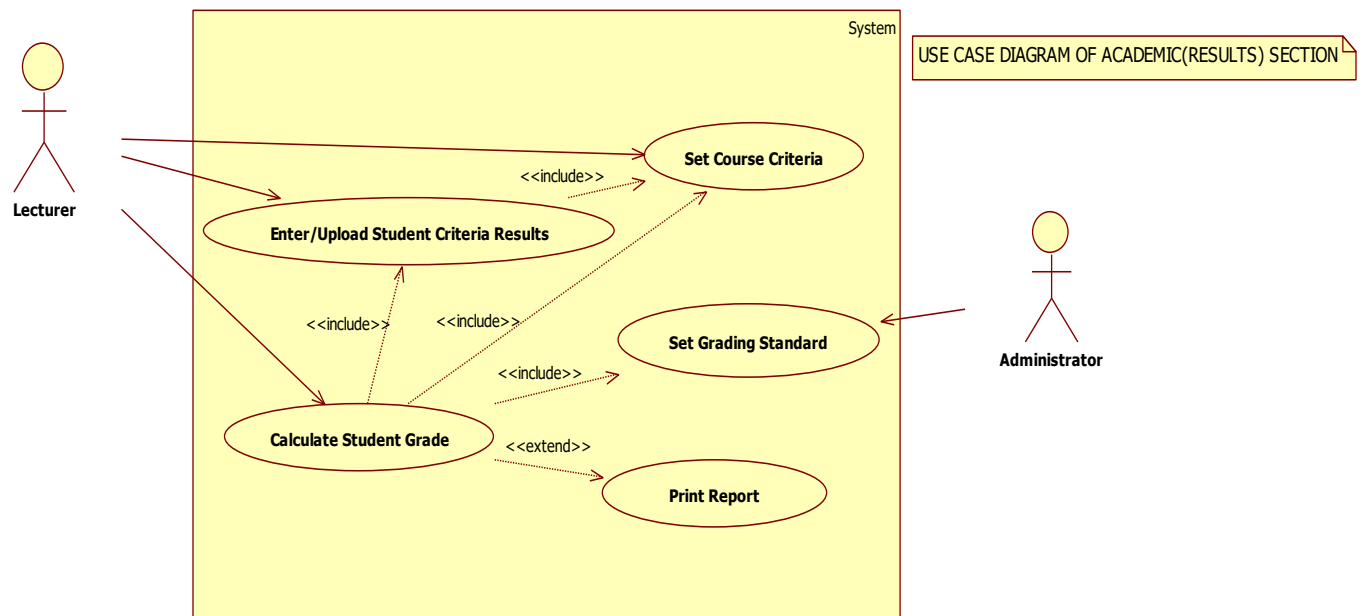


Figure 4.4 Use case diagram for the academic section

4.4.6 USE CASE DIAGRAM OF THE TRANSACTION SECTION

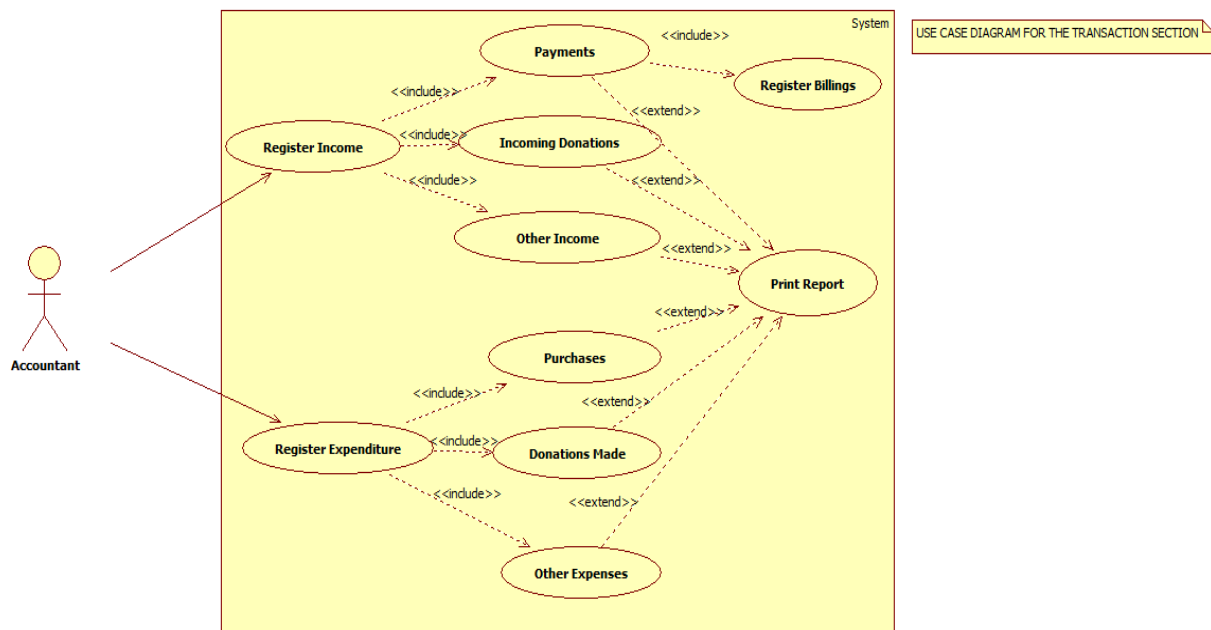


Figure 4.5 Use case diagram of transaction section

4.4.7 USE CASE-DIAGRAM OF THE INBUILT EVS SECTION

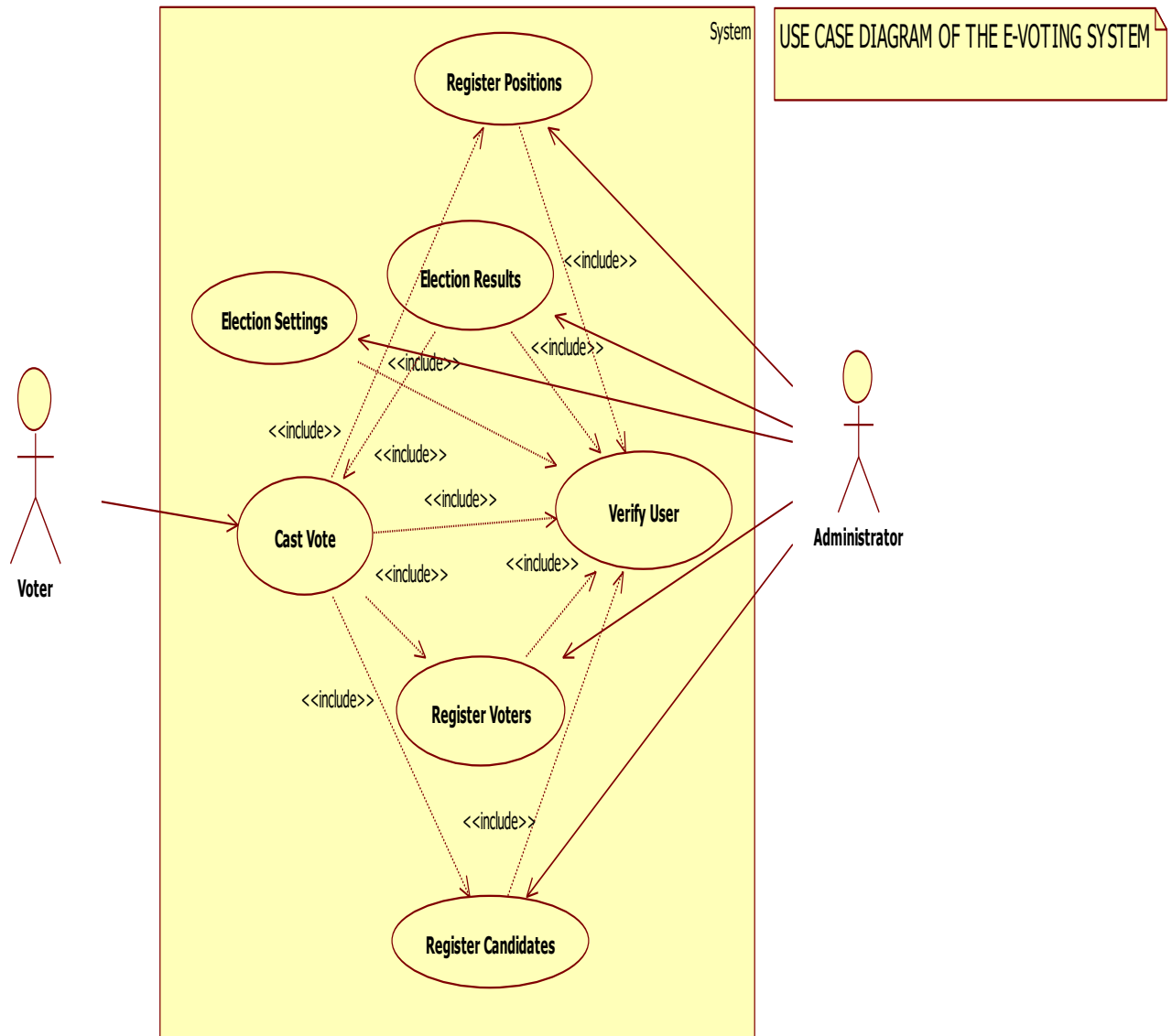


Figure 4.6 Use case diagram of the inbuilt Electronic voting system

4.5 ACTIVITY DIAGRAMS

In Unified Modeling Language (UML), an activity diagram is a graphical representation of an executed set of procedural system activities and considered a state chart diagram variation. Activity diagrams describe parallel and conditional activities, use cases and system functions at a detailed level.

An activity diagram is used to model a large activity's sequential work flow by focusing on action sequences and respective action initiating conditions. The state of an activity relates to the performance of each workflow step. (Janssen, 2016)

The activity diagram consists of the starting point of the process represented by a solid circle and transition arrows representing the flow or transition from one activity to the next. Rounded rectangles represent the activities, and a bull's eye circle represents the ending point of the process.

The following activity diagrams indicate the flow of operations that need to be performed in various section of the software.

4.5.1 SYSTEM LOGIN ACTIVITY DIAGRAM

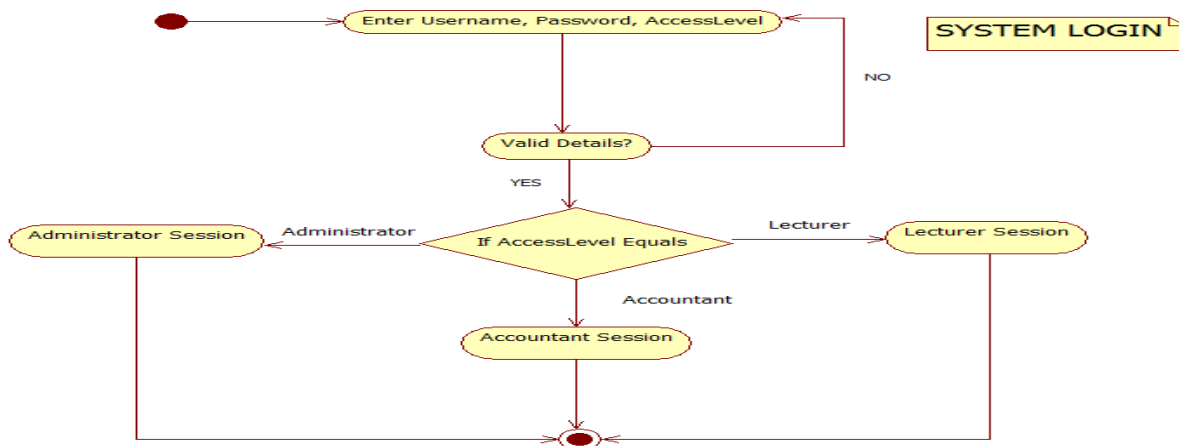


Figure 4.7 System Login activity diagram

4.5.2 STUDENT REGISTRATION ACTIVITY DIAGRAM

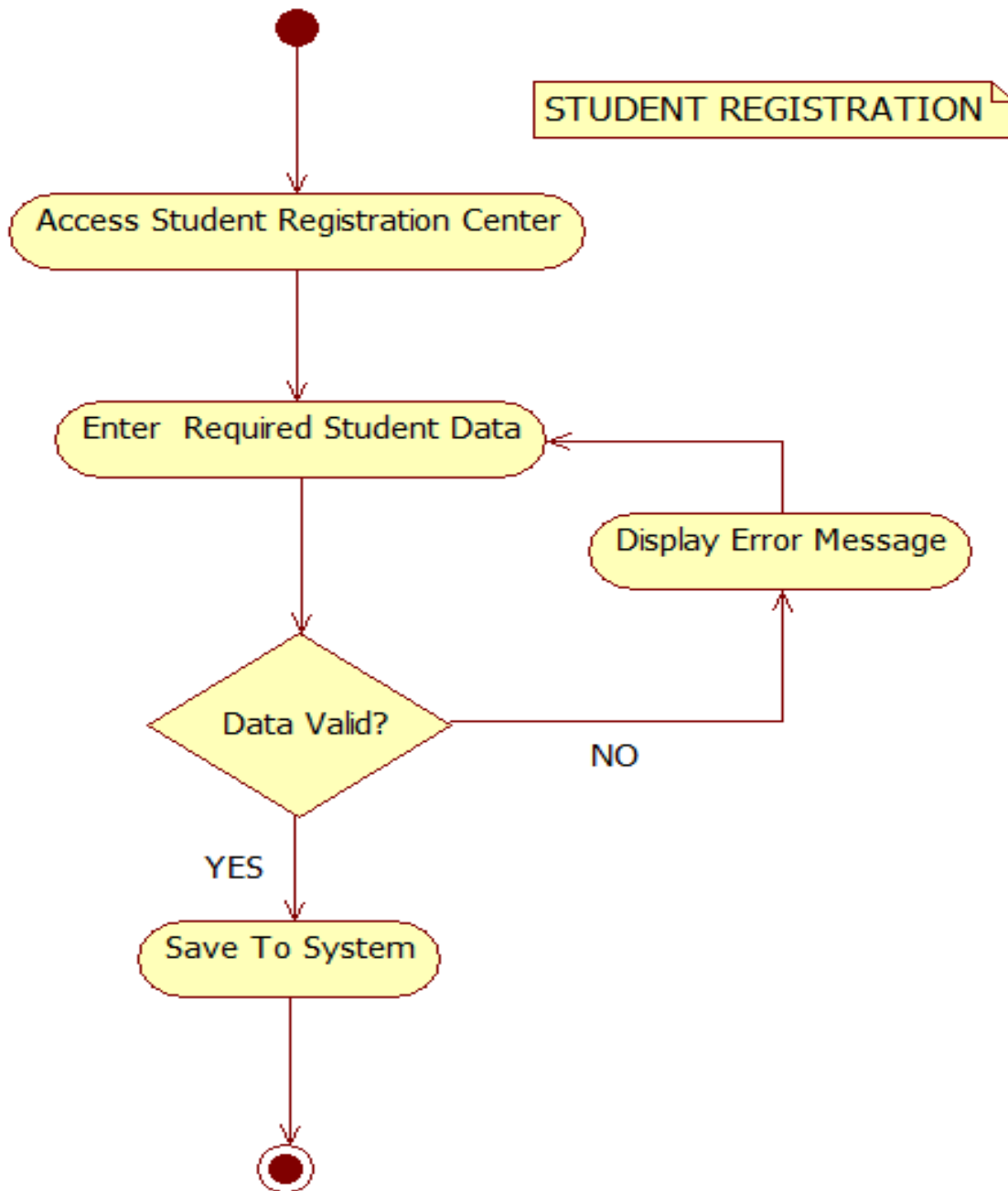


Figure 4.8 Student Registration Activity Diagram

4.5.3 PROGRAMME-COURSE ASSIGNMENT ACTIVITY DIAGRAM

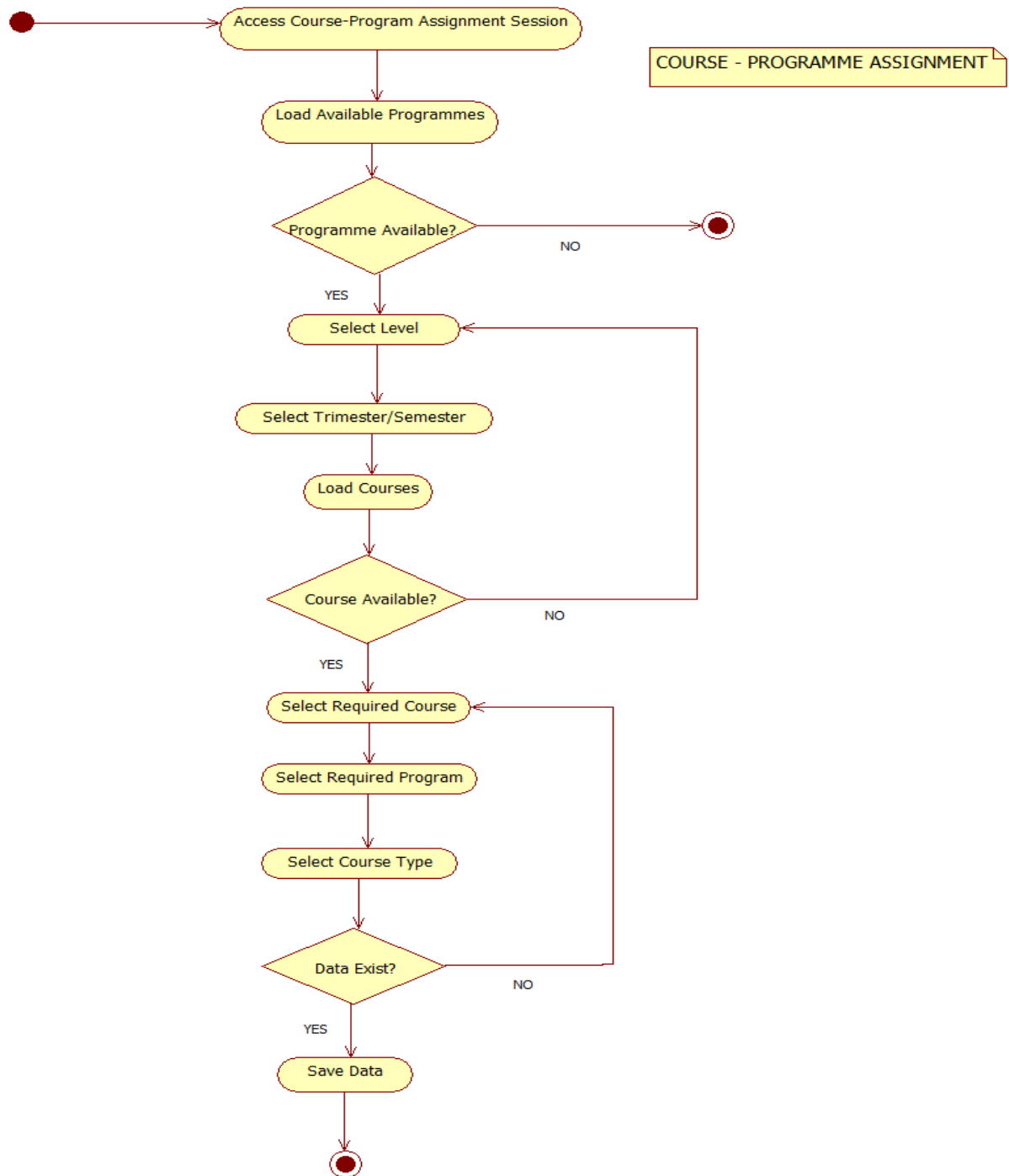


Figure 4.9 Programme-Course assignment activity diagram

4.5.4 COURSE-LECTURER ASSIGNMENT ACTIVITY DIAGRAM

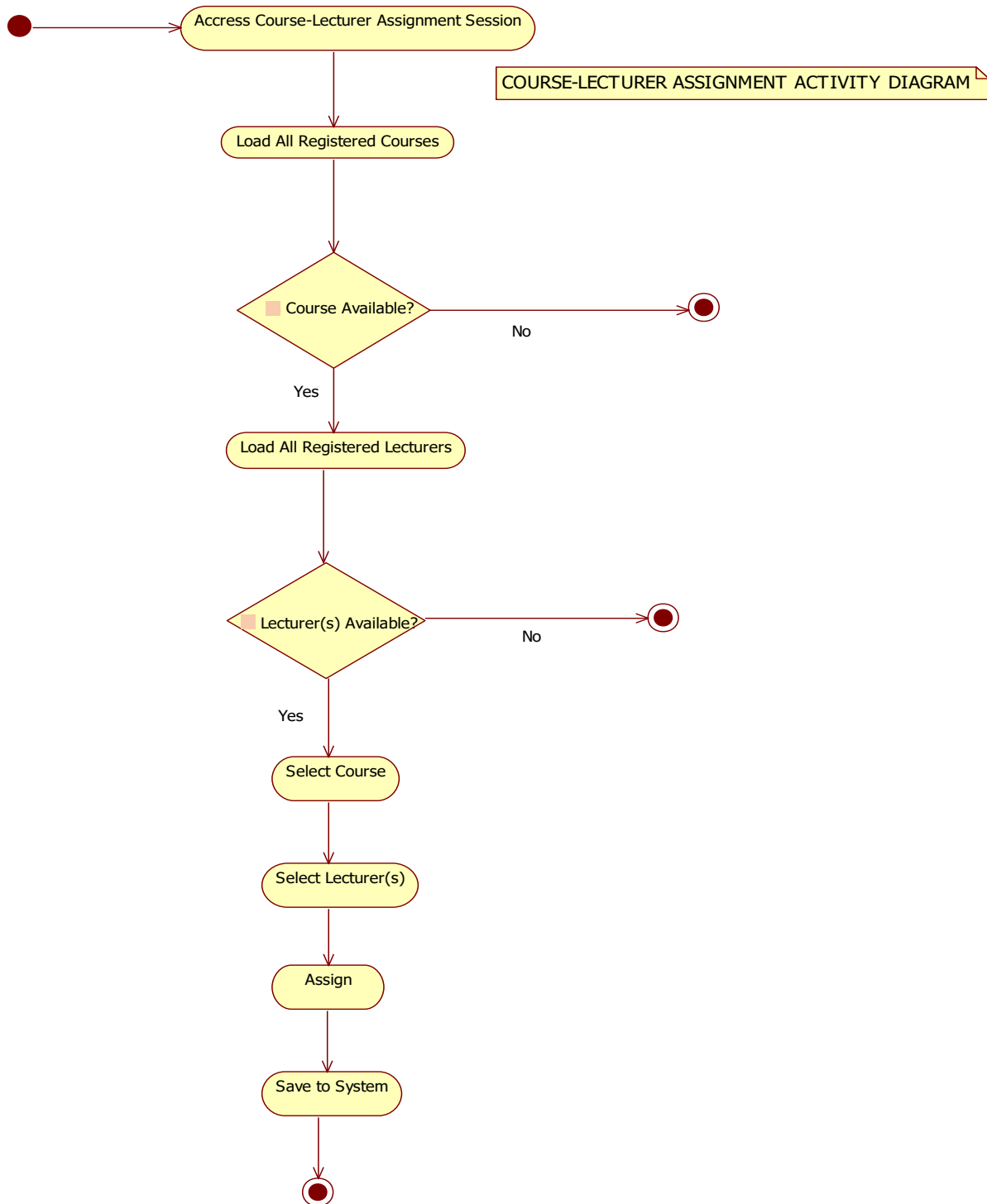


Figure 4.10 Course-Lecturer activity diagram

4.6 USER INTERFACE DESIGN

The system is made up of several interface (seventy-nine (79) forms) designs that handle different task. The graphical user interface (GUI) for the departmental management system is very user friendly and does not require any special training to learn how to use the software. Below are images and description of the graphical user interface (GUI).

4.6.1 The Splash Screens

There are two main splash screens involved in this software. The main splash screen which starts when the application first loads and other splash screens which starts during system transition such as voting section transition and entry into the E-voting system section.

Splash screens were used during the development of the system in order to attract the user attention to special entry into the system and other sections of the system keeps long when loading. The following diagrams shows various splash screens involved in the system.



Figure 4.11 The main splash screen

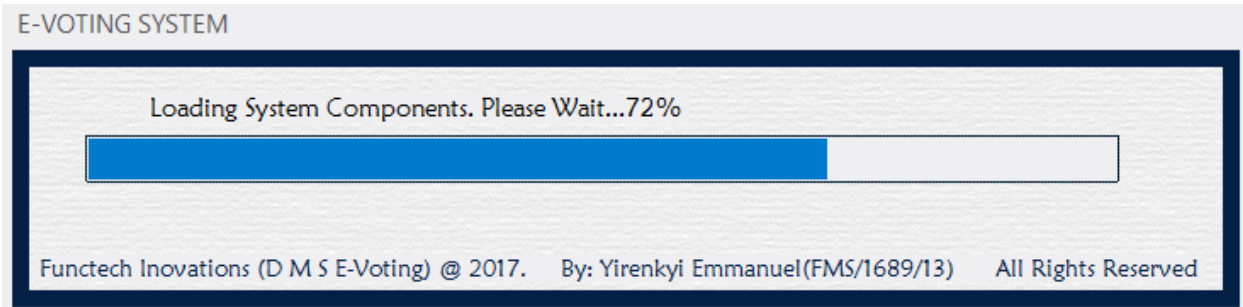


Figure 4.12 A splash screen indicating transition into the E-voting system section

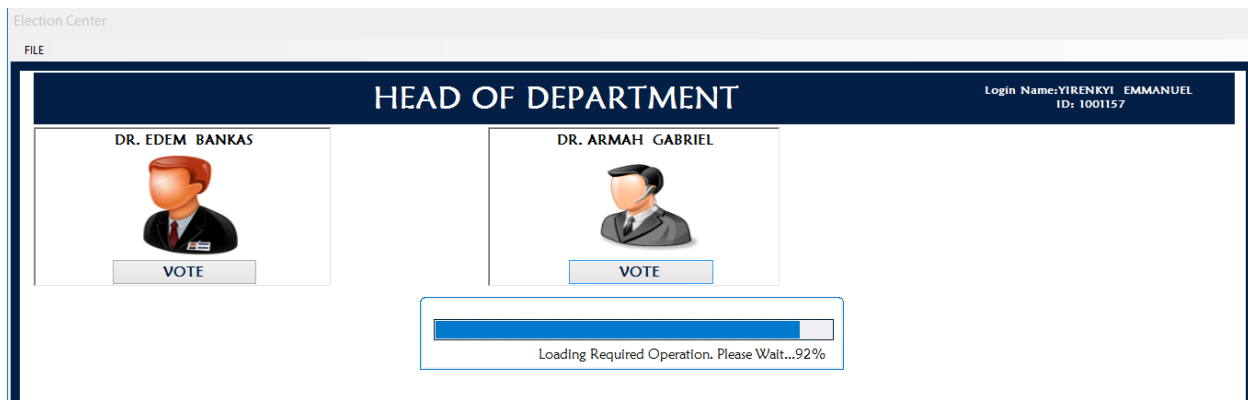


Figure 4.13 A splash screen indicating transition into a new section after a vote has been cast

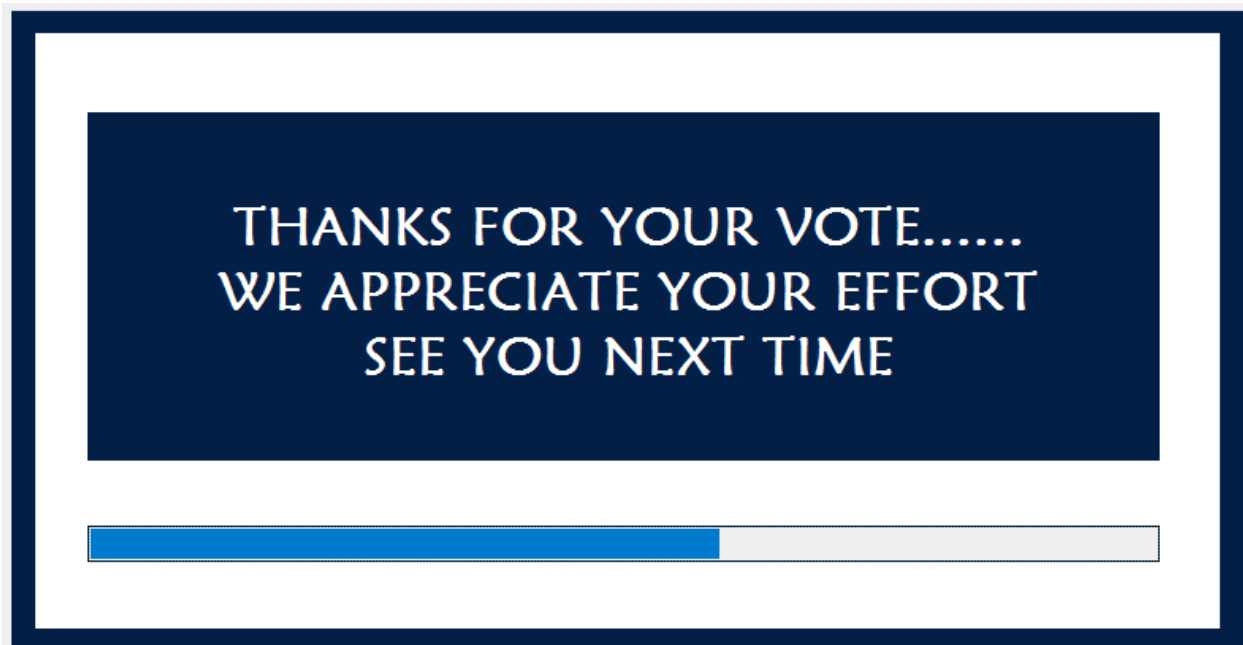


Figure 4.14 A splash screen showing a thank-you message after using the voting section

4.6.2 The Login Interfaces

There are two main login interfaces involved in this software. The main login interface which authenticate and authorizes the user into the application and the E-voting login interfaces which also authenticate and authorizes the user into the E-voting section.

The main login interface loads after the main splash screen has finish loading. With this interface, the user has to enter his or her username, password and select his access level. After which he or she will click a login button. The system then checks if the information entered are valid. If so, the system displays a successful message else it displays an error message. The user also has an option to quite the application under this section. The main login section authenticate and authorize the user into the main system section.

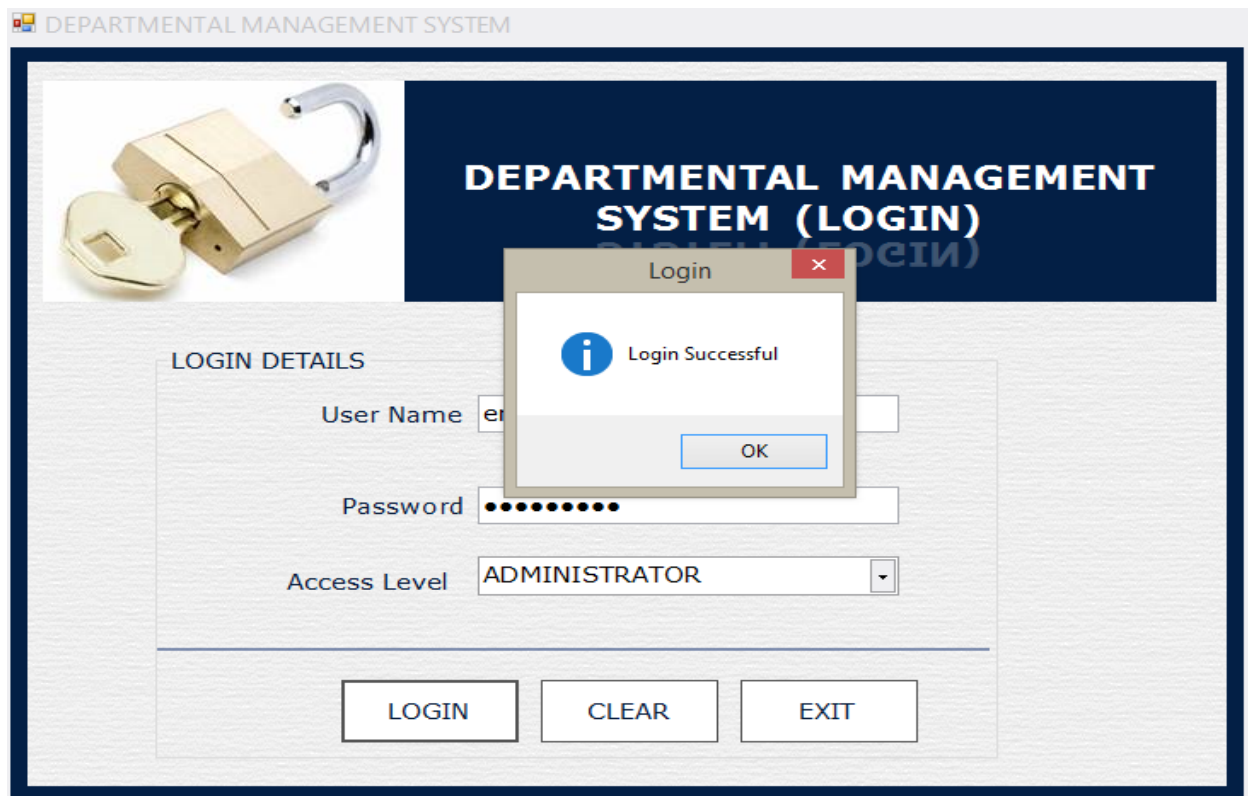
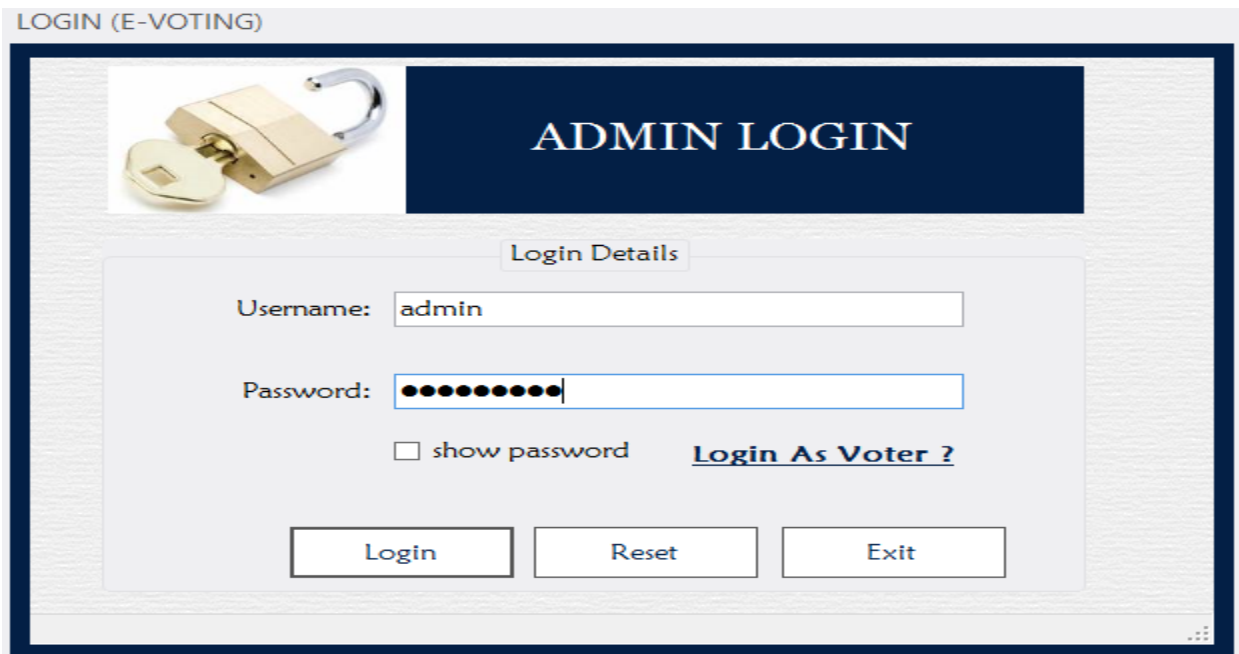


Figure 4.15 The main login section.

The E-Voting system login section also consist of three login sections. The administrative login section and entry code section which authenticate and authorize the user to perform administrative task in the E-voting system section and the voter login section which also validate the user to cast his or her vote.

With the administrative login section, the user has to provide his username and password which helps to validate him or her into the administrative interface to begin his or her task. With this interface the user can click on the “Login As Voter ?” link in order to login as a voter if the person is a voter which will help him /her access the voter login section.



LOGIN (E-VOTING)

ADMIN LOGIN

Login Details

Username: admin

Password: ●●●●●●●●

☐ show password [Login As Voter ?](#)

Login Reset Exit

Figure 4.16 The E-Voting administrative login section

After a user login as an administrator in the E-Voting section, he has access to the administrative interface but does not have access to perform administrative task unless he enters an entry code which will validate him to perform administrative task such as registering voters, registering candidates, determining election results and others.

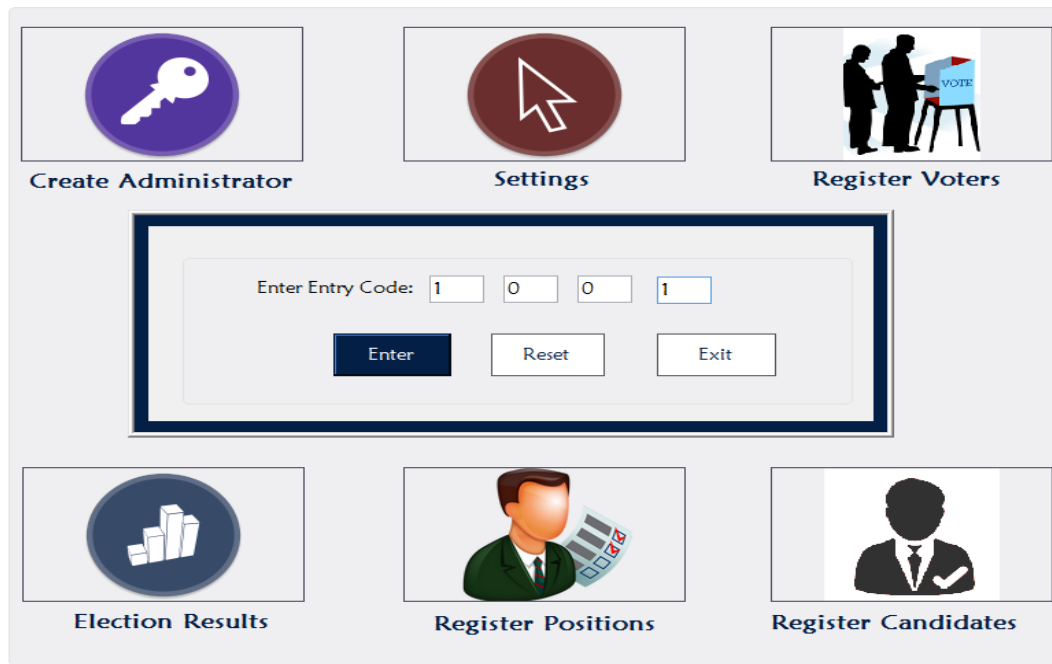


Figure 4.17 The Entry code login section

With the voter login section, the voter has to enter his or her voter ID number which will be generated by the system which will be given to the voter to help him or her cast his or her vote. This number will validate the user into the voting section. Once a voter ID is used, it becomes inactive and cannot be used again. The GUI below shows the Voter login section.

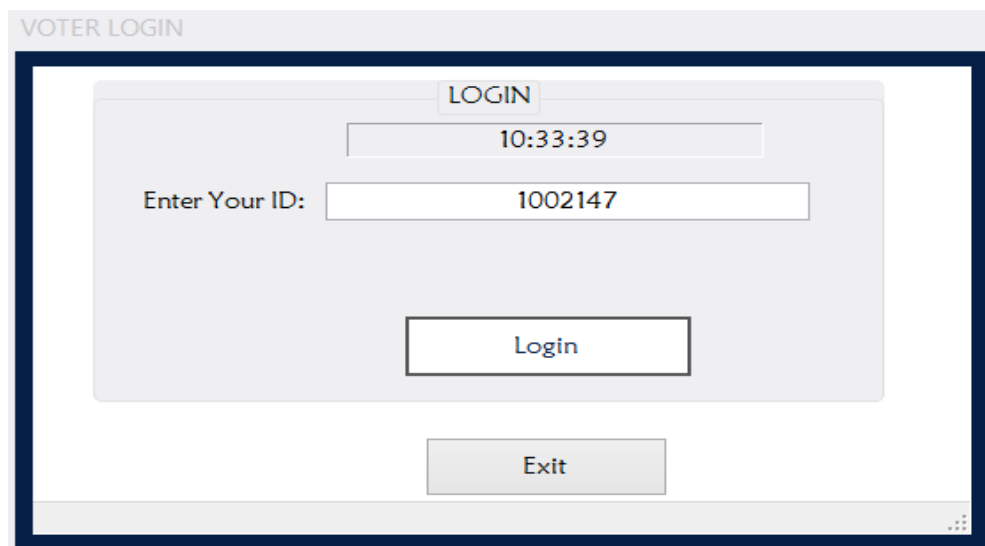


Figure 4.18 The Voter login section

4.6.3 The Main Interfaces

The system has two main interfaces: The overall interface for the system which initially shows after the main login to the system and the E-voting main interface system for the E-voting section which shows after administrative login into the E-voting section. Main interfaces were used in this application to help house other interface sections when performing specific task. A main interface also helps the user to know which aspect of the software he or she has access to. A main interfaces also reveals the components of the system.

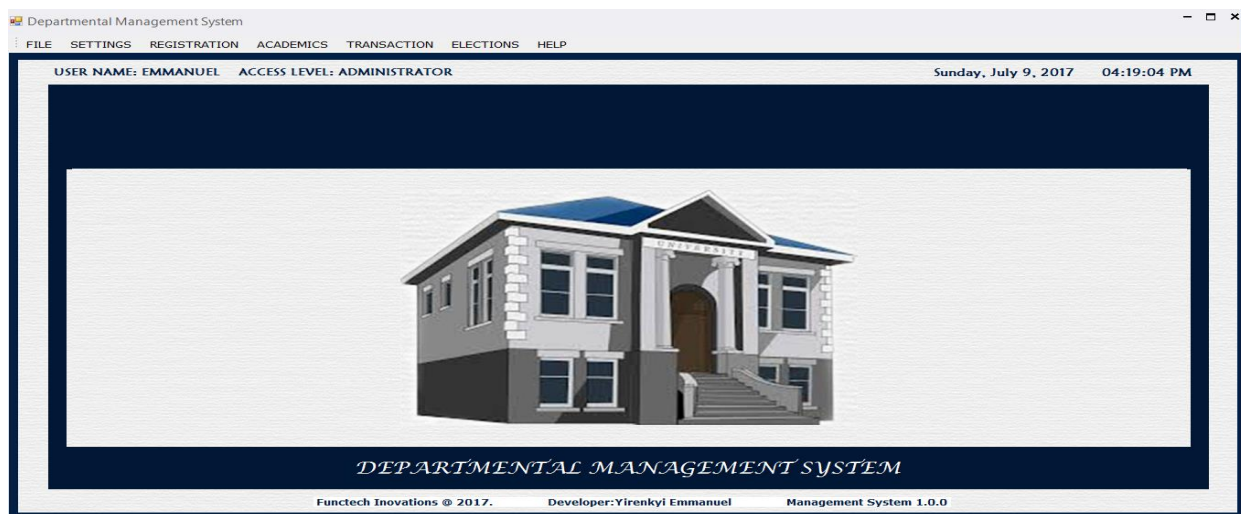


Figure 4.19 The overall main interface system section.

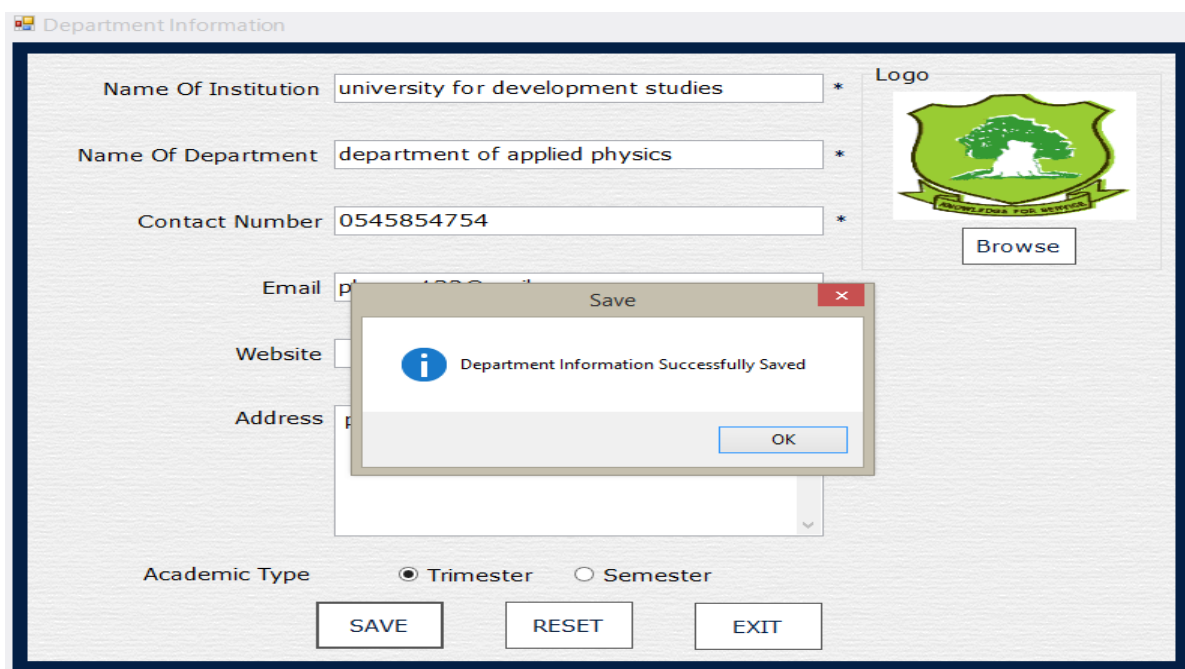


Figure 4.20 The E-voting main interface section.

.6.4 Some Registrations and Assignment Section of the software

❖ *Setting Department Information*

This section of the software helps to set the department information by setting the name of the institution and department, contact number, email, address, academic type and logo. This information is needed by the software as it helps in generating report heading and forms or controls customization. All required fields in this software is indicated



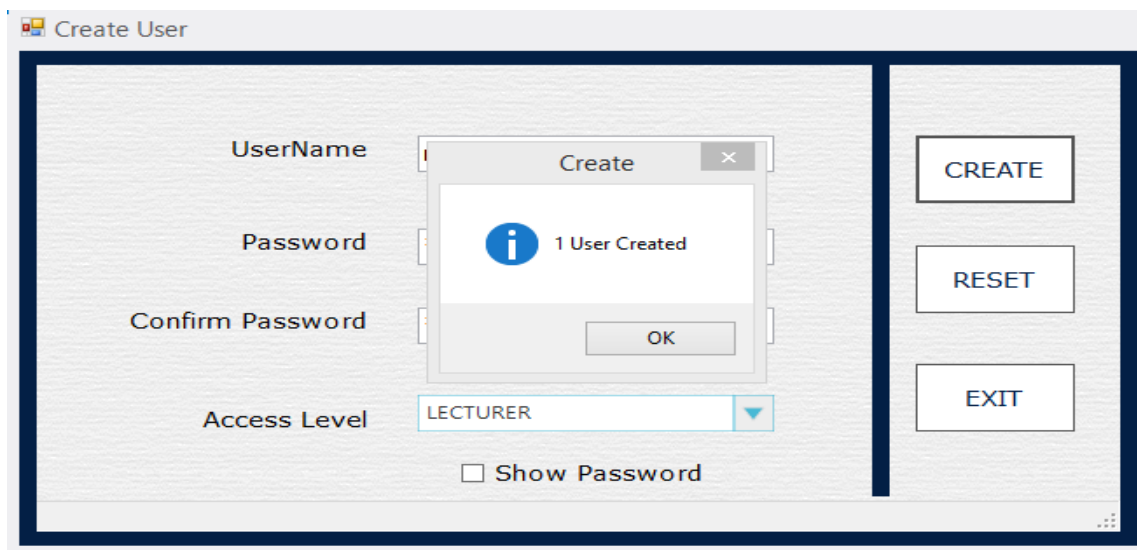
The screenshot displays the 'Department Information' window. It contains several text input fields: 'Name Of Institution' (filled with 'university for development studies'), 'Name Of Department' (filled with 'department of applied physics'), 'Contact Number' (filled with '0545854754'), 'Email', 'Website', and 'Address'. To the right, there is a 'Logo' section with a placeholder image of a green shield and a 'Browse' button. At the bottom, there are radio buttons for 'Academic Type' with 'Trimester' selected and 'Semester' unselected. Three buttons labeled 'SAVE', 'RESET', and 'EXIT' are at the bottom. A 'Save' dialog box is open in the center, showing an information icon and the text 'Department Information Successfully Saved' with an 'OK' button.

Figure 4.21 Setting department Information

❖ *Creating and Modifying Users*

Creating of users is done easily when using the system. In this section, the user's name, password and access level is required. This information is essential as it is used to log the user into the system and to authorize him/her. The user's name and password can be updated or changed by the user

under the manage settings section found under the file menu of the main section. The access level can also be modify by the administrator under the modify user section.

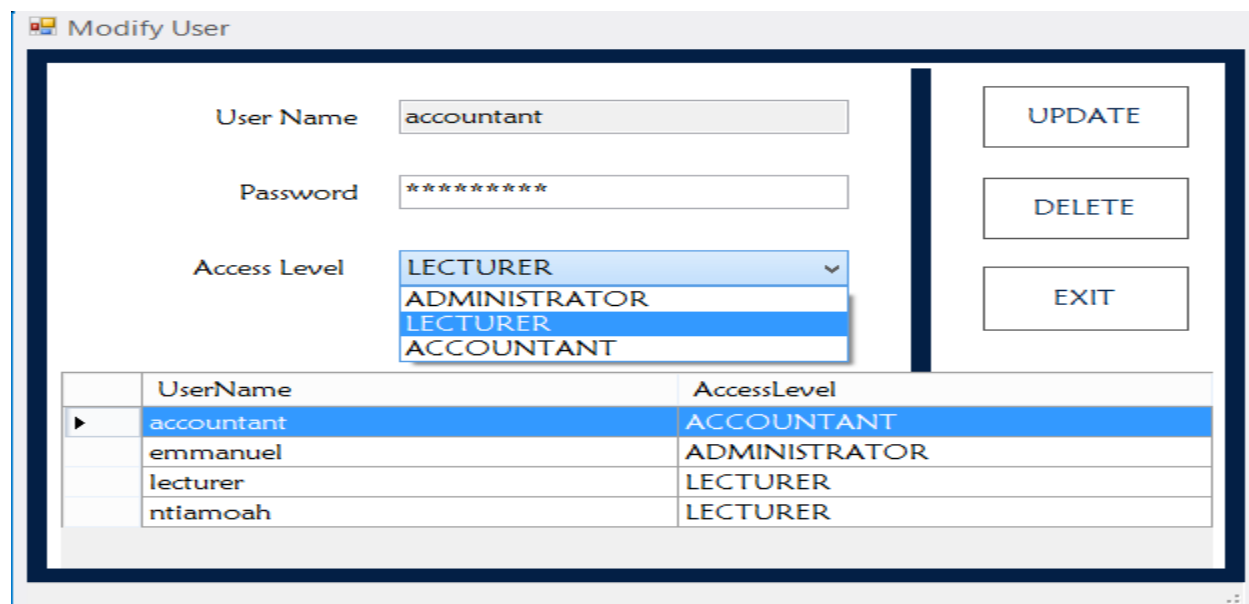


The 'Create User' window displays a form with the following fields and controls:

- UserName**: Text input field.
- Password**: Password input field.
- Confirm Password**: Password input field.
- Access Level**: Dropdown menu currently set to 'LECTURER'.
- ☐ **Show Password**: Checkbox to toggle password visibility.
- Buttons**: 'CREATE', 'RESET', and 'EXIT' buttons on the right side.

A modal dialog box titled 'Create' is open in the center, displaying an information icon and the message '1 User Created', with an 'OK' button at the bottom.

Figure 4.22 Create user section



The 'Modify User' window displays a form with the following fields and controls:

- User Name**: Text input field containing 'accountant'.
- Password**: Password input field containing '*****'.
- Access Level**: Dropdown menu currently set to 'LECTURER'.
- Buttons**: 'UPDATE', 'DELETE', and 'EXIT' buttons on the right side.

Below the form is a table listing existing users:

UserName	AccessLevel
accountant	ACCOUNTANT
emmanuel	ADMINISTRATOR
lecturer	LECTURER
ntiamoah	LECTURER

Figure 4.23 Modify user section

❖ *Registering Associations and their Leaders*

Registering associations and their respective leaders is done using the association information section. This information is needed when registering students.

Figure 4.24 Association Registration section

❖ *Student and Staff Registration Section*

The student registration section shows all the required information necessary to register a student under the department.

Figure 4.25 Student Registration section

The staff registration section shows all the required information needed to register a staff member.

The screenshot shows a web application window titled "STAFF REGISTRATION". It contains a search bar at the top with "Search By Staff ID:" and a "Search" button. Below this is a form for "Personal Information" with fields for Staff ID (STAFF-90001), First Name (gabriel), Other Name(s), Last Name (armah), Title (Dr), Gender (Male selected), Date Of Birth (13/Feb/1975), Contact Number (0545882736), Email (armah@gmail.com), and Address (p.o.box 554, Accra, Ghana). To the right is a "Profile Picture" section with a placeholder image and "Browse" and "Remove" buttons. Below that is a "Position" dropdown menu (Lecturer) and a "Bank Information" section with fields for Bank Name, Account Number, and Bank Branch. At the bottom are buttons for SAVE, UPDATE, DELETE, RESET, GET DATA, and EXIT. A "Save" dialog box is overlaid in the center, displaying an information icon, the text "1 Record Saved", and an "OK" button.

Figure 4.26 Staff Registration section

❖ Programme-CourseAssignment

The programme-course assignment section is used when assigning courses to various programmes offered by the department. Before this section can be used, programmes offered by the department need to be registered into the system using the programme registration section and various courses also need to be registered using the course registration section.

To use this section, one needs to select the required level and trimester or semester and Click on the load courses button. This then loads the available courses under the searching criteria. The interface below loads all level 100 first trimester programmes. The user must then select the required course code and choose a corresponding programme and then click on the register button to assign the course to the programme.

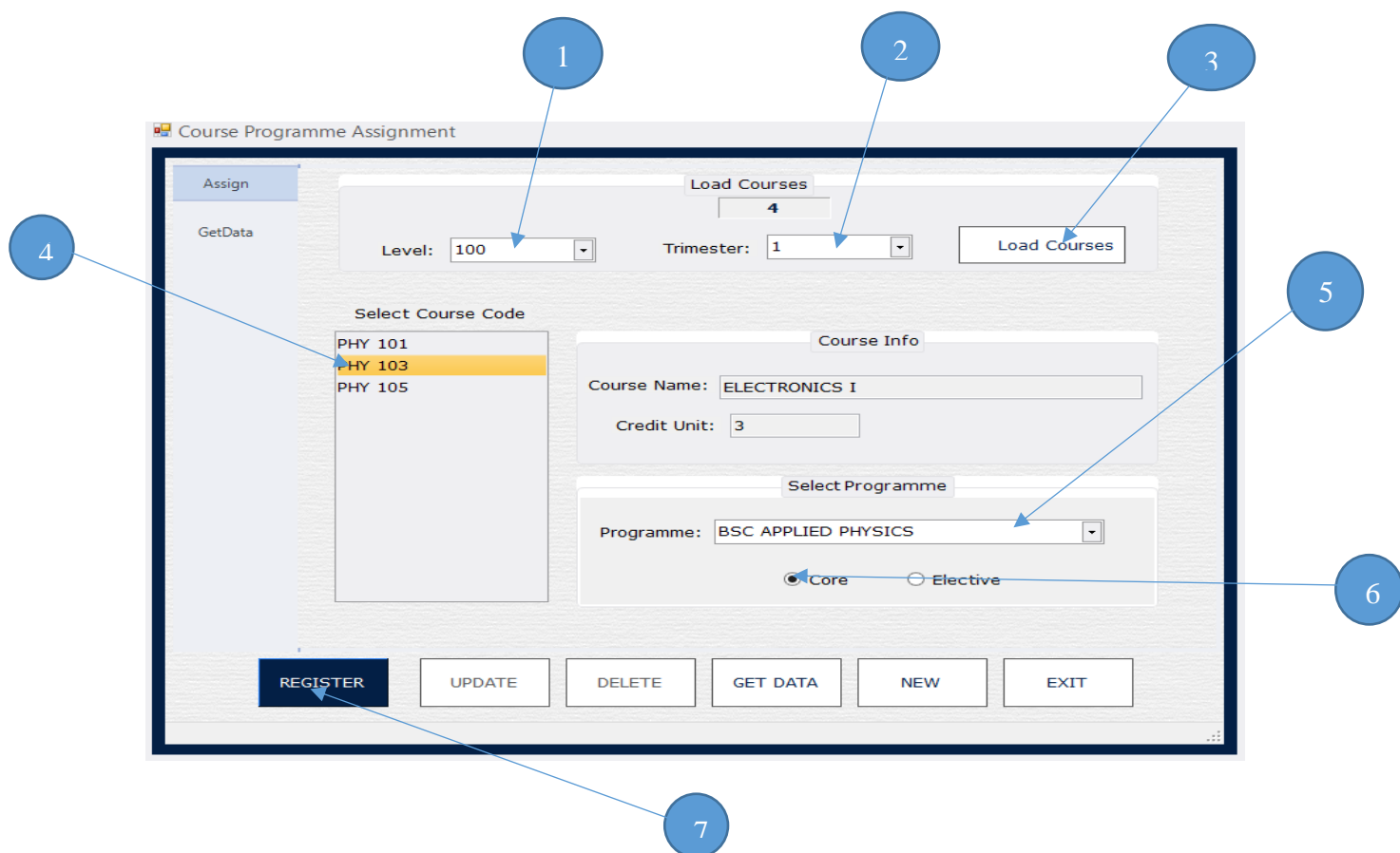


Figure 4.27 Course-Programme Assignment Section

KEYS

- | | |
|---|---------------------------|
| 1 | Select Level |
| 2 | Select Trimester/Semester |
| 3 | Load Course |
| 4 | Select Course |
| 5 | Select Programme |
| 6 | Select Course Type |
| 7 | Register |

❖ *Course-Lecturer Assignment Section*

Similar to the programme-course assignment section, the course-lecturer assignment section works only when there are registered courses and also registered lecturers. It then assist the user to assign courses to lecturers with ease. The following interface shows the various steps involves during a course-lecturer assignment.

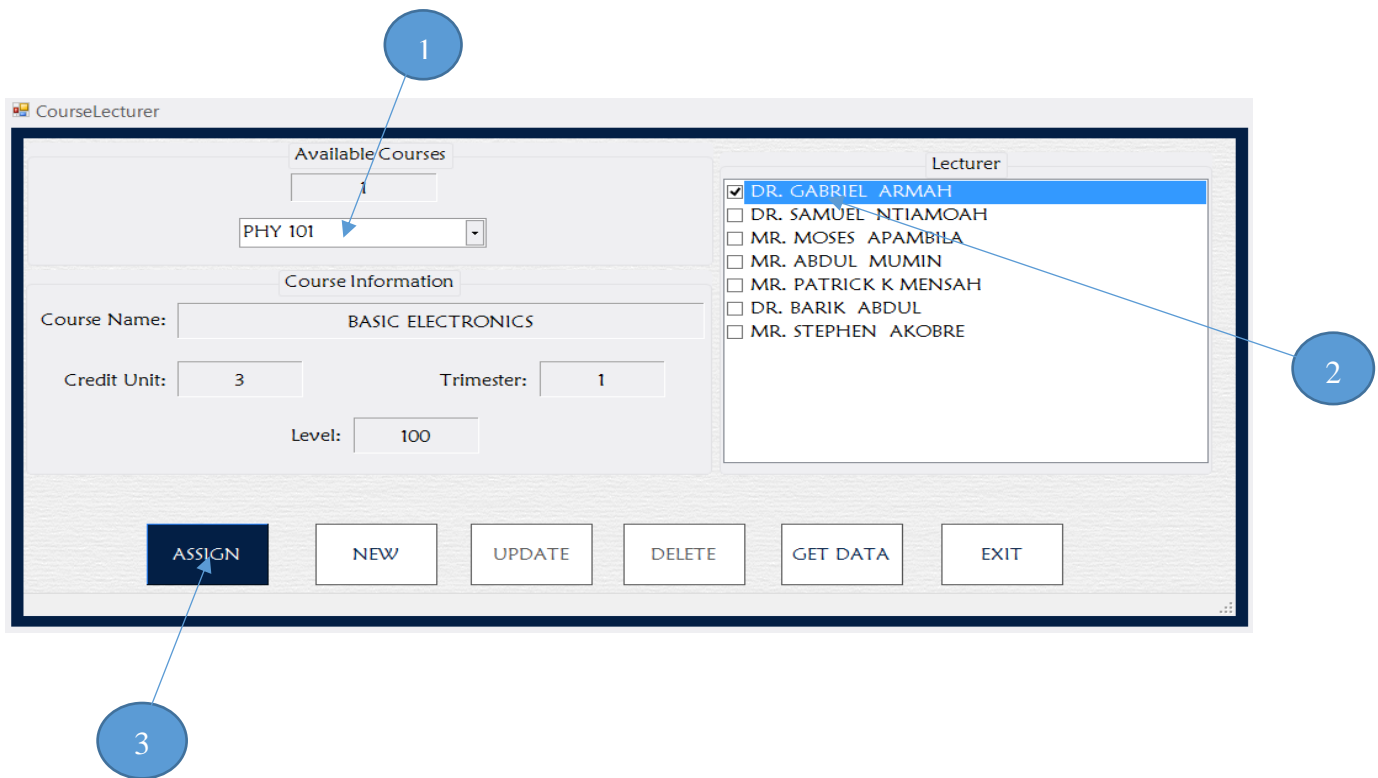


Figure 4.28 Course-Lecturer Assignment section

KEYS

- 1 Select Course
- 2 Select Lecturer
- 3 Assign

4.6.5 The Academic Section of the software

❖ *University Grade Mark Section*

The University Grade Mark Section helps to set the standard grading mark for the institution. Without setting the Grade mark in this section, student course results cannot be performed since the information in this section is needed to assign grade, grade point and remarks to a student total mark score in a particular course.

GRADING MARKS

SET UNIVERSITY GRADING MARKS

Record ID: 10

Score

Minimum Score: 80

Maximum Score: 100

Grading

Grade: A+

Grade Point: 5.0

Remarks: Excellent

SAVE NEW GET DATA EXIT

Figure 4.29 University Grade Mark Section

It must be noticed that the minimum and maximum score must follow a certain order. The minimum score entered must be the same as the last maximum score previously entered. Thus example of valid score range for the system is (0-40, 40-45, 45-50, 50-55, 55-60, 60-65, 65-70, 70-80, 80-100). No gap must be left between a maximum score range and the next maximum score range. Example of invalid score range is (0-39, 40-44, 45-49, 50-54 etc.)

For a valid score range such as the first example above, all scores between 0 and less than 40 will be assign the corresponding grade and grade point with remarks.

❖ *Course – Grading Settings Section*

With this section lecturers will set the course grading criteria for their courses. Five criteria are accepted by the software: Thus Final Examination, Mid Trimester or Semester Exams, Presentation, Practical and Assignment. In the figure below PHY 103 (Electronics I) will be graded with a total score of 120 with criteria of Final Exams, Mid Trimester Exam, Practical and Assignments.

The screenshot displays the 'Set Grading Criteria' application window. It is divided into two main sections: 'Course Information' and 'Assessment Criteria'. In the 'Course Information' section, the Course Code is 'phy 103', Course Name is 'ELECTRONICS I', Level is '100', and Trimester is '1'. The 'Assessment Criteria' section lists five criteria: 'Final Exams' (checked, 70), 'Mid Trim Exam' (checked, 30), 'Presentation' (unchecked), 'Practical' (checked, 15), and 'Assignments' (checked, 5). The total score is calculated as 120. A 'Save' dialog box is currently open, showing a message: 'Grading Criteria for PHY 103 has been set' with an 'OK' button. At the bottom of the application window are four buttons: 'SAVE', 'NEW', 'GET DATA', and 'EXIT'.

Figure 4.30 *Error! No text of specified style in document..1 Course-Grading Section*

❖ *Loading Student Scores*

Loading student score can be done with an individual record entry or through results upload. The diagram below shows how the practical results of the student is uploaded by using an already prepared excel sheet.

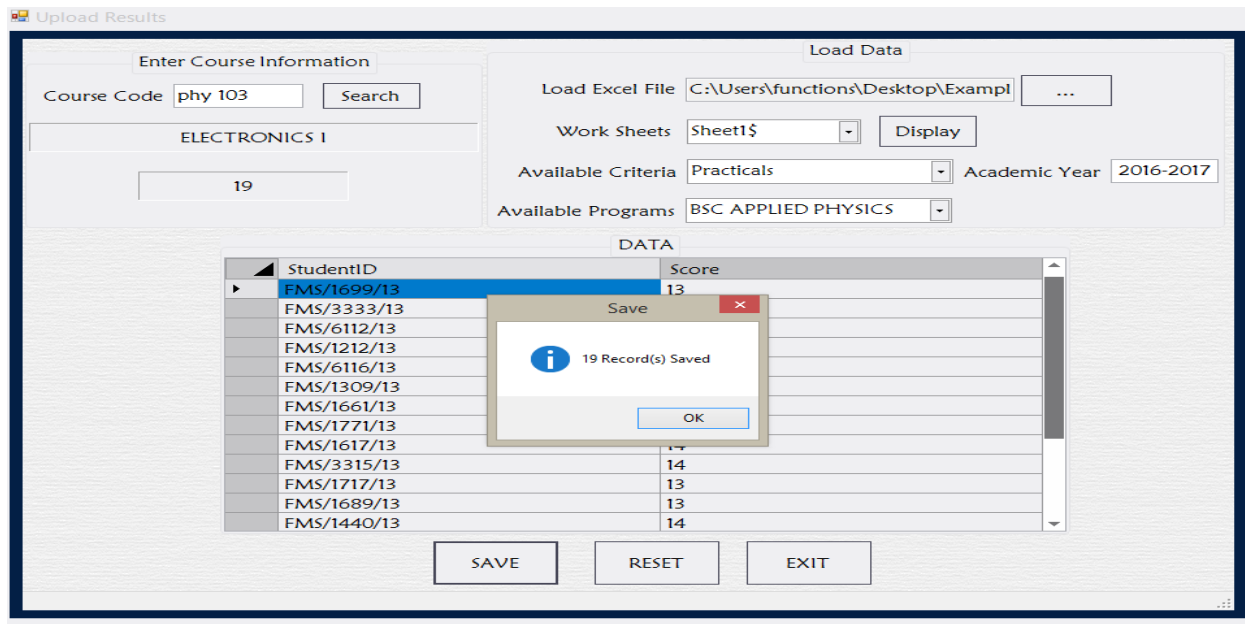


Figure 4.31 Load student score section

❖ Calculate Results Section

The calculate results section helps in calculating grade of student score. The lecturer (user) has to walk through three steps for the students' grade to be calculated based on the uploaded results. This section can also be used to identify if a student grade from a particular criteria is missing. In the example below for PHY 103, the total score column indicates the 'Total_Score' for each student after all assessment results has been combined. The "No_Of_Criteria" column indicates the number of criteria taken for the calculation for each student. If a value within this section is not up to the total criteria of four, then the student result is incomplete in this instance. This section is shown after the show total button has been clicked. To calculate the student grade, click on the "Calculate Grade and Save Records" button

frmCalculateResults

Course Info | Total Results | Total Marks

Academic Year: 2016-2017 Show Total Calculate Grade & Save Results

DATA

StudentID	Total_Score	NO_Of_Criteria	Programme	AcademicYear
FMS/0012/13	81	4	BSC APPLIED PHYSICS	2016-2017
FMS/1212/13	61	4	BSC APPLIED PHYSICS	2016-2017
FMS/1309/13	73	4	BSC APPLIED PHYSICS	2016-2017
FMS/1356/13	82	4	BSC APPLIED PHYSICS	2016-2017
FMS/1440/13	76	4	BSC APPLIED PHYSICS	2016-2017
FMS/1617/13	82	4	BSC APPLIED PHYSICS	2016-2017
FMS/1661/13	92	4	BSC APPLIED PHYSICS	2016-2017
FMS/1689/13	108	4	BSC APPLIED PHYSICS	2016-2017
FMS/1699/13	100	4	BSC APPLIED PHYSICS	2016-2017
FMS/1714/15	72	4	BSC APPLIED PHYSICS	2016-2017
FMS/1717/13	83	4	BSC APPLIED PHYSICS	2016-2017
FMS/1771/13	83	4	BSC APPLIED PHYSICS	2016-2017
FMS/1789/13	82	4	BSC APPLIED PHYSICS	2016-2017
FMS/1818/13	56	4	BSC APPLIED PHYSICS	2016-2017
FMS/3315/13	88	4	BSC APPLIED PHYSICS	2016-2017
FMS/3333/13	82	4	BSC APPLIED PHYSICS	2016-2017
FMS/6112/13	62	4	BSC APPLIED PHYSICS	2016-2017
FMS/6116/13	60	4	BSC APPLIED PHYSICS	2016-2017

<<BACK NEXT>> CLEAR EXIT

Figure 4.32 Student Total Score in calculate result section

frmCalculateResults

Course Info | Total Results | Total Marks

Search By

Student ID: PHY 103 19 Record(s) Found

Search Delete All Modify Print Delete Final Student Grade

RESULTS

StudentID	Score	Grade	GradePoint	Remarks	Program
FMS/0012/13	67.5	B+	4	Good	BSC APPLIED PHYSICS
FMS/1212/13	50.8	C	2.5	Satisfactory	BSC APPLIED PHYSICS
FMS/1309/13	60.8	B	3.5	Above Average	BSC APPLIED PHYSICS
FMS/1356/13	68.3	B+	4	Good	BSC APPLIED PHYSICS
FMS/1440/13	63.3	B	3.5	Above Average	BSC APPLIED PHYSICS
FMS/1617/13	68.3	B+	4	Good	BSC APPLIED PHYSICS
FMS/1661/13	76.7	A	4.5	Very Good	BSC APPLIED PHYSICS
FMS/1689/13	90	A+	5	Excellent	BSC APPLIED PHYSICS
FMS/1699/13	83.3	A+	5	Excellent	BSC APPLIED PHYSICS
FMS/1714/15	60	B	3.5	Above Average	BSC APPLIED PHYSICS
FMS/1717/13	69.2	B+	4	Good	BSC APPLIED PHYSICS
FMS/1771/13	69.2	B+	4	Good	BSC APPLIED PHYSICS
FMS/1789/13	68.3	B+	4	Good	BSC APPLIED PHYSICS
FMS/1818/13	46.7	D+	2	Pass	BSC APPLIED PHYSICS
FMS/3315/13	73.3	A	4.5	Very Good	BSC APPLIED PHYSICS

<<BACK NEXT>> CLEAR EXIT

Figure 4.33 Students Grade in calculate result section

4.6.6 THE TRANSACTION SECTION INTERFACES

Dues and Levies Settings

Billing Information

4

Billing Name: member dues *

Amount: 40 * Discount (%): 5

Level Billing: ☒ All Staff ☐ Lecturers ☐ Non-Lecturers

Buttons: Save, New, Delete, Delete All, Update, Exit

Billing Information Successfully Saved

OK

RecordID	Billing Name	Level Billing
1	LECTURERS	LECTURERS
3	NON LECTURERS	NON LECTURERS

Figure 4.34 Registering Billing information

Payments

Dues and Levies and Information

Level Billing: ☒ All Staff ☐ Lecturers ☐ Non-Lecturers

Select Dues: MEMBER DUES

Discount (%): 5 40

Grand Total: 38

Payment Date: 04/ Jan

Payment Mode: CASH

Amount Paid: 50

Balance: 12

Payment Record Successfully Saved

OK

Member Information

5

Staff ID: staff-90001 Search

DR GABRIEL ARMAH

Buttons: Save & Print, Get Data, Save Only, Print, Reset, Update, Close, Delete

Figure 4.35 Dues and levies payments section under income

4.6.7 THE E-VOTING SYSTEM SECTION

Before a voting can commence, certain settings have to be made by the administrator in the system. This include registering election positions, registering candidates for the various positions, registering voters with their corresponding voter ID which can be generated by the system. The following interfaces shows the various scenes from the E-Voting system section. This section can only be managed by only one administrator. The election voting section is iterated based on the number of positions or portfolios registered.

CREATE ADMINISTRATOR

ADMIN INFOR

Username: admin

Password: ●●●●●●●●

Confirm Password: ●●●●●●●●

System Exit and Entry Code: 1 0 0 1

☐ Show Password

SAVE GET DATA RESET EXIT

Password Match

Figure 4.36 Creating administrator to manage election

Register Positions

FILE

Record ID: 2

Enter Position: secretary

SAVE UPDATE DELETE RESET EXIT

RecordID
1

Insert

1 Record Inserted

OK

Figure 4.37 Registering election positions

REGISTER CANDIDATES

Personal Information

RecordID:

First Name: *

Other Name(s):


Surname:

Gender: ☒ Male

Date of Birth:

Age:

Candidate Picture



Election Information

Position: *

Register


 1 Candidate Registered

Figure 4.38 Registering Election Candidates

REGISTER VOTERS

Entry Registration | Read From Excel

Excel File: ...

Work Sheets:

VoterID	FirstName	OtherName	LastName
1001107	JALILU		BAWA
1001122	ALBERT		ASORE
1001127	EMMANUEL		GYABENG
1001132	ABDUL-MUNIM	AWAL	MOHAMMED
1001142	BATACHEGA	DAVID	AYUU
1331108	WEREKO	SHADRACK K.	BUADU
1331121	HAFIZ		ABUBAKARI
1331126	MOHAMMED		INUSAH
1331134	JOE-PETER		MUMUNI
1331140	GIDEON		BADU
1661109	DANIEL		DOSU
1661122	YUSSIF	MOHAMMED M...	ACHULO

Figure 4.39 Uploading voters list

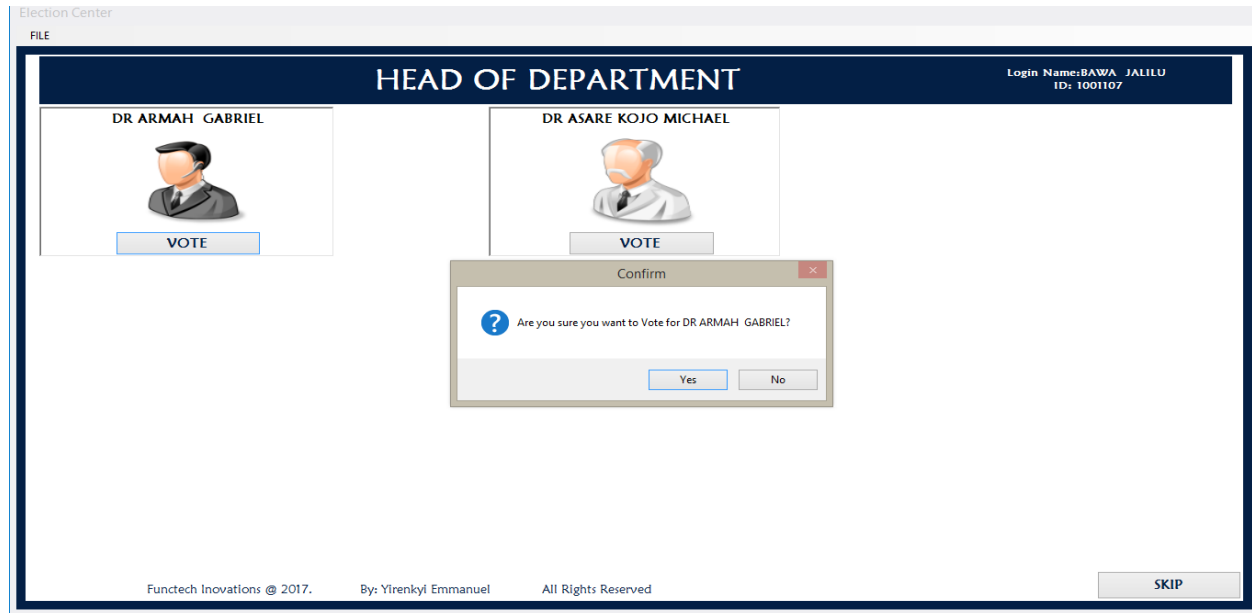


Figure 4.40 A voter Casting his vote



Figure 4.41 Election Results after voting.

CHAPTER FIVE

5.0 CONCLUSION

The project was developed using Visual Basic 2013 as a front-end tool and T-SQL as a back-end tool. These languages were chosen based on the analysis of the existing system, with flexibility for future enhancement.

The expanded functionality of today's software requires an appropriate approach towards software development. With this the iterative waterfall model was used during the development process.

The Departmental Management System is develop to manage and enhance department activities. The system helps in registration of students, staff members and associations.

The system allows the distribution of courses to lecturers in an easy manner.

The system also helps to compute students' grade ranging from assignment to final exams and also keep track of the results.

With our system, the department can monitor their financial status since the system tracks all income and expenditure.

Finally, the system includes an inbuilt electronic voting system that can be used to run elections by the department.

5.1 RECOMMENDATIONS

Though a lot of work has been done on the Departmental Management System to get to this far, I believe that there are several other functionalities that could be added to make it more efficient. However due to time constraint, it was not possible for this to be done. We therefore recommend that any future work on this project, we will strive to include the following:

- Integration of biometric system for students attendance checking.
- Integration of biometric functionality in the E-voting system.
- Sending of mails to students and parents after results have been approved.
- Managing the financial status and other affairs of associations within the department.

REFERENCES

1. Anne, P. (2002). *murach's beginnig visual basic .NET*. Paris: Mike Murach and Association.
2. Appleman, D. (2012). *Visual Basic Programmer's Guide to the Win32 API*. Washington DC: Sams.
3. Balena, F. (2013). *Programming Microsoft Visual Basic 2012*. Boston: Microsoft Press.
4. Bradley, J. C. (2010). *Programming in visual basic 2010*. Cape Town: McGraw-Hill/Irwin.
5. Carney, k. (2015). *Visual Basic .NET for Complete Begginners*. New York: Home and Learn.
6. Clark, D. (2013). *Bigginig C# Object-Oriented Programming*. New York: Todd Meister.
7. Comell, G. (2013). *Visual Basic 2012 From the Ground Up*. New York: Osbone Mcgraw Hill.
8. Culnan, M., & Swason, J. (2010). *Mapping the Intellectual Structre of MIS*. Chicago: Apress.
9. Danijel, A. (2014). *Professional Refactoring in Visual Basic*. Washington: Microsoft Press.
10. Deital, P., & Ayer, G. (2012). *Simplify Visual Basic 2012*. Washington D.C: Prentice Hall.
11. Farrell, J. (2012). *Visual basic programs to accompany programming logic and design 6th edition*. Boston: Cengage Learning.
12. Gaddis, T., & Irvine, K. (2014). *Starting Out With Visual Basic 2012*. Boston: Pearson Press.
13. Halverson, M. (2014). *Microsoft Visual Basic .NET Deluxe Learning Edition*. New York: Microsoft Press.
14. Halvorson, M. (2013). *Learn Microsoft Visual Basic 2012*. New York: Microsoft Press.
15. Halvorson, M. (2013). *Microsoft Visual Basic 2013 Step By Step*. California: O'Reilly Midia,Inc.
16. Hollis, B. (2010). *Visual Basic 2010 and .NET 4*. Chicago: Wrox Press.
17. Irvine, k., & Gaddis, T. (2012). *Advanced Visual Basic 2010*. Boston: Person Press.
18. ISTQB, C. (2017). *Waterfall model, Advantages,Disadvantages and how to use it*. Retrieved from istqexamcertification web site: <http://istqbexamcertification.com>
19. James, R. (2010). *Coding Guidelines*. Chicago: CreateSpace Independent.
20. Janessen, M. (2016). *Activity Diagram*. Retrieved from technopedia.com: <http://www.techopedia.com/definition/27489/activity-diagram>
21. Juas, F. (2013). *Sams Tech Yourself Visual Basic 2012 in 24 Hours*. Sams.
22. Kauffman, J., Robbert, J., & Melhing, W. (2001). *Information Technology and Economic Performance*. California: Irvine Work Press.
23. Lucey, T. (2017). *Management Informtion System*. Retrieved from britannica.com: <http://www.www.britannica.com>

24. McManus, J. (2014). *Database Access With Visual Basic 2012*. Sams.
25. Morrissey, J. (2002). Bibliometric and bibliographic analysis in an era of electronic scholarly communication. In *Bibliometric and bibliographic analysis in an era of electronic scholarly communication*. (p. 149). Chicago: Pearson.
26. Newsome, B. (2012). *Beggining visual basic 2012*. Boston: wrox.
27. Nyamekye, G. S., & Afoakwah, P. E. (2014). *Computerised Student's Registration Submission and Report System with Lectures Profile Management System*.
28. Richard, B. (2014). *Beggining VB.Net (2nd Edition)*. Chicago: Wrox Press.
29. Rod, S. (2012). *Visual Basic 2012 Programme's reference*. New York: wrox.
30. Rouse, M. (2015). *UML use case diagram*. Retrieved from techtarget.com: <http://whatis.techtarget.com/definition/use-case-diagram>
31. Scheider, I. (2012). *Introduction to Programming Using visual basic (10th Edition)*. Sams.
32. Sheldon, B. (2013). *Visual Basic 2012 and .NET 4.5 Programming*. Chicago: Microsoft Press.
33. Stermann, P. (2014). *University Management System*. Retrieved from whatis.techtarget.com: <http://whatis.techtarget.com/definition/Management System>
34. Treloar, A. (2015). *Developing an Information Management Strategy for University*. Chicago: AustralAsia Press.
35. Vogel, P. (2013). *Visual Basic Object & Component HandBook*. New York: Todd Meister.
36. Warhol, M. (2010). *The Art of programming with visual Basic*. Chicago: Sams.
37. Weber, B. (2002). In *Information Systems as a Reference Discipline* (p. 144). New York: Pearson Education Press.
38. WebSphere, A. (2009, March 19). *Software development process*. Retrieved from websphere-automation.blogspot.com: <http://websphere-automation.blogspot.com>
39. Whyte, L. (2008). *Managing Information with IT*. Retrieved from misinfo.com: <http://www.misinfo.com>

APPENDIX

CODING AND IMPLEMENTATION

This section deals with how the system was implemented using the visual basic 2013 programming language.

The Login Form

```
Imports System.Data
```

```
Imports System.Data.SqlClient
```

```
Public Class frmLogin
```

```
Private Sub btnExit_Click(sender As Object, e As EventArgs) Handles btnExit.Click  
    Me.Close()  
End Sub
```

```
Private Sub btnReset_Click(sender As Object, e As EventArgs) Handles btnReset.Click  
    Reset()  
End Sub
```

```
'Reset the controls
```

```
Sub Reset()  
    'Reset all the controls  
    txtPassword.Clear()  
    txtUsername.Clear()  
    txtUsername.Focus()  
    ErrorProvider1.Clear()  
End Sub
```

```
Private Sub btnLogin_Click(sender As Object, e As EventArgs) Handles btnLogin.Click  
    ErrorProvider1.Clear()  
    If isValid() Then  
        Dim strUserName As String = txtUsername.Text.Trim
```

```

Dim Password As String = txtPassword.Text.Trim
Dim strpassword As String = ENCRYPT_PASSWORD(Password)
Dim strAccessLevel As String = cboAccessLevel.Text.Trim

Dim strSelect As String = "SELECT * FROM User_Registration WHERE
UserName='" & strUserName & "'"

Dim mySqlSelect As New SqlCommand(strSelect, myConnectionString)
Dim mySqlDataReader As SqlDataReader = Nothing

Try
    If myConnectionString.State = ConnectionState.Closed Then
myConnectionString.Open()
        mySqlDataReader = mySqlSelect.ExecuteReader
        If mySqlDataReader.Read Then
            'The username exist so Check password
            Dim systemPassword As String = mySqlDataReader.Item("Password")
            If strpassword = systemPassword Then
                'Username and password Exist so Check AccessLevel
                Dim systemAccessLevel As String =
mySqlDataReader.Item("AccessLevel")
                If strAccessLevel = systemAccessLevel Then
                    strAccessLogin = strAccessLevel
                    strAccessUsername = strUserName.ToUpper
                    MessageBox.Show("Login Successful", "Login",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                    Reset()
                    blnClosingLoginFrm = True
                    Form1.Show()
                    Me.Close()
                    '#####
                Else
                    MessageBox.Show("Invalid AccessLevel", "Note",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                    mySqlDataReader.Close()
                End If
            End If
        End If
    End Try

```

```

        End If

    Else

        'Invalid password so prompt the user
        MessageBox.Show("Invalid Password", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

        mySqlDataReader.Close()

    End If

    mySqlDataReader.Close()

Else

    mySqlDataReader.Close()

    'Username does not exist so prompt user
    MessageBox.Show("Invalid User name", "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End If

Catch ex As Exception

    MessageBox.Show("An Error Occured", "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Exclamation)

Finally

    myConnectionString.Close()

End Try

End If

End Sub

'A function to validate the controls
Function isValid() As Boolean

    If txtUsername.Text.Trim = String.Empty Then

        ErrorProvider1.SetError(txtUsername, "User name is required")

        txtUsername.Focus()

        Return False

    End If

    If txtPassword.Text.Trim = String.Empty Then

```

```

        ErrorProvider1.SetError(txtPassword, "Password required")
        txtPassword.Focus()
        Return False
    End If

    If cboAccessLevel.Text.Trim = String.Empty Then
        ErrorProvider1.SetError(cboAccessLevel, "Access Level is required")
        cboAccessLevel.Focus()
        Return False
    End If

    Return True
End Function

Private Sub frmLogin_FormClosing(sender As Object, e As FormClosingEventArgs)
    Handles MyBase.FormClosing
        If blnClosingLoginFrm = False Then
            If MessageBox.Show("Are you sure you want to quit this applicaton?",
                "Confirm", MessageBoxButtons.YesNo, MessageBoxIcon.Question) =
                Windows.Forms.DialogResult.Yes Then
                e.Cancel = False
            Else
                e.Cancel = True
            End If
        End If
    End Sub
End Class

```