

**DESIGN AND IMPLEMENTATION OF AN INDUSTRIAL TRAINING
PLACEMENT SYSTEM FOR UNIBEN STUDENTS**

BY

GOD-DO-WELL AKPOMEROGHENEBO ONODJE

PSC1908925

**DEPARTMENT OF COMPUTER SCIENCE,
FACULTY OF PHYSICAL SCIENCES,
UNIVERSITY OF BENIN,
BENIN CITY,
EDO STATE, NIGERIA.**

NOVEMBER 2025

**DESIGN AND IMPLEMENTATION OF AN INDUSTRIAL TRAINING
PLACEMENT SYSTEM FOR UNIBEN STUDENTS**

BY

GOD-DO-WELL AKPOMEROGHENEBO ONODJE

PSC1908925

**A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF COMPUTER
SCIENCE, FACULTY OF PHYSICAL SCIENCES, UNIVERSITY OF BENIN, BENIN
CITY, EDO STATE, NIGERIA.**

**IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF A
BACHELOR OF SCIENCE (B.Sc.) DEGREE IN COMPUTER SCIENCE**

NOVEMBER 2025

CERTIFICATION

This is to certify that this project work was carried out by **GOD-DO-WELL**
AKPOMEROGHENEBO ONODJE with Matriculation Number **PSC1908925** under my
supervision. It is adequate and satisfactory, both in scope and content, for the award of Bachelor
of Science (B.sc) Degree in Computer Science of the University of Benin.

DR. D.N. IDEHEN

Project Supervisor

DATE

APPROVAL

This project work is hereby approved in partial fulfilment of the requirements for the award of Bachelor of Science (B.Sc.) Degree in Computer Science from the University of Benin.

DR. (MRS.) R.A. USIOBAIFO

Head of Department

DATE

DEDICATION

This project is dedicated to God Almighty, my source and strength who enabled me through this programme and to my wonderful family whose love has been a source of strength.

ACKNOWLEDGEMENT

My utmost acknowledgement goes to God Almighty for giving me the strength, wisdom and direction throughout my academic journey. I would like to express my gratitude to my project supervisor Dr. D.N. Idehen for his consistent guidance towards ensuring the successful completion of this project.

I would also like to specially thank Prof. (Mrs.) V.V.N. Akwukwuma (Dean Faculty of Computing), Dr. (Mrs.) R. A. Usiobaifo (Head of Department), the project coordinator Dr.M. Osagie, Mr. S.O.P. Oliomogbe (Retired) and other lecturers in the Department of Computer Science who I have been opportune to cross paths with, and have impacted me immensely these past few years: Prof. (Mrs.) F.A. Egbokhare, Prof. A. A. IMIANVAN, Prof. G. O. Ekuobase, Prof. (Mrs.) A. O. Egwali, Prof. F. I. Amadin, Prof. F. A. U. Imouokhome, Prof. (Mrs.) S. Konyeha, Prof. (Mrs.) V. I. Osubor, Prof. F. O. Chete, Dr. E. Nwelih, Dr. (Mrs.) G. O. Aziken, Mr. E. E. Obasohan, Dr. F. O. Oliha, Dr. (Mrs.) R. O. Osaseri, Dr. C. E. Igodan, Mr. K. O. Otokiti, Mr. I. E. Obayagbona, Miss. L. O. Usiosefe, Mr. J. O. Okhuoya, Mr. G. I. Evbuomwan and the non-teaching staff, for your various roles and support. I acknowledge your contributions with appreciation.

I will also like to acknowledge my friends, Baba Dee, Amasi Amas, Ajala Chosen, Andy Egbele, Valentine Oiwoh and my brother Appealcourt Bright for their encouragement, support and guidance throughout my stay in the University of Benin.

Finally, a general thanks to all who may not be mentioned here, but has sincerely contributed to the success of this work.

Table of Contents

CERTIFICATION	i
APPROVAL	ii
DEDICATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT.....	ix
CHAPTER ONE	1
INTRODUCTION	1
1.0 Background of the Study	1
1.1 Statement of the Problem.....	1
1.2 Aim and Objectives of the Study	2
1.3 Significance of the Study	2
1.4 Scope of the Study	3
1.5 Limitations of the Study.....	3
1.6 Report Layout	3
CHAPTER TWO	4
LITERATURE REVIEW	4
2.0 Introduction.....	4
2.1 SIWES Objectives	5
2.2 Challenges of SIWES	5
2.2.1 Improper and Inefficient Supervision	6
2.2.2 Reluctance of Organizations to Accept Trainees and Limited Space.....	6
2.2.3 Pleasure	6
2.2.4 Ethnocentrism, nepotism and cronyism.....	7
2.3 Review of Related Works	7
Summary of Related Works.....	9
CHAPTER THREE	11
SYSTEM ANALYSIS AND DESIGN.....	11
3.0 Introduction.....	11

3.1 Analysis of the Proposed System.....	11
3.2 Requirements of the System	11
3.3 System Architecture.....	12
3.4 Use Case Diagrams	13
3.5 Sequence Diagram	16
3.6 Class Diagrams	19
3.7 Entity-Relationship Diagram	22
3.8 Database Design.....	23
3.9 System Flow Chart.....	25
CHAPTER FOUR.....	28
SYSTEM IMPLEMENTATION	28
4.0 Introduction.....	28
4.1 Requirements to Run the System.....	28
4.1.1 Server Hardware Specification	28
4.1.2 Server Software Specification.....	28
4.1.3 Client Hardware Specification.....	29
4.1.4 Client Software Specification	29
4.2 System Sample Output.....	29
4.2.1 Home Page.....	29
4.2.2 Student Registration Page	30
4.2.3 Sign-in Page.....	31
4.2.4. Student Dashboard	31
4.2.5 View Profile Page (Student)	32
4.2.6 View Vacancies (Student).....	33
4.2.7 View Jobs Applied.....	33
4.2.8 Employer Registration Page	34
4.2.9 Unverified Employer Page.....	35
4.2.10 View All Companies Page.....	35
4.2.11 Create Vacancy Page	37
4.2.12 View Applications Page.....	38
4.2.13 View All Students Page	39

4.3 HOW TO INSTALL THE PROGRAM	41
4.4 HOW TO RUN THE PROGRAM.....	41
4.5 TYPE OF PROGRAMMING LANGUAGE USED	41
4.6 REASONS FOR CHOOSING JSP.....	41
CHAPTER FIVE	43
SUMMARY RECOMMENDATION AND CONCLUSION	43
5.0 SUMMARY	43
5.1 RECOMMENDATION	43
5.2 FUTURE WORK.....	44
5.3 CONCLUSION.....	44
REFERENCE	45
APPENDIX A.....	47

LIST OF FIGURES

Figure 1 System Architecture	12
Figure 2 Company Use Case.....	13
Figure 3 Student Use Case	14
Figure 4 Admin Use Case	15
Figure 5 Supervisor Use Case.....	16
Figure 6 Administrator Sequence Diagram	17
Figure 7 Company Sequence Diagram	18
Figure 8 Student Sequence Diagram.....	19
Figure 9 Class Diagram	20
Figure 10 Database Class Diagram.....	21
Figure 11 E-R Diagram.....	22
Figure 12 Administrator Flow chart Diagram	25
Figure 13 Employer Flow Chart Diagram	26
Figure 14 Student Flow Chart Diagram.....	27
Figure 15 Home Page.....	30
Figure 16 Student Registration Page.....	30
Figure 17 Sign in Page	31
Figure 18 Student Dashboard.....	32
Figure 19 View Profile Page (Student).....	32
Figure 20 View Vacancies (Student)	33
Figure 21 View Jobs Applied	34
Figure 22 Employer Registration Page	34
Figure 23 Unverified Employer Page	35
Figure 24 View All Companies Page.....	36
Figure 25 Nigeria Corporate Affairs Commission Company Search Page	36
Figure 26 Change Verification Status Page	37
Figure 27 Create Vacancy Page	38
Figure 28 View Applications Page	38
Figure 29 View Profile Page (Student).....	39
Figure 30 Jamb Students Verification Page.....	40
Figure 31 Edit Students Verification Page.....	40

ABSTRACT

The Students Industrial Work Experience Scheme (SIWES) serves as a vital bridge between academic learning and practical industry exposure for undergraduate students. Despite its importance, students of the University of Benin often encounter significant challenges in securing suitable Industrial Training (IT) placements, largely due to limited access to reliable information on available opportunities, organizational requirements, and appropriate application procedures. This project presents the design of a web-based Industrial Training Placement System aimed at enhancing and streamlining the placement process for UNIBEN students.

The proposed system provides a centralized platform where students can access verified IT-compliant organizations, review key information such as required skills, available roles, stipends, and organizational expectations, and submit placement applications conveniently. The Software Development Life Cycle (SDLC) methodology was adopted, guiding the structured development process through requirements analysis, system modeling, interface design, and database structuring.

The system is expected to significantly reduce the stress associated with manual placement searches, improve access to authentic placement information, and enhance the alignment between students' areas of study and available industry opportunities. Future enhancements may include integration with institutional systems and expansion into a mobile application for broader accessibility.

CHAPTER ONE

INTRODUCTION

1.0 Background of the Study

Industrial work experience is a vital component of all practice-oriented academic programs. According to Vivian and Chinedu (2019), it harmonizes the theoretical knowledge gained in classrooms with the practical skills needed in real-world applications, which is essential for building a strong technological foundation. Ukozor (2022) further emphasized that nations attain greatness through robust science and technology development, which relies heavily on practical skill acquisition.

The importance of industrial training in Nigeria led to the establishment of the Industrial Training Fund (ITF) through Decree No. 47 of 1971. According to the Industrial Training Fund the ITF was the first federal government agency mandated to develop skilled manpower for national growth. During its operations, the ITF identified a significant gap between the theoretical knowledge of locally trained engineers and their practical skills. To address this, the Students Industrial Work Experience Scheme (SIWES) was introduced in 1973 to bridge the gap between theory and practice. The scheme aims to complement higher institutions' efforts in producing graduates who are not only theoretically knowledgeable but also technologically competent and practically skilled.

1.1 Statement of the Problem

Participation in the SIWES program is a mandatory requirement for most undergraduate students before earning a bachelor's degree. However, obtaining an industrial placement is often challenging as students must secure temporary employment relevant to their field of study.

Given the prevailing high unemployment rate in Nigeria, many students encounter significant difficulties in securing suitable placement opportunities and consequently undertake their industrial training in organizations that have little or no relevance to their field of study.

In addition, many organizations that require IT students face difficulty attracting them due to limited resources or lack of visibility, thereby missing out on potential interns. This creates a communication gap between students seeking placements and organizations offering opportunities.

1.2 Aim and Objectives of the Study

The aim of this project is to design and develop an industrial training placement system for students using the University of Benin (UNIBEN), as a case study. The specific objectives of this study are to:

- a) Investigate the current industrial training placement procedures for students.
- b) Analyze data collected from the existing placement process.
- c) Design a comprehensive industrial training placement system.
- d) Evaluate the effectiveness and functionality of the proposed system.

1.3 Significance of the Study

This project is significant because it addresses the real and persistent challenge UNIBEN students face in securing industrial placements. The developed software will serve as a bridge between students and organizations, reducing stress and improving communication between both parties.

1.4 Scope of the Study

The research focuses primarily on students of the University of Benin. While industrial training is a common requirement across global tertiary institutions, the system developed in this project is tailored to help UNIBEN students quickly locate and apply to organizations within or outside their locality.

1.5 Limitations of the Study

The research was limited by time and financial resources, which restricted extensive data collection from departments and ITF. In addition, access to official placement data from the University of Benin SIWES office was limited, making it difficult to validate all organizational entries. The system was developed and tested using a restricted sample of student and company data hence, its performance across all departments could not be fully evaluated.

Despite these challenges, significant progress was achieved in meeting the study's objectives.

1.6 Report Layout

Chapter One: Introduces the background, problem statement, objectives, significance, scope, and limitations of the study.

Chapter Two: Reviews literature on SIWES, the Industrial Training Fund (ITF), and related systems.

Chapter Three: Discusses system investigation and analysis, highlighting the existing system and the proposed improvements.

Chapter Four: Covers the system design, development, and implementation stages.

Chapter Five: Presents the summary, conclusion, and recommendations of the research.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

The Students Industrial Work Experience Scheme (SIWES) is a skills training program established in 1973 to prepare students from universities, polytechnics, colleges of technology, agriculture, and education for real-world industrial environments after graduation. It was designed to bridge the gap between classroom theory and practical application, providing students with on-the-job experience, exposure to work methods, and hands-on use of equipment and machinery often unavailable in their institutions.

At its inception in 1974, SIWES began with 784 students from 11 institutions and 104 eligible courses. By 2008, participation had grown to 210,390 students from 219 institutions covering over 112 eligible courses. However, this rapid growth has been hindered by recurring economic challenges affecting the smooth administration of the scheme

A study conducted by Oyeniyi et al. (2021) in collaboration with the Industrial Training Fund (ITF) highlighted that student placement into relevant industries remains a major challenge, largely due to the increasing number of higher institutions and students participating in the SIWES program, which has outpaced the available industrial opportunities. Many students report unpleasant experiences securing placements, often struggling to find firms that can accommodate and financially support them (Oluwaseun, 2022), despite ITF appeals for organizations to accept interns. Similar placement challenges are faced by medical students due to factors such as physician migration, health worker strikes, and limited expansion of healthcare facilities (Oghenekaro, 2017).

While there are still organizations willing to accept interns, an information gap often prevents students from connecting with them. This research proposes the development of an Industrial Training Placement System aimed at bridging the gap between students and organizations. The system will streamline the process of matching students with relevant companies, enhance communication, and provide a centralized platform for managing placement records. By improving accessibility and efficiency, the proposed system will benefit both students seeking industry exposure and organizations looking for skilled trainees.

2.1 SIWES Objectives

Specifically, the objectives of the Students Industrial Work Experience Scheme (SIWES) are to:

- Provide an avenue for students in institutions of higher learning to acquire industrial skills and experience during their course of study.
- Prepare students for the industrial work situation they are likely to meet after graduation.
- Expose students to work methods and techniques in handling equipment and machinery that may not be available in their institutions.
- Make the transition from school to the world of work easier, and enhance students' contacts for later job placements.
- Provide students with an opportunity to apply their theoretical knowledge in real work situations.
- Provide an avenue for students to gain practical knowledge in their field of study.

2.2 Challenges of SIWES

The Students Industrial Work Experience Scheme (SIWES) faces several persistent challenges that hinder its effectiveness; some are discussed hereunder.

2.2.1 Improper and Inefficient Supervision

Effective supervision is critical to the success of SIWES, yet it remains a weak point in its administration. The operational guidelines mandate oversight from three officials: an institution-based supervisor, an industry-based supervisor, and an ITF representative. Industry-based supervisors are expected to evaluate student progress weekly, while institution-based supervisors should conduct monthly visits to ensure the training aligns with the student's field of study. ITF officials are to visit at least once during the attachment to confirm participation and relevance. However, some schools and ITF find it very difficult to go or send representatives to the various workplaces of their internship students – to monitor, supervise, grade and advise them on the right things to do. Because of this, some students don't even go for training at all. And those that go are not properly trained because of the lack of supervision (Saint, 2023).

2.2.2 Reluctance of Organizations to Accept Trainees and Limited Space

Despite ITF regulations mandating companies to accept trainees, economic downturns, downsizing, and closures have led many organizations to refuse placements. This shortage of opportunities forces some students to migrate to only industrial states such as Lagos, Port-Harcourt, Abuja for their training, causing overpopulation and competition (Saint, 2023).

2.2.3 Pleasure

Most Nigerian internship students see the training period as a time to carry out unnecessary activities, such as travelling, partying due to lack of interest and motivation. They eventually miss out on essential knowledge and information which will guide them in their various future occupations, goals and aspirations (Saint, 2023).

2.2.4 Ethnocentrism, nepotism and cronyism

In the selection of interns, merits and conformity to the Federal Character Commission Act are the least considerations. The process is heavily influenced by politicians and their ilk. It's a matter of whose child, relative or friend you're, where you're from, the language you speak and who is speaking for you. Of course, those without any connection to the powers that be in the society are constantly being left out while the 'sons of Abraham' have several offers to choose from, discarding the remaining jobs unofficially (Oghenekaro, 2017).

2.3 Review of Related Works

Hasti et al. (2019), in “Web-Based Internship Information System”, addressed challenges in all phases of the internship process — from registration and placement selection to supervision, report submission, and evaluation. Using a descriptive research method, object-oriented system design, and the Prototype development approach, the study gathered data through interviews, observation, and literature review. The resulting web-based system reduced management errors and provided students with faster, more reliable internship information.

Ferdian et al. (2021), in “A Framework of University Internship Information System with Web-Based Design Analysis”, designed a specialized web platform to streamline the internship process for the Industrial Engineering Study Program, Atma Jaya Catholic University. Developed using the System Development Life Cycle (SDLC) framework and prototyping, the system employed HTML, CSS, PHP, and JavaScript. Development stages included planning, requirement analysis, database design, prototyping, and implementation, followed by testing to ensure functionality and usability.

Olasehinde et al. (2022), in “Design and Implementation of a Web-Based Internship Placement Recommendation System: A Case Study of Federal Polytechnic Ile-Oluji, Nigeria”, created a recommendation-based platform using a hybrid of Content-Based, Collaborative, and Knowledge-Based Filtering. This approach matches students to suitable organizations based on academic background and interests. Both students and companies can register, submit applications, and manage placements online. Built with PHP, MySQL, HTML, CSS, and JavaScript, the system followed an SDLC approach. Testing showed improved placement efficiency and accuracy compared to manual processes.

Dangana et al. (2023), in “Development of SIWES Management System for Nasarawa State University, Keffi: A Web-Based Approach”, developed a fully integrated web application to manage the Student Industrial Work Experience Scheme (SIWES). The system centralizes student registration, supervisor assignment, document uploads, and evaluation tracking, with features such as bulk student imports, authentication, and automated letter downloads. Following the Object-Oriented Analysis and Design Methodology (OOADM) with UML modeling, the system was built using PHP, HTML/CSS, JavaScript, and MySQL. Adopting the SDLC model, the project delivered a structured, efficient, and scalable solution to replace the manual system.

Hamam et al. (2025), in “Web-Based Internship Recruitment System with Integration of Instant Notification Service for University Students”, developed an online recruitment platform designed to connect students with internship opportunities while providing real-time updates through an integrated instant notification service. The system enables students to search and apply for internships, track application progress, and receive status alerts, while employers can post opportunities and review applications through their dashboards. Built using a combination of web technologies for interactivity and responsiveness, the development followed a structured

methodology to ensure a functional and user-friendly interface. Testing confirmed that the inclusion of the notification service improved communication efficiency and reduced delays in the recruitment process.

Summary of Related Works

Author (Year)	Aim	Methodology	Results / Findings	Limitations
Hasti et al. (2019)	To develop a web-based internship information system to manage registration, placement selection, report submission and appraisal.	Descriptive research, Object-Oriented, Prototype development approach.	Provided an integrated web system for registration, supervisor selection, report submission and evaluation, reducing manual errors.	Limited to a particular institution, prototype scope limited, no evaluation of scalability, security, or mobile access.
Ferdian et al. (2021)	To design a university internship information system to facilitate internship workflow for students, lecturers and coordinators.	Software Development Life Cycle with prototyping; front-end with HTML/CSS, back-end with PHP and JavaScript, database-driven design.	System mapped internship process flow into a web application, improving coordination and document handling among stakeholders.	Prototype testing, empirical user study limited, performance and cross-department scalability not fully evaluated.
Olasehinde et al. (2022)	To design a hybrid recommendation-based internship placement system to better match students with organizations.	Hybrid recommender (Content-Based + Collaborative + Knowledge-Based); implementation with PHP, MySQL, HTML/CSS, JavaScript; SDLC methodology.	Improved placement matching accuracy and reduced administrative burden; system suggested relevant organizations based on profiles.	Relied on limited dataset from one institution, no real-time data integration and limited evaluation metrics.
Dangana et al. (2023)	To develop a SIWES management portal for Nasarawa State	Object-Oriented Analysis and Design with UML modeling, implemented using PHP, JavaScript,	Delivered admin, coordinator and student dashboards, support for bulk	No mobile app or notification features, limited external integration (e.g.,

	University to automate student registration, supervisor assignment and document management.	HTML/CSS and MySQL.	uploads, document upload/download and evaluation tracking.	employer APIs).
Hamam et al. (2025)	To build an AI-driven industrial training system integrating cloud services and IoT for real-time oversight.	Agile methodology; technologies include Python (backend), Firebase (cloud DB), IoT sensors for monitoring, and cloud APIs for integration.	Enabled intelligent matching, live monitoring and automatic reporting to supervisors.	High implementation cost, dependency on stable network and IoT hardware.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.0 Introduction

This chapter shows all the modules and components used to design the system, and how they work together. It also shows us how the users of the system interact with the system.

3.1 Analysis of the Proposed System

System analysis and design deal with planning the development of information systems through understanding and specifying in detail what a system should do and how the components of the system should be implemented and work together. System analysts solve business problems through analyzing the requirements of information systems and designing such systems by applying analysis and design techniques.

3.2 Requirements of the System

To successfully develop the SIWES System, it would have to meet the following requirements:

1. Companies should be able to register on the system
2. Students should be able to register on the system
3. Companies should be able to manage their profile on the system
4. Companies should be able to post Industrial Training Vacancy on the system
5. Companies should be able to view applications from prospective students on the system.
6. Companies should be able to send invitation to prospective students on the system
7. Students should be able to manage their profile on the system
8. Students should be able to search for Industrial Training vacancy.
9. Students should be able to apply for Industrial Training in companies.

10. The administrator should be able to verify companies
11. The administrator should be able to verify students
12. The system should have an attendance module that records the attendance of the student in his/her respective industrial attachment office.
13. The student should be able to generate a report of the activities done per day.
14. The system should be able to suggest to students the most suitable organization that will give the best experience based on the student's course of study.
15. The system should be able to assign students to supervisor.

3.3 System Architecture

A system architecture diagram is used to show the relationship between different components. Usually they are created for systems which include hardware and software and these are represented in the diagram to show the interaction between them. However, it can also be created for web applications. For a web application the system architecture design would include components such as, database, application server, web server, internet, browser etc. Not all of these have to be included in the diagram and there are other components that can be included. Below is the system architectural diagram of our application.

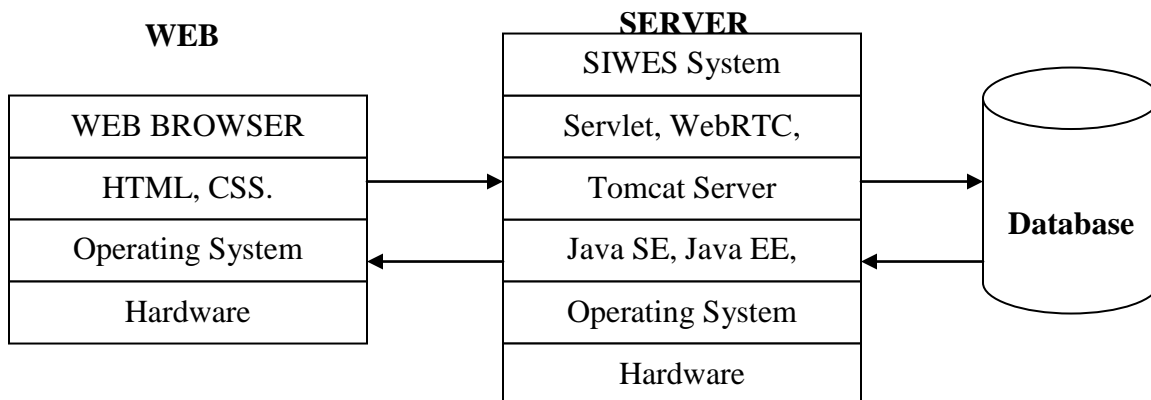


Figure 1 System Architecture

3.4 Use Case Diagrams

The use case diagram is used to show the interaction between the system use cases and its clients without much detail. A use case diagram displays an actor and its use cases, the actors are also the users of the system.

The users or actors of our system include

1. Administrator
2. Companies
3. Student
4. Supervisor

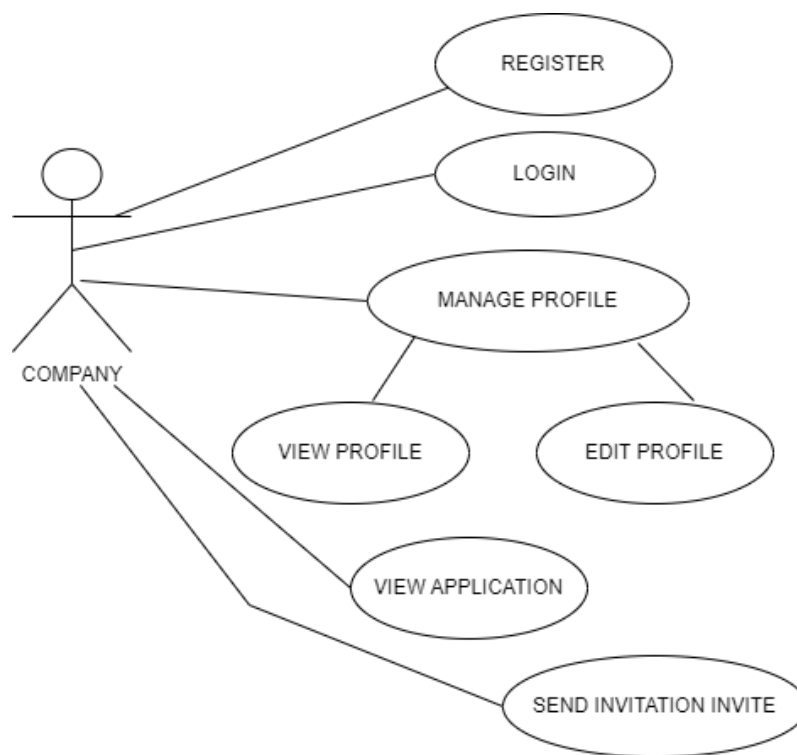


Figure 2 Company Use Case

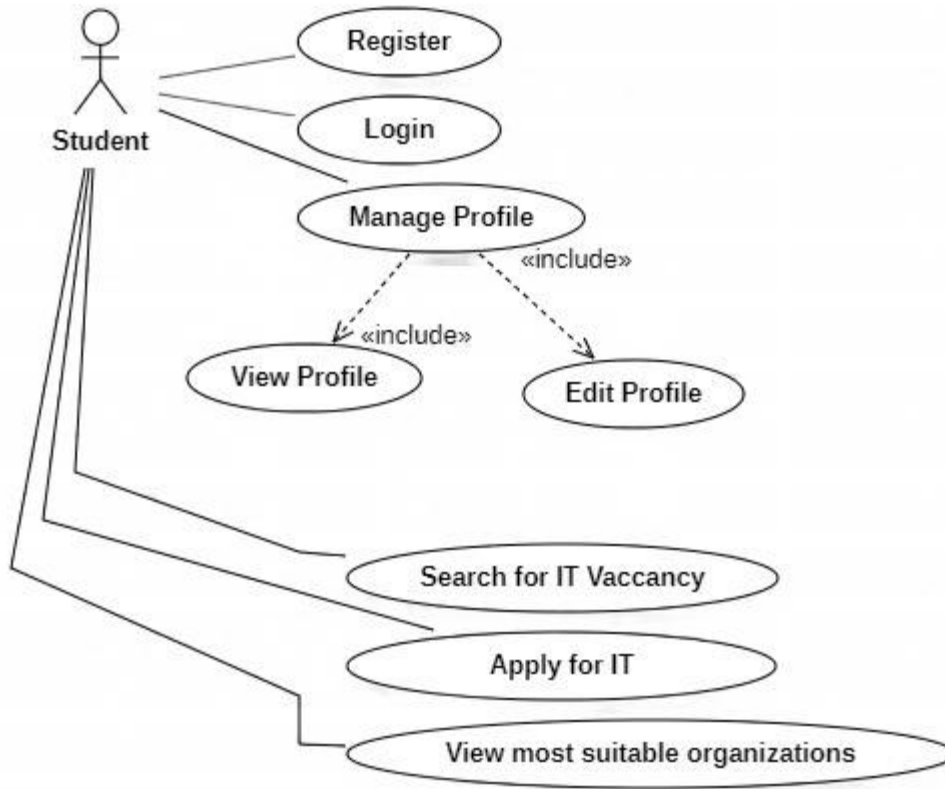


Figure 3 Student Use Case

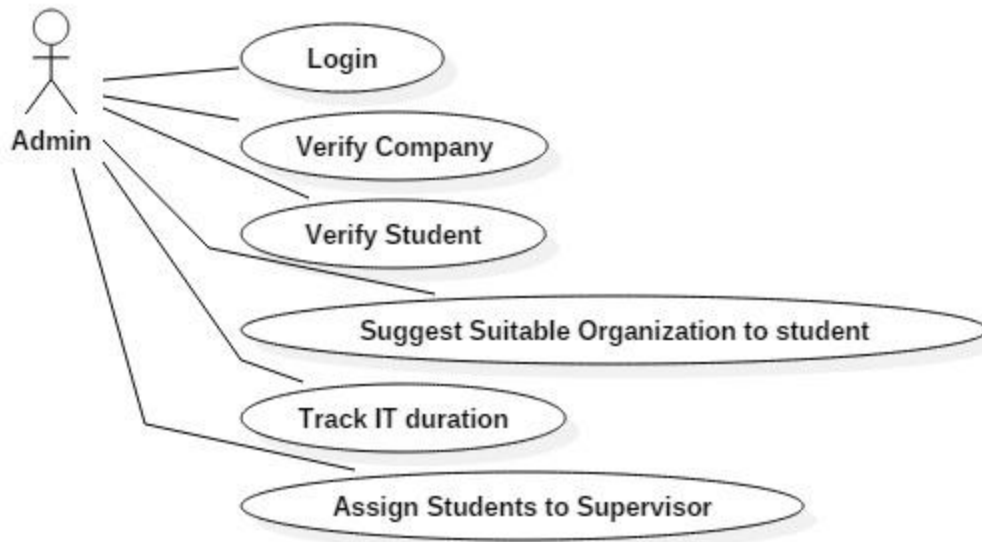


Figure 4 Admin Use Case

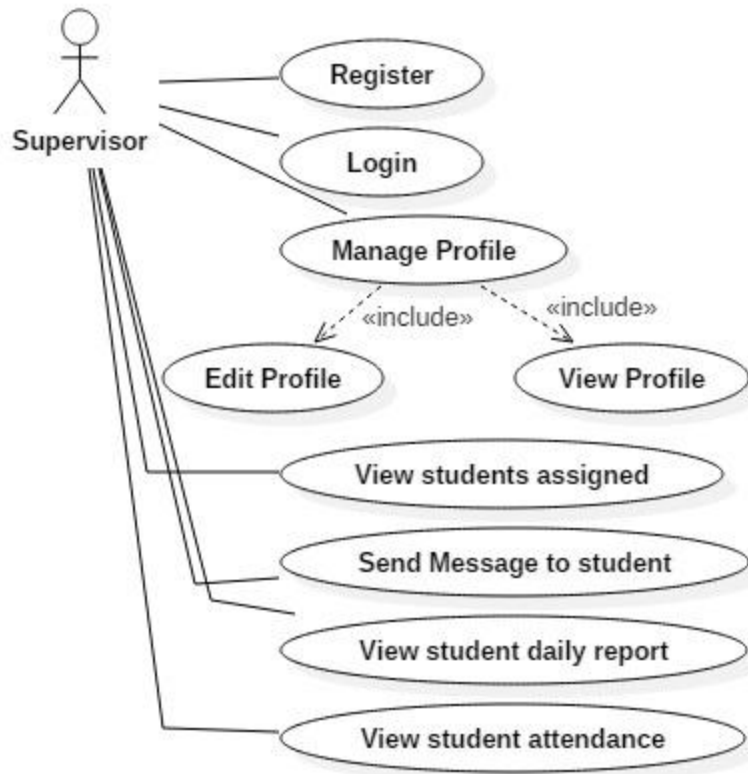


Figure 5 Supervisor Use Case

3.5 Sequence Diagram

Sequence diagrams are simple subsets of interaction diagrams. They map out sequential events in an engineering or business process in order to streamline activities. Sequence diagrams are used to show how objects interact in a given situation. An important characteristic of a sequence diagram is that time passes from top to bottom: the interaction starts near the top of the diagram and ends at the bottom (i.e. **L**ower equals **L**ater).

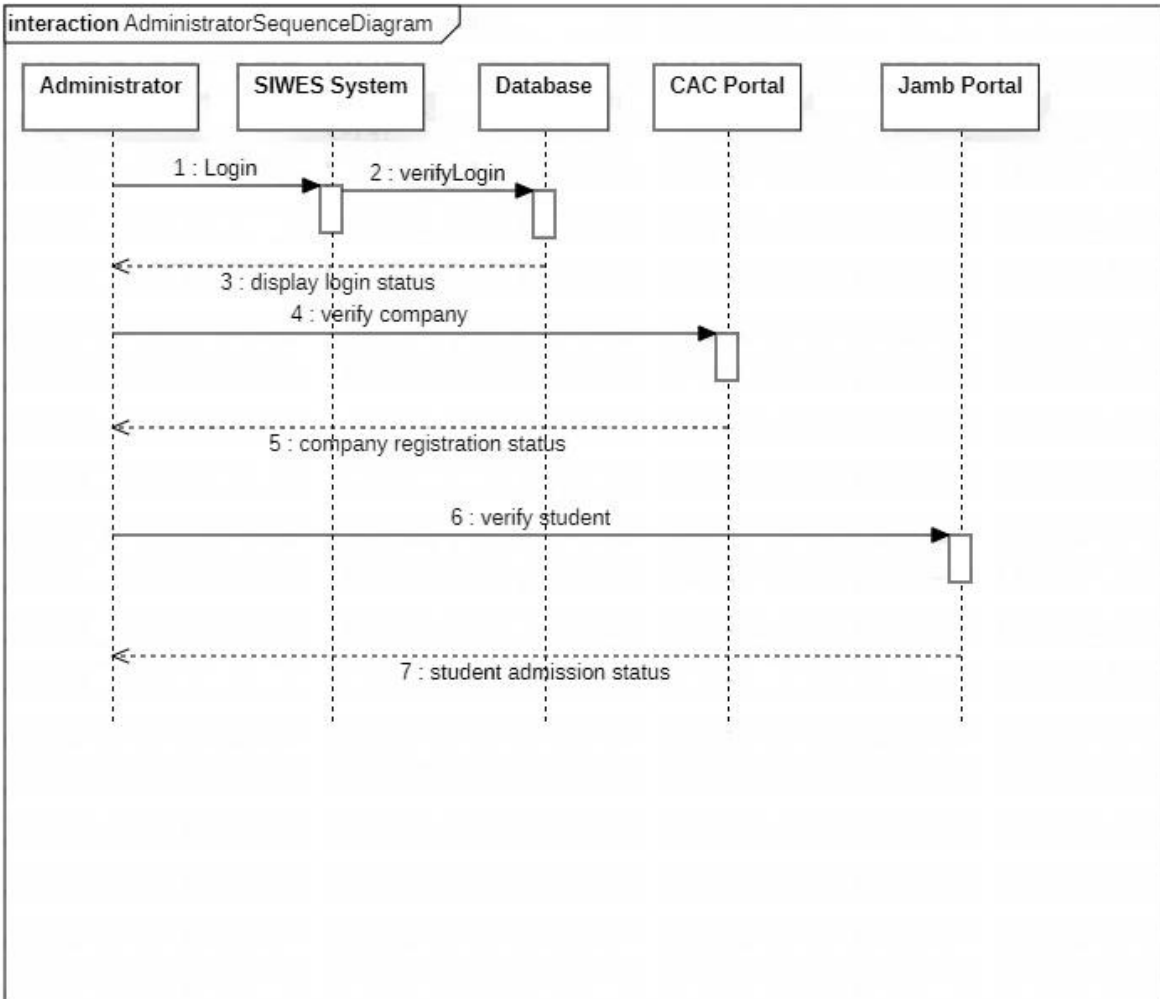


Figure 6 Administrator Sequence Diagram

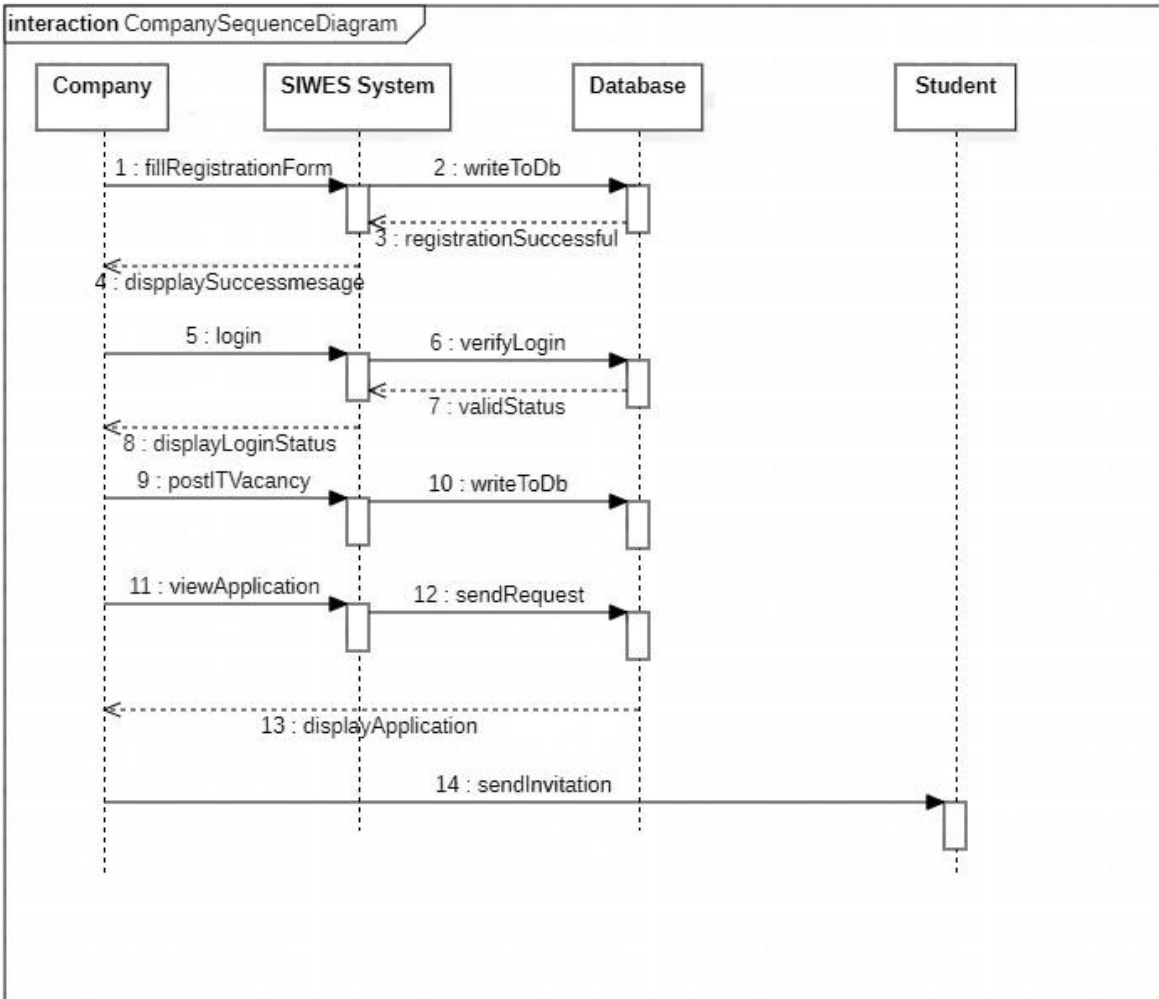


Figure 7 Company Sequence Diagram

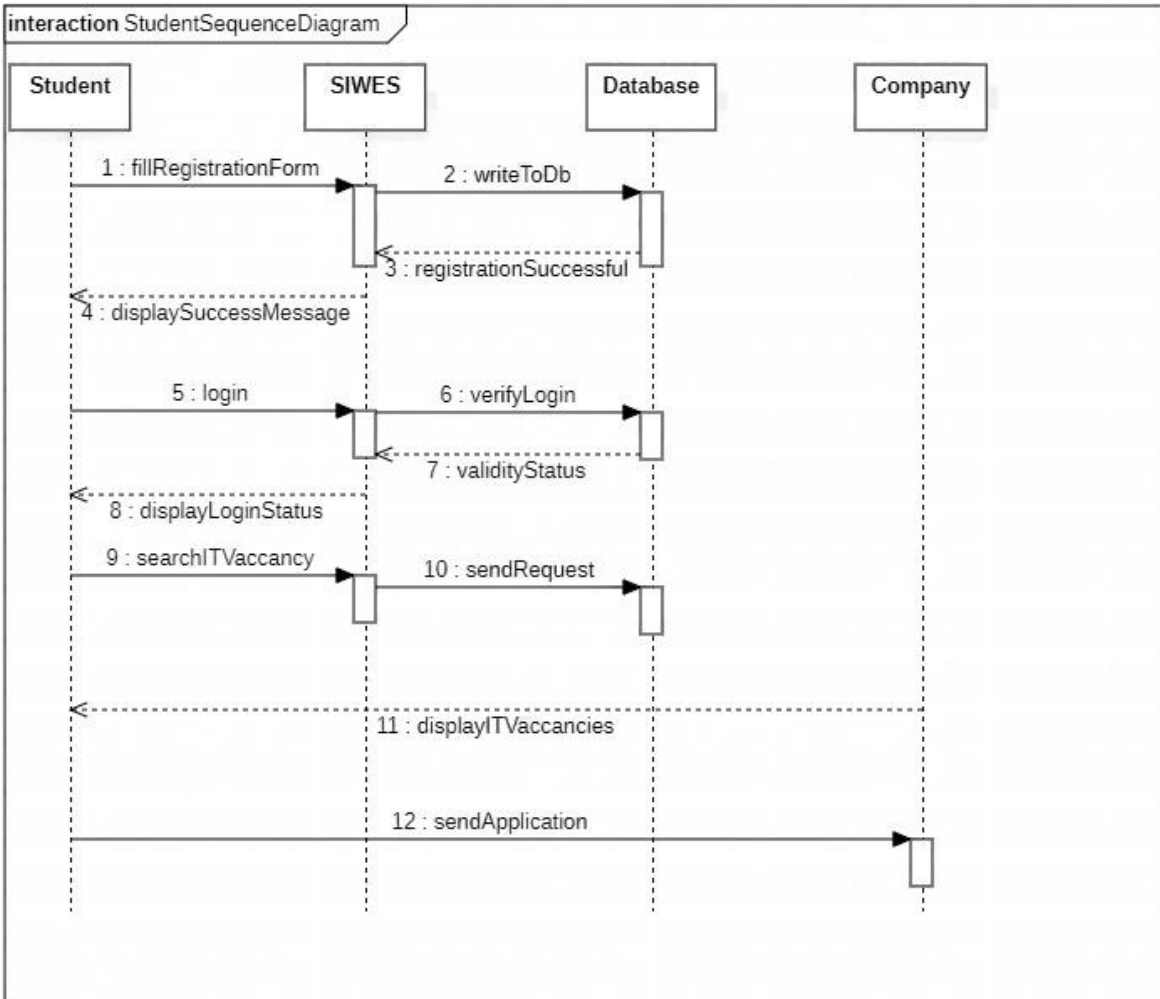


Figure 8 Student Sequence Diagram

3.6 Class Diagrams

We begin our OOD process by identifying the classes required to build the conference management system. We describe these classes using class diagrams and implement them in Java. The class diagram enable us to model via class diagrams, each class is modelled as a rectangle with three compartments. The top one contains the name of the class centered horizontally in bold face. The middle compartment contains the class attributes, while the bottom

compartment contains the class behaviour or operation. Below is the class diagram for the computer based testing system.

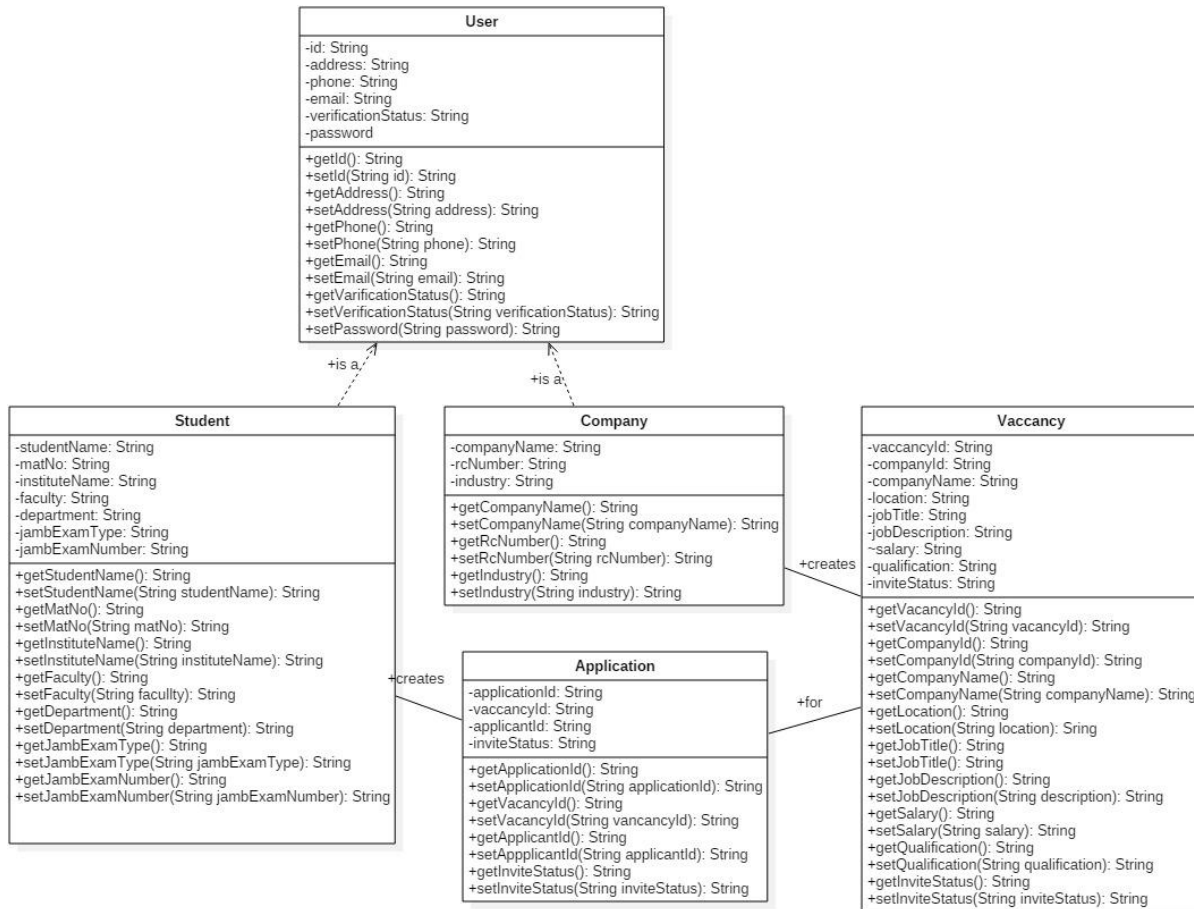


Figure 9 Class Diagram

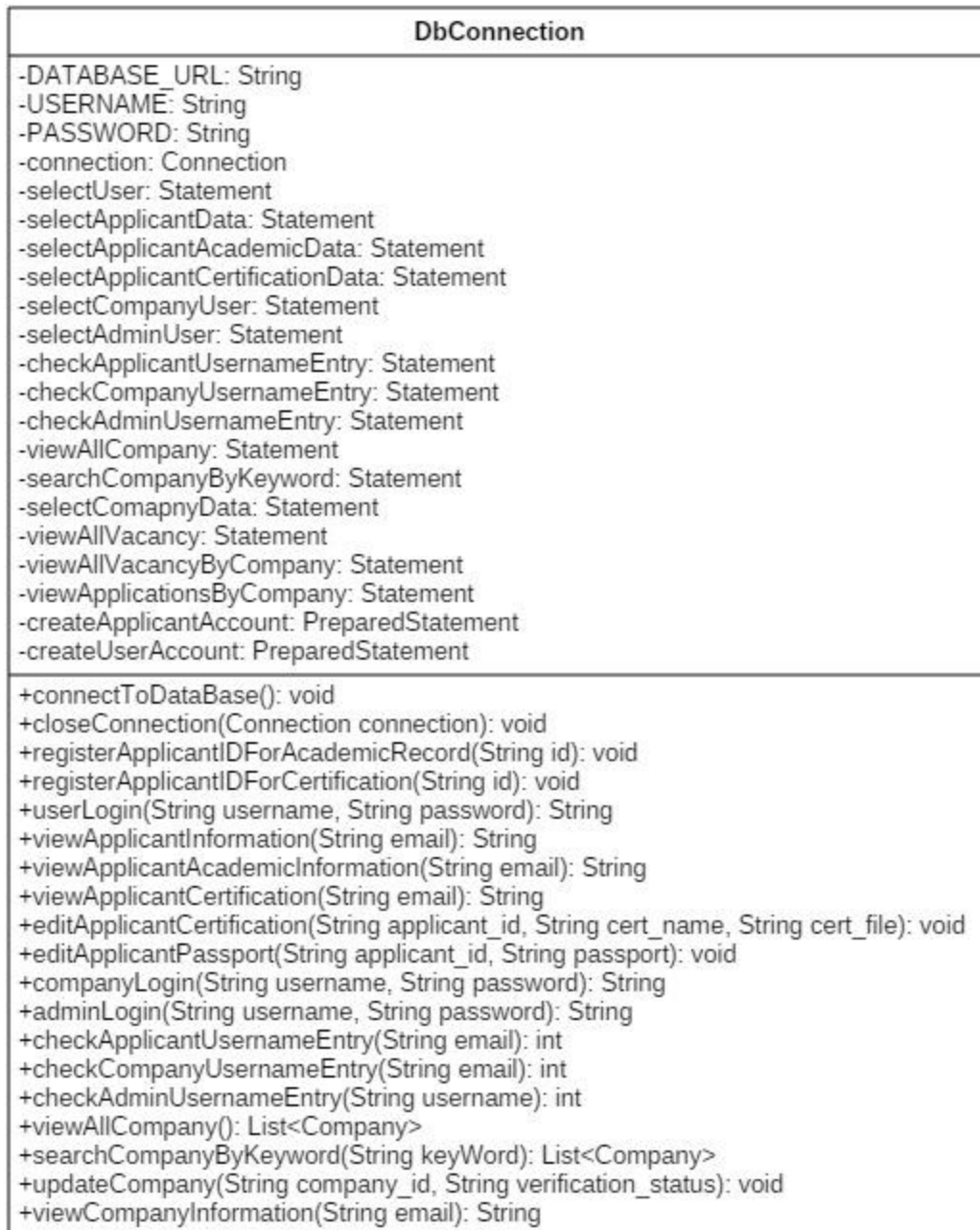


Figure 10 Database Class Diagram

3.7 Entity-Relationship Diagram

An Entity-relationship model (ERM) is an abstract and conceptual representation of data. E-R modelling is a database modelling method, used to produce a type of conceptual schema of a system. Diagrams created by this process are called ER diagrams. In this section we present a set of E-R diagrams the conceptual database schema of the system.

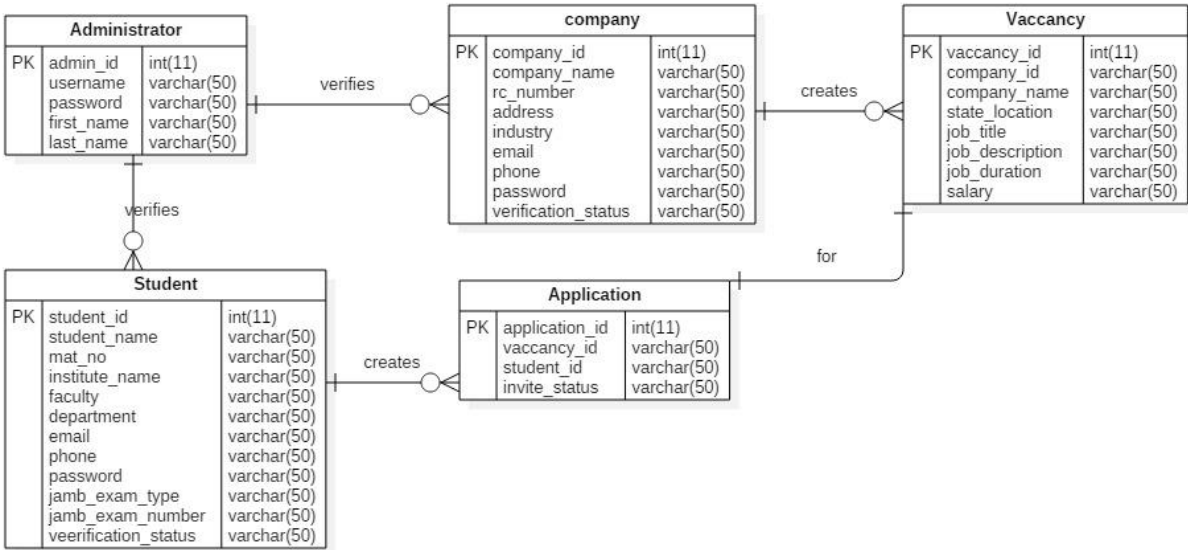


Figure 11 E-R Diagram

3.8 Database Design

In this section, the basic structure of the tables composing the database for the project are shown along with information about primary and foreign keys.

Table 1 Admin

Field	Type	Null	Key	Default	Extra
admin_id	Auto_increment	NO	PRI	NULL	
Username	Varchar(50)	YES		NULL	
Password	Varchar(50)	YES		NULL	
first_name	Varchar(50)	YES		NULL	
last_name	Varchar(50)	YES		NULL	

Table 2 Application

Field	Type	Null	Key	Default	Extra
application_id	Auto_increment	NO	PRI	NULL	
vaccancy_id	Varchar(50)	YES		NULL	
student_id	Varchar(50)	YES		NULL	
invite_status	Varchar(50)	YES		NOT_INVITED	

Table 3 Company

Field	Type	Null	Key	Default	Extra
company_id	Auto_increment	NO	PRI	NULL	
company_name	Varchar(50)	YES		NULL	
rc_number	Varchar(50)	YES		NULL	
Address	Varchar(50)	YES		NULL	
Industry	Varchar(50)	YES		NULL	
Email	Varchar(50)	YES		NULL	
Phone	Varchar(50)	YES		NULL	
Password	Varchar(50)	YES		NULL	
verification_status	Varchar(50)	YES		NOT_VERIFIED	

Table 4 Student

Field	Type	Null	Key	Default	Extra
student_id	Auto_increment	NO	PRI	NULL	
student_name	Varchar(50)	YES		NULL	
mat_no	Varchar(50)	YES		NULL	
institute_name	Varchar(50)	YES		NULL	
Faculty	Varchar(50)	YES		NULL	
Department	Varchar(50)	YES		NULL	
Email	Varchar(50)	YES		NULL	
Phone	Varchar(50)	YES		NULL	
Password	Varchar(50)	YES		NULL	
jamb_exam_type	Varchar(50)	YES		NULL	
jamb_exam_number	Varchar(50)	YES		NULL	
verification_status	Varchar(50)	YES		NOT_VERIFIED	

Table 5 Vacancy

Field	Type	Null	Key	Default	Extra
vacancy_id	Auto_increment	NO	PRI	NULL	
company_id	Varchar(50)	YES		NULL	
company_name	Varchar(500)	YES		NULL	
state_location	Varchar(500)	YES		NULL	
job_title	Varchar(500)	YES		NULL	
job_description	Varchar(500)	YES		NULL	
job_duration	Varchar(500)	YES		NULL	
Salary	Varchar(500)	YES		NULL	

3.9 System Flow Chart

This is a graphical representation of the sequence of operations in an information system or program. Information system flowcharts show how data flows from source documents through the computer to final distribution to users. The following figures are the system flow chart for our computer based testing system.

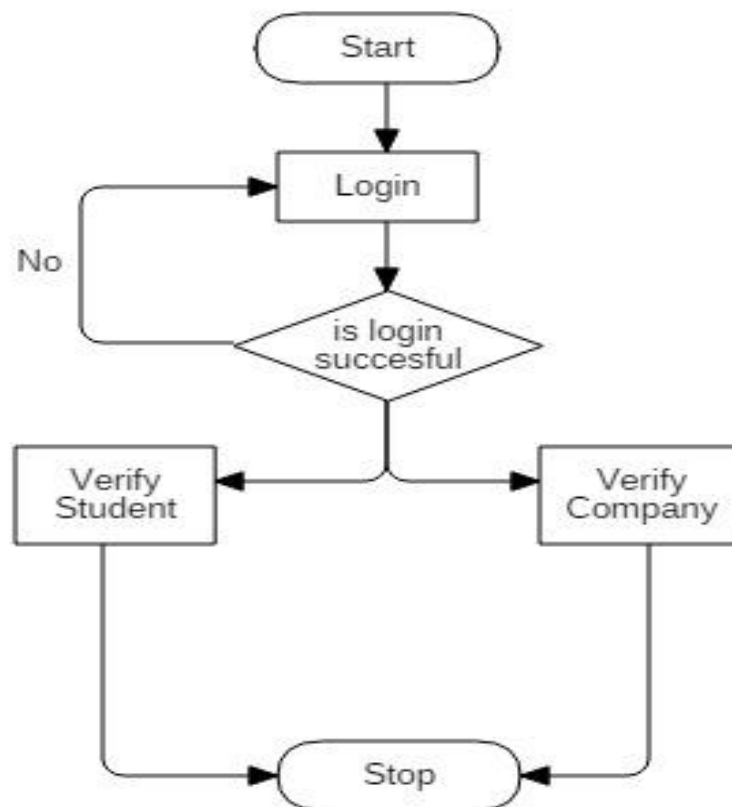


Figure 12 Administrator Flow chart Diagram

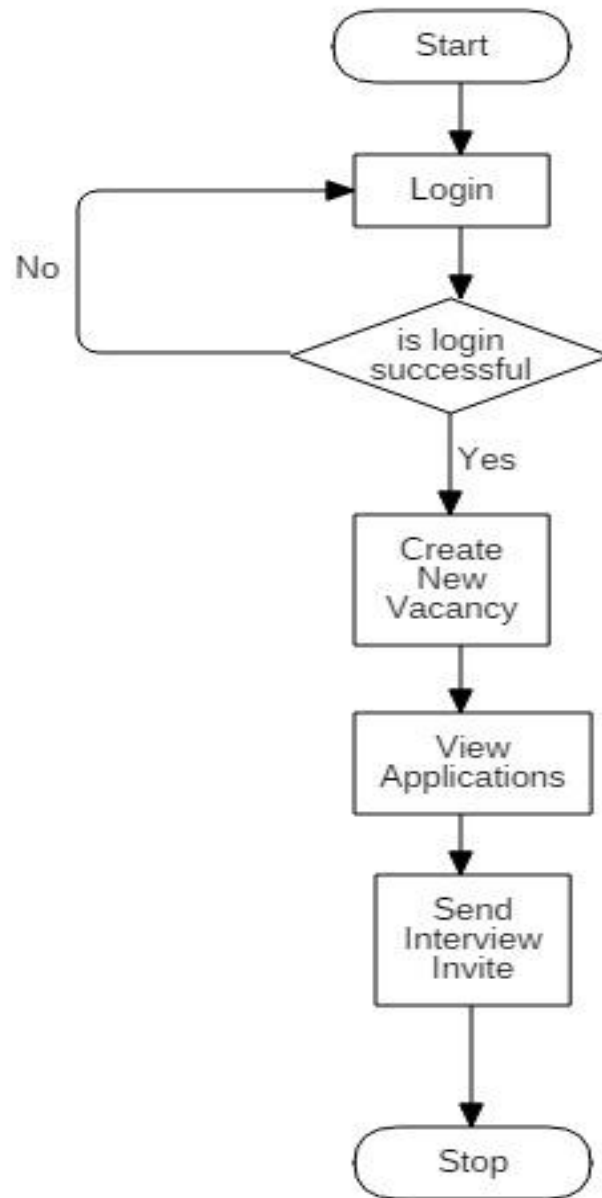


Figure 13 Employer Flow Chart Diagram

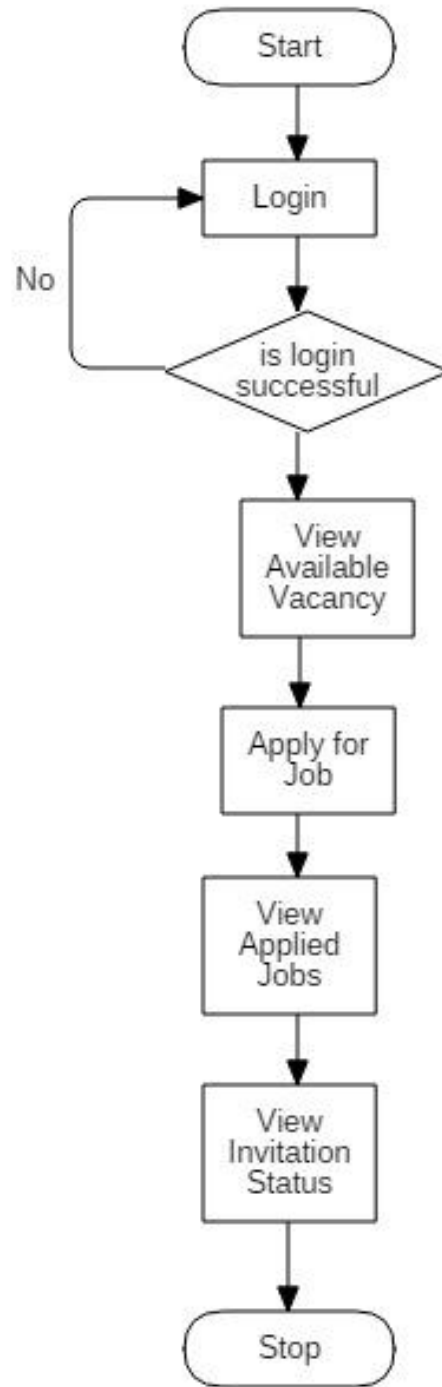


Figure 14 Student Flow Chart Diagram

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.0 Introduction

Systems implementation is the construction of the new system and the delivery of that system into production (that is, the day-to-day business or organization operation). It uses the structure created during architectural design and the results of system analysis to construct system elements that meet the stakeholder requirements and system requirements developed in the early life cycle phases.

4.1 Requirements to Run the System

To run the system, we will need certain hardware and software specification, for both the client and the server computer. This section will outline these specifications

4.1.1 Server Hardware Specification

1. 1GB of RAM
2. 320 GB Hard Disk
3. 2.0 GHz processor speed.
4. 15 inches Monitor Screen

4.1.2 Server Software Specification

1. 32-bit Operating System, Windows or Linux
2. Java Runtime Environment, version 7
3. Apache Tomcat Server version 7
4. MySQL version 5

4.1.3 Client Hardware Specification

1. 1GB of RAM
2. 320 GB Hard Disk
3. 2.0 GHz processor speed.
4. 15 inches Monitor Screen

4.1.4 Client Software Specification

1. 32-bit Operating System, Windows or Linux
2. Web browser

4.2 System Sample Output

This section displays the sample interface, and describes the functions of each web page in the system.

4.2.1 Home Page

This is the first page that is seen once the application is launched by visiting the URL. From the home page, the student, and the employers can create accounts using the buttons labeled “Student” and “Employer” respectively. Clicking on the site name “SIWES” will always take you back to the home page.

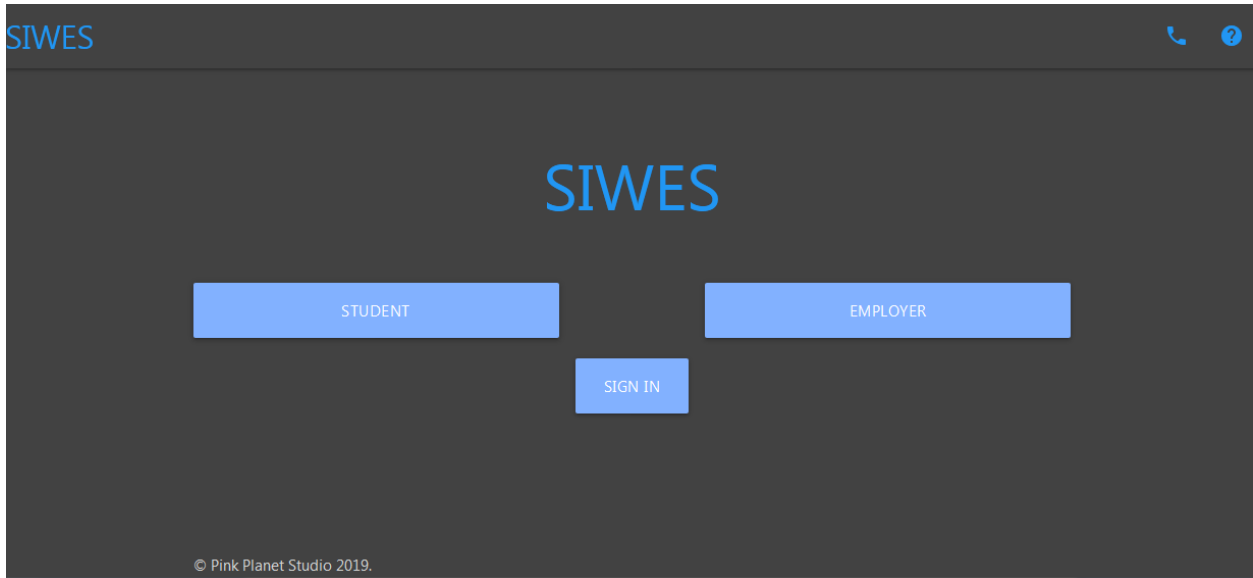


Figure 15 Home Page

4.2.2 Student Registration Page

Clicking the “Student” button will take you to the student registration page. This page contains a small form to capture the basic information of a student. To create an account, the student fills the form and clicks the “Create Account” Button

Figure 16 Student Registration Page

4.2.3 Sign-in Page

This page contains the sign in form for users to enter their username and password. The sign in page sends the username and password to the `LoginRouteController.java` servlet that authenticates it and determines if the user is a student an employer or an administrator.

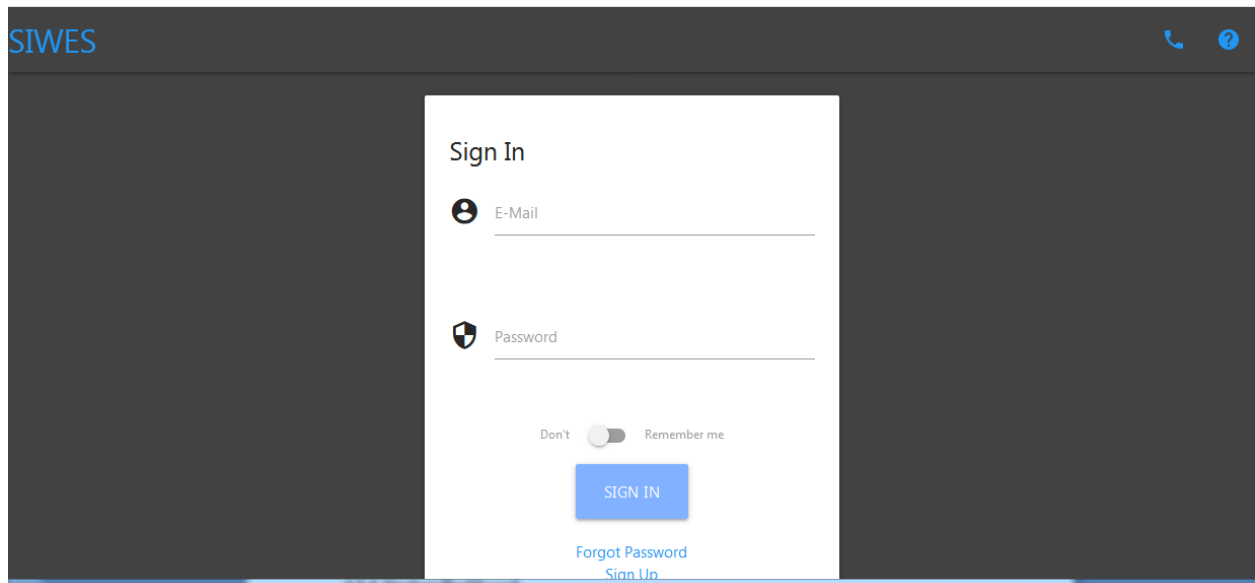


Figure 17 Sign in Page

4.2.4. Student Dashboard

Below is the student dashboard, after logging in, it contains a navigation bar to the left which contains links for the student to manage his/her profile and also to view vacancies and apply for jobs.



Figure 18 Student Dashboard

4.2.5 View Profile Page (Student)

This page is used by the students to view their profile.



Figure 19 View Profile Page (Student)

4.2.6 View Vacancies (Student)

This page is used by students to view vacancies posted by employers, the page displays a list of vacancies with an “Apply” button attached to each entry on the list, from here, the student can easily apply for jobs they are interested in by clicking the “Apply” button.

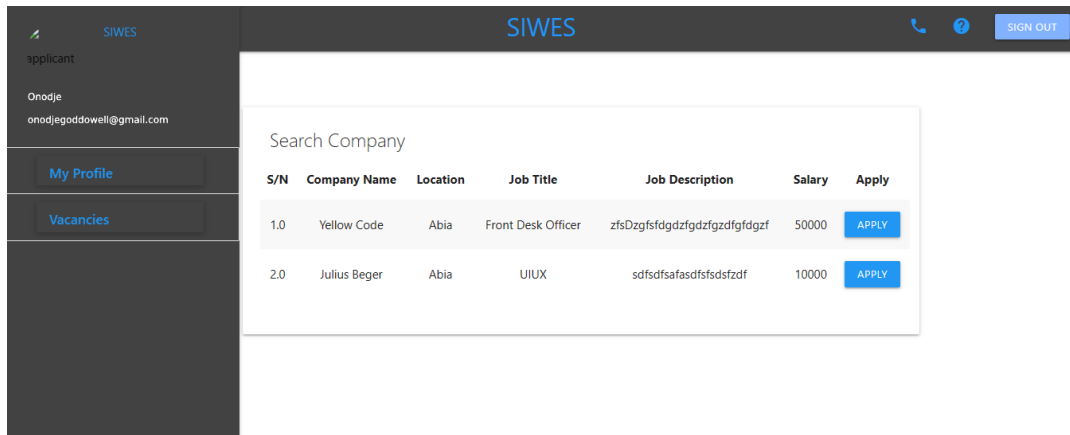


Figure 20 View Vacancies (Student)

4.2.7 View Jobs Applied

From this page, the student can view a list of all the jobs they have applied for, their invitation status (if they have been invited for an interview or not) is also displayed. If an student has not been invited, their invitation status displays with a red background and green otherwise.

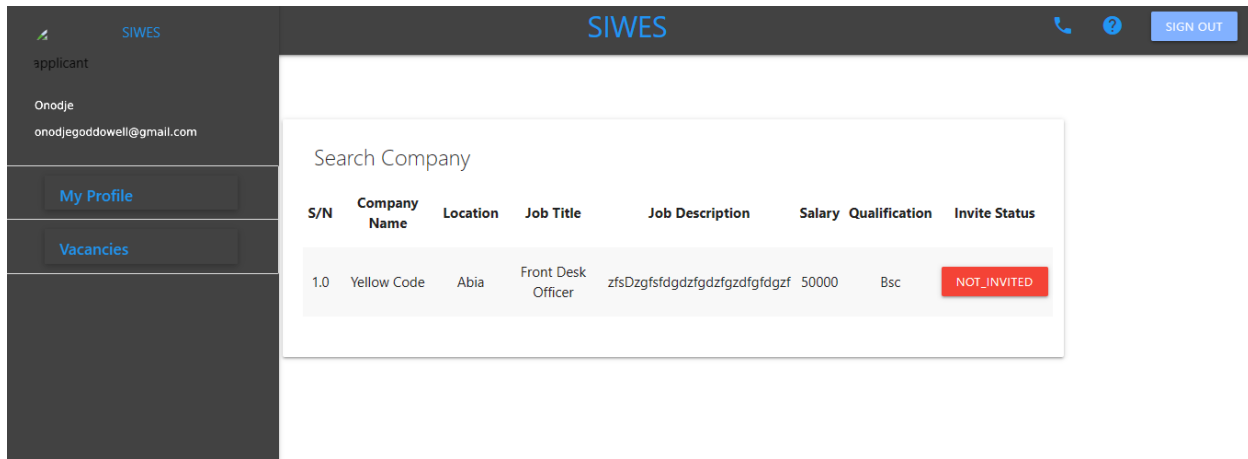


Figure 21 View Jobs Applied

4.2.8 Employer Registration Page

Clicking the “Employer” button will take you to the employer registration page. This page contains a form to capture the information of an employer. To create an account, the employer fills the form and clicks the “Create Account” Button

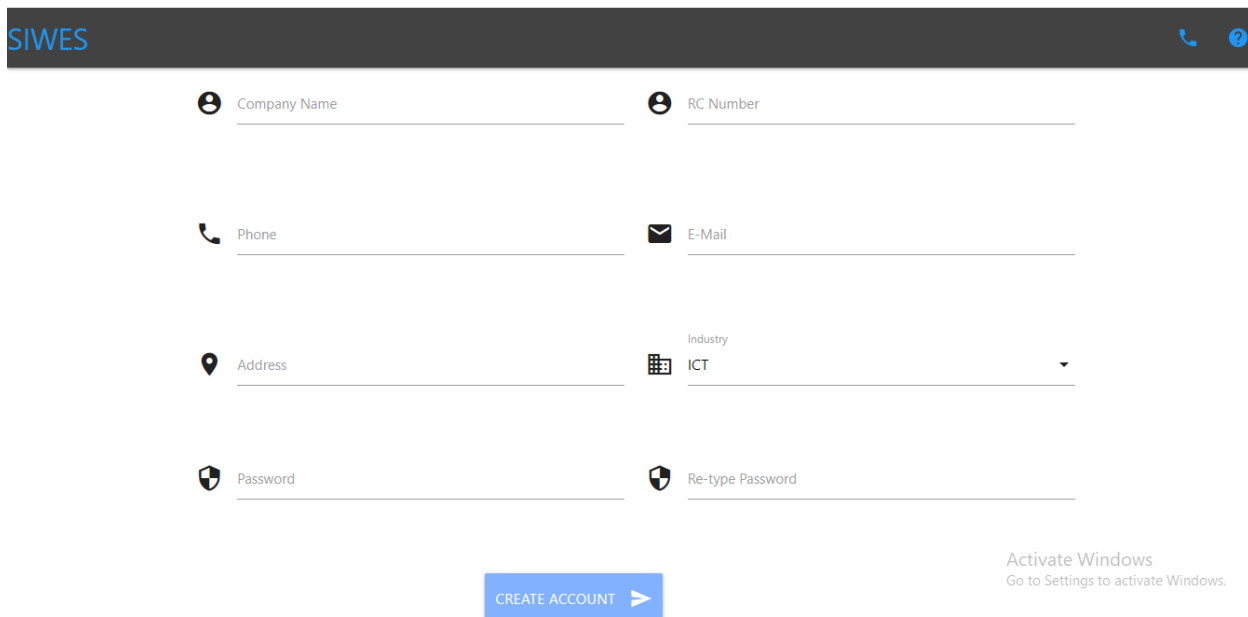


Figure 22 Employer Registration Page

4.2.9 Unverified Employer Page

This page displays to all newly registered employers, this page informs the newly registered employer that their newly created account has not yet been verified by the administrator. This is where the security, of the system is exhibited, this security feature ensures that rogue employers and employment scammers cannot gain access to the system until proper due diligence has been done on their registered profile.

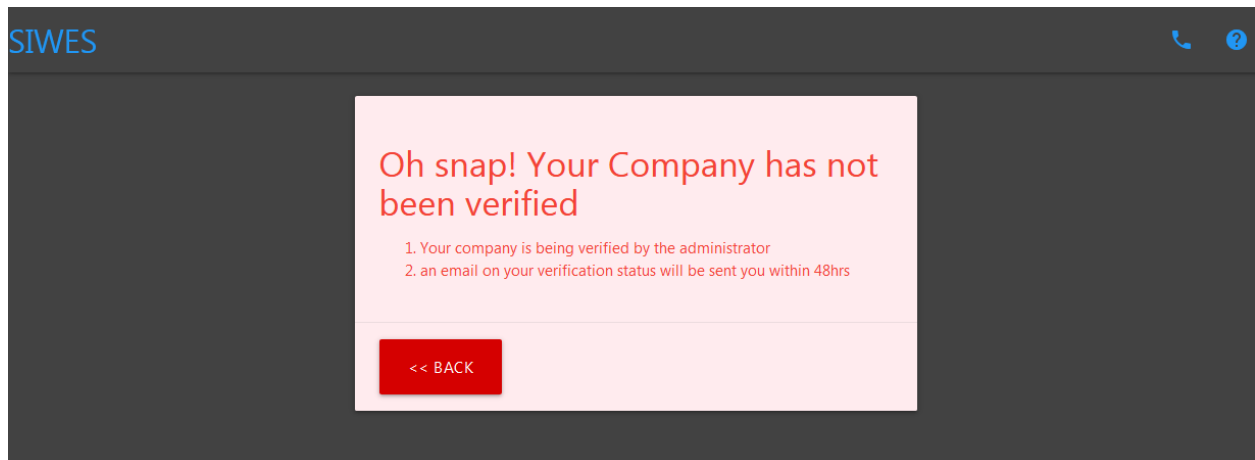


Figure 23 Unverified Employer Page

4.2.10 View All Companies Page

This page is used by the administrator to view all the companies/employers registered on the system along with their corresponding RC Number, also the verification status of the company is also displayed. Verified companies display with a green background and text “VERIFIED” while unverified companies display with a red background and text “NOT_VERIFIED”.

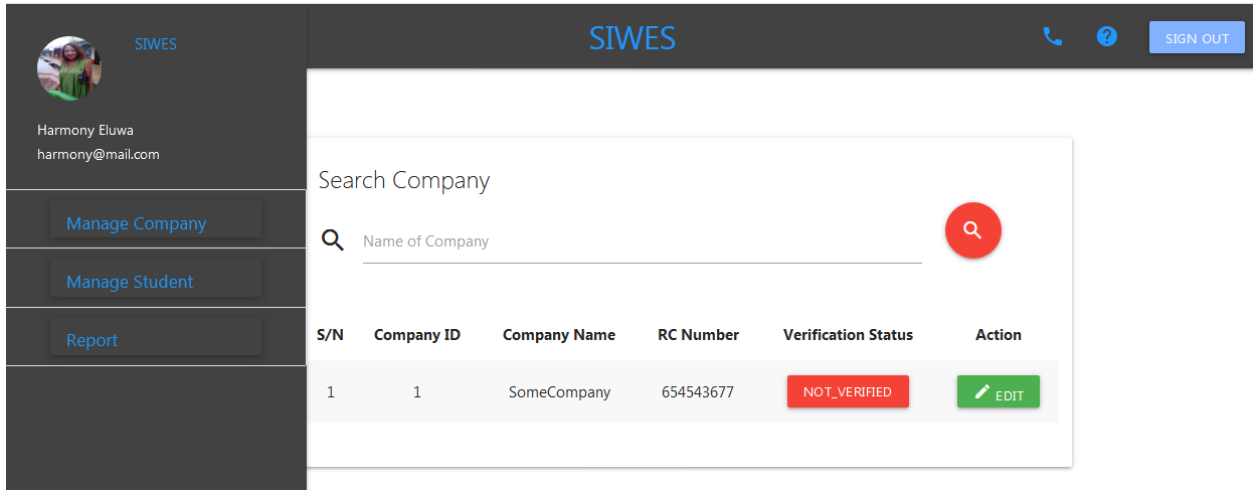


Figure 24 View All Companies Page

Clicking the “NOT_VERIFIED” redirects the Nigeria Corporate Affairs Commission web page for verifying companies as shown below.



Figure 25 Nigeria Corporate Affairs Commission Company Search Page

After checking from the CAC website, if the company has been duly registered, the administrator can click on the “Edit” Button to change the verification status of the employer as shown below.

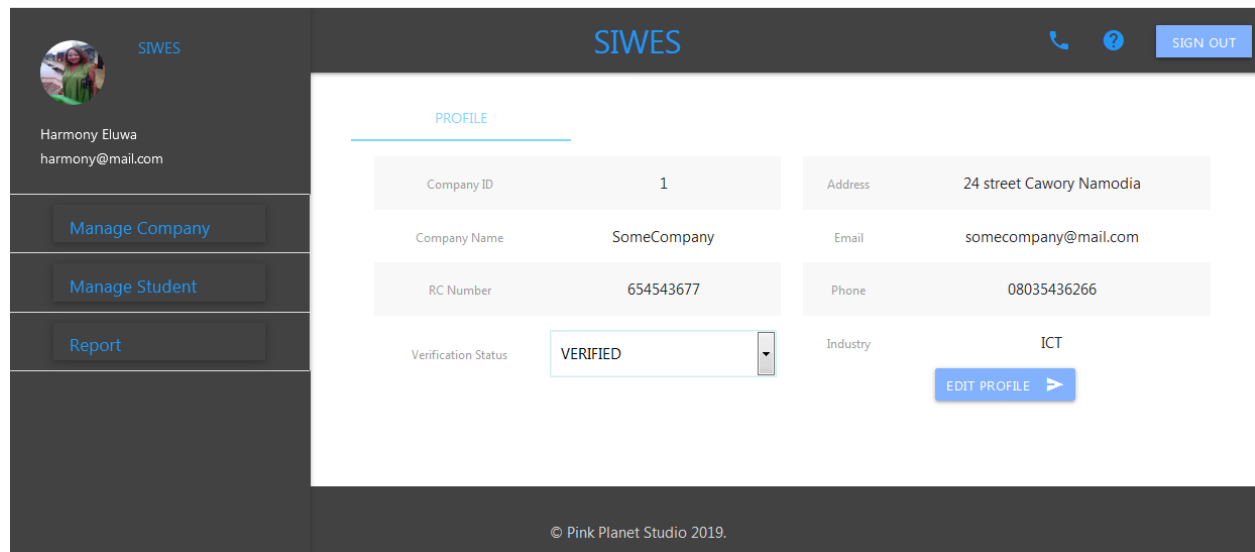


Figure 26 Change Verification Status Page

After verification, the employer will now be able to sign in successfully and carry out other functions described in the following section.

4.2.11 Create Vacancy Page

After successful sign in, the employer will be able to create vacancy of openings in their establishment by clicking on the “Create Vacancy” link in the left navigation bar and the form below will be presented.



Figure 27 Create Vacancy Page

4.2.12 View Applications Page

This page is used by the employer to see who has applied for the job they posted and they can also invite the student for an interview by clicking the “Invite” button.

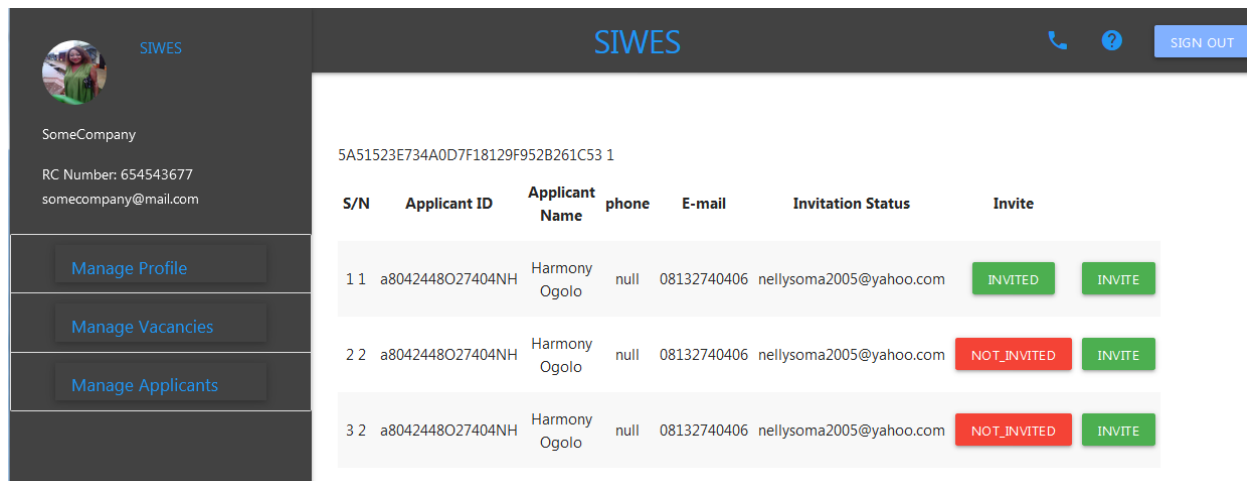


Figure 28 View Applications Page

4.2.13 View All Students Page

This page is used by the administrator to view the list of all the students registered in the SIWES portal, from here, the administrator can view some details about the student including their Jamb registration number and their Jamb examination type which will be used to verify the student as shown below.

S/N	ID	Name	E-Mail	Reg Number	Exam Type	Status	Action
1	1	Harmony Ogolo	nellysoma2005@yahoo.com	2324342	2019 Unified Tertiary Matriculation (UTME)	NOT_VERIFIED	EDIT

Figure 29 View Profile Page (Student)

Students who haven't been verified have a red button labelled "NOT_VERIFIED" against their name. to verify a student, we click on the red button and redirected to Jamb portal where we enter the jamb registration number and the exam type as shown below.

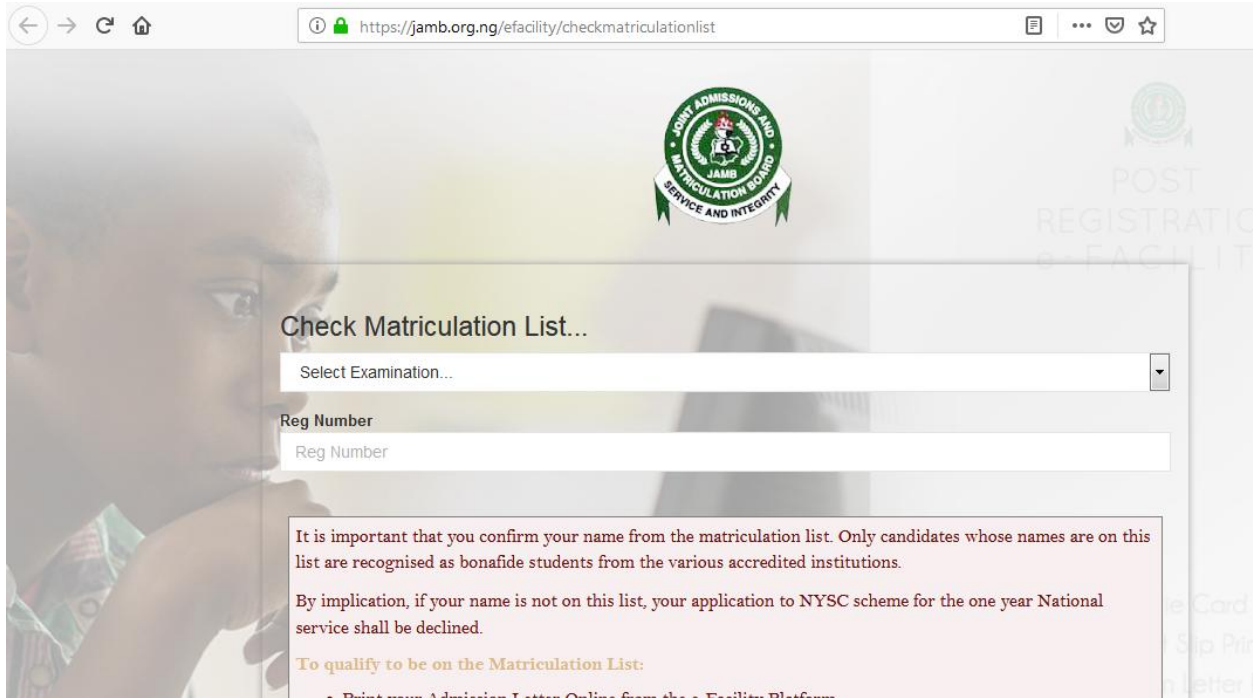


Figure 30 Jamb Students Verification Page

If the students detail is in jamb, then we can go ahead and change the student’s verification status to verified by clicking the green “Edit” button and the edit student page is displayed as shown below

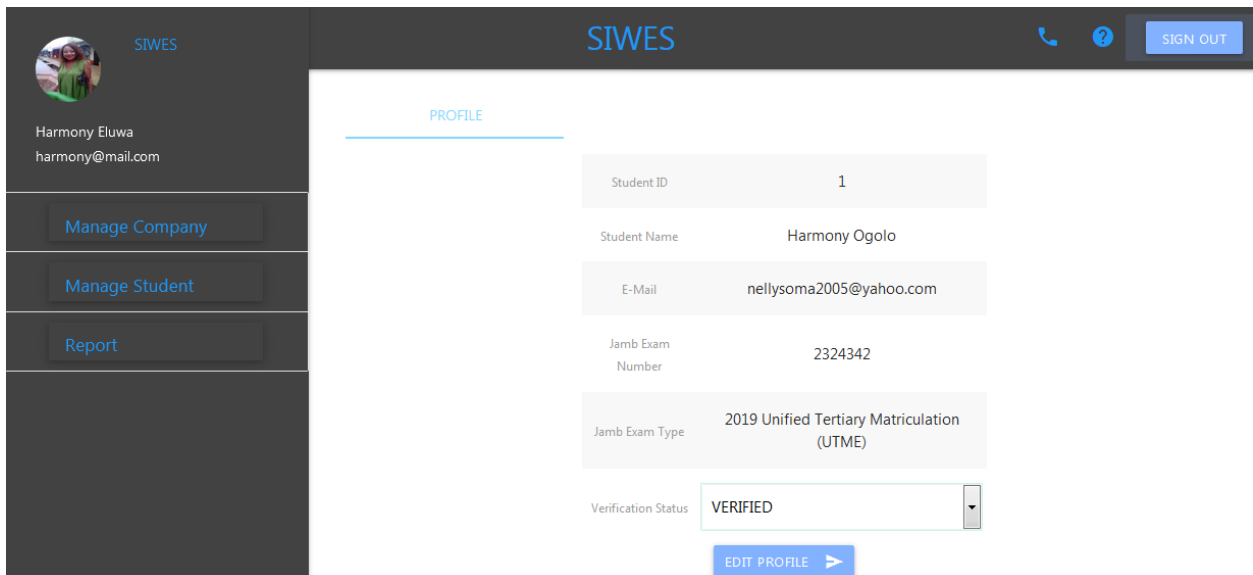


Figure 31 Edit Students Verification Page

4.3 HOW TO INSTALL THE PROGRAM

To successfully install the system on the sever follow these easy steps,

1. Ensure that the server meets the above software and hardware requirements.
2. The software will be built in to a .war file, copy the .war file into the web app's folder of the apache tomcat folder.
3. The complementing database will be distributed as .sql which contains all the tables.

Create a database called siwes and import the .sql file.

4.4 HOW TO RUN THE PROGRAM

1. Startup Tomcat by running the file C:\jakarta-tomcat-5.0.28\bin\startup.bat
2. Open web browser and enter the address: <http://localhost:8080/siwes> to start

4.5 TYPE OF PROGRAMMING LANGUAGE USED

Java Server Pages (JSP) is a Java-based technology that is run on a server to facilitate the processing of Web-based requests. It helps software developers create dynamically generated web pages based on HTML, Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlets at runtime; each JSP servlet is cached and re-used until the original JSP is modified.

JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, with the resulting page being compiled and executed on the server to deliver a document.

4.6 REASONS FOR CHOOSING JSP

JSP has a number of advantages over many of its alternatives. The reason for choosing JSP as a development platform is as follows:

1. JSP programming environment provides the separation between presentation logic and business logic.
2. JSP programming environment provides parallel development of web applications.
3. JSP programming environment provides page compilation automatically.
4. JSP environment provides implicit/global exception handling mechanism.
5. Web authors can change and edit the fixed template portions of pages without affecting the application logic. Similarly, developers can make logic changes at the component level without editing the individual pages that use the logic.
6. The developer can embed the java code directly into HTML code in JSP.
7. JSP uses HTTP protocol as default request/response for communication and thus making it as most prominent web enabling technology.

CHAPTER FIVE

SUMMARY RECOMMENDATION AND CONCLUSION

5.0 SUMMARY

In this project, we develop a web application to assist students searching for Industrial Training Placement to easily get information of available ITF compliant organizations view their profile and send applications from the comfort of their homes.

The application is designed using mainly the Java technology, it is designed as a web application so it can run on browsers, as a result, the user interfaces are developed using web technologies such as HTML5, CSS3 and JavaScript. The application logic is developed using Java beans and Java Servlet. The application is designed using the Model-View-Controller approach (MVC). The data persistent layer is maintained using the MySQL relational database.

5.1 RECOMMENDATION

After the full implementation of the system using 300 level, Computer Science Department as the case study, it is therefore recommended that;

1. Industries, the government and tertiary institutions should adopt the system.
2. Industries, government officials, university SIWES department personnel and students should be trained on how to use the system.
3. Students should prepare afore time for SIWES and get all the required equipment ready
4. The school authorities should liaise with ITF and the establishments where the students are to serve to ensure that the system is adopted.
5. Finally, all the bodies involved in the management of the scheme should liaise with one another and ensure that the system is adequately funded, as one of the reasons for

inefficient supervision as mentioned by one of the authorities was inadequate funding of the program.

5.2 FUTURE WORK

Due to the limited time involved in developing this project work, some key features could not be integrated, it is my recommendation that in future work, the following features be added.

1. Efforts should be made to acquire APIs from Co-operate Affairs Commission of Nigeria to enable automatic verification of companies.
2. Efforts should be made to acquire APIs from the Joint Admission and Matriculation Board of Nigeria to enable automatic verification of students.
3. Effort should be made to develop a mobile version of the system for the IOS and Android platforms.

5.3 CONCLUSION

The students industrial work experience scheme (SIWES) is an amazing program and a welcome development to the Nigerian tertiary education system especially for science and engineering students as it plays a major role in bridging the gap between chalkboard theories and real-life applications of those theories. Since its enactment in 1973, it has aided in reducing the problem of lack of adequate practical skills and has aided in preparing young graduates for employment in industries. I have in these six months been exposed to industrial based skills necessary for a smooth transition from the classroom to the world of works. It has granted me an opportunity to be familiarized and exposed to the needed experiences in practical software development and I have so far gained knowledge and skills which are usually not available within the confines of classrooms.

REFERENCE

- Dangana, J. A., Aimufua, G., Emmanuel, C. S., & Mohammed, Y. (2023). Development of SIWES management system for Nasarawa State University, Keffi: A web-based approach. *International Journal of Research and Innovation in Applied Science (IJRIAS)*, 8(3), 76–85. <https://rsisinternational.org/journals/ijrias/articles/development-of-siwes-management-system-for-nasarawa-state-university-keffi-a-web-based-approach/>
- Ferdian, A. P., Riana, M. S., & Yung-Tsan, J. (2021). A framework of university internship information system with web-based design analysis. In *Proceedings of the Second Asia Pacific International Conference on Industrial Engineering and Operations Management* (pp. 876–884). Surakarta, Indonesia, September 14–16. IEOM Society International. <https://ieomsociety.org/proceedings/2021indonesia/>
- Hamam, M. H. H., Ahmad, S. Z., & Lim Seow Tiap, J. (2025). Web-based internship recruitment system with integration of instant notification service for university students. *Journal of Computing Research and Innovation*, 10(1), 25–39.
- Hasti, N., Lesari, S., & Gustiana, I. (2019). Web-based internship information system. *IOP Conference Series: Materials Science and Engineering*, 662(2), 022090. IOP Publishing. <https://doi.org/10.1088/1757-899X/662/2/022090>
- Oghenekaro, A. O. (2017). Medical internship in Nigeria: Expectations, challenges, and recommendations. *Medical World Nigeria*. Retrieved from <https://medicalworldnigeria.com/post/medical-internship-in-nigeria-expectations-challenges-and-recommendations>
- Ojo, O. (2022, December 19). Challenges of undergraduate internship for Nigerian law students. *Vanguard News*. <https://www.vanguardngr.com/2022/12/challenges-of-undergraduate-internship-for-nigerian-law-students/>
- Olasehinde, O. O., Abiona, A. A., Johnson, O. V., & Omilegan, D. (2022). Design and implementation of a web-based internship placement recommendation system: A case study of Federal Polytechnic Ile-Oluji, Nigeria. *Journal of Applied Information Systems*, 14(4), 52–60. <https://www.researchgate.net/publication/359403105>
- Oyeniya, J. O., Adeniyi, R. A., & Lawal, K. A. (2021). Challenges of Student Industrial Work Experience Scheme (SIWES) implementation in Nigeria: Issues and prospects. *Journal of Higher Education Research*, 9(2), 44–52. <https://journals.heran.org/index.php/JHER/article/download/447/401/785>
- Saint, C. (2025). 8 SIWES challenges in Nigeria and applicable solutions. *SIWES Beginner*. <https://siwesbeginner.com/siwes-challenges-and-solutions>
- Ukozor, F. I. (2022). Science and technology growth and development in Nigeria. *Citadel Journal*, 1(3), 18–29.

Vivian, K. A., & Chinedu, D. O. (2019). Influence of Students' Industrial Work Experience Scheme (SIWES) in enhancing employable skills of business education students in federal tertiary institutions in Anambra State. *Multidisciplinary Journal of Education, Research and Development*, 3(1), 40–49.

APPENDIX A

Home.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>

    <jsp:include page="meta.jsp"/>
    <title>Home</title>
    <jsp:include page="css.jsp"/>

  </head>
  <body>
    <jsp:include page="navbar.jsp"/>
    <div class="section grey darken-3">
      <div class="container">
        <h1><p class="center-align blue-text accent-1">SIWES</p></h1>
        <!--<h5><p class="center-align white-text"><i class="material-icons" style="color:
#00C853;">hdr_strong</i> Find a teacher <i class="material-icons" style="color:
#00C853;">hdr_strong</i> Get a job</p></h5>-->
        <div class="row">
          <a href="createApplicantAccount.jsp" id="download-button" class="btn-large
waves-effect waves-light blue accent-1 pulse col s12 m5">Student</a>
          <!--<hr class="col s12 m6 offset-m3">-->
          <div class="col m2"></div>
          <a href="createCompanyAccount.jsp" id="download-button" class="btn-large
waves-effect waves-light blue accent-1 pulse col s12 m5">Employer</a>
        </div>
        <div class="row center">
          <a href="signin.jsp" id="download-button" class="btn-large waves-effect waves-
light blue accent-1 pulse">Sign In</a>
        </div>
      </div>
    <br/>
    <br/>
    <br/>
  </div>

  <jsp:include page="footer.jsp"/>
  <jsp:include page="js.jsp"/>
  <script>
    $(document).ready(function () {
      $('.sidenav').sidenav();
    });
  </script>
```

```

    </body>
</html>
Navbar.jsp
<div class="navbar-fixed">
    <nav>
        <div class="nav-wrapper grey darken-3" role="navigation">
            <a href="index.jsp" class="brand-logo blue-text accent-1">SIWES</a>
            <a href="#" data-target="mobile-demo" class="sidenav-trigger"><i class="material-
icons">menu</i></a>
            <ul class="right hide-on-med-and-down">
                <li><a href="#"><i class="material-icons blue-text accent-1">call</i></a></li>
                <li><a href="#"><i class="material-icons blue-text accent-1">help</i></a></li>
            </ul>
        </div>
    </nav>
</div>

```

```

<ul class="sidenav" id="mobile-demo">
    <li><a href="#"><i class="material-icons blue-text accent-1">email</i></a></li>
    <li><a href="#"><i class="material-icons blue-text accent-1">help</i></a></li>
</ul>

```

DbConnection.java

```
package dao;
```

```

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import model.Applicant;
import model.Application;
import model.Company;
import model.Vacancy;
public class DbConnection {

    private static final String DATABASE_URL = "jdbc:mysql://localhost/siwes";
    private static final String USERNAME = "root";

```

```
private static final String PASSWORD = "root";
private static Connection connection = null;//manage connection
```

```
private Statement selectUser = null;
private Statement selectApplicantData = null;
private Statement selectApplicantAcademicData = null;
private Statement selectApplicantCertificationData = null;
private Statement selectCompanyUser = null;
private Statement selectAdminUser = null;
private Statement checkApplicantUsernameEntry = null;
private Statement checkCompanyUsernameEntry = null;
private Statement checkAdminUsernameEntry = null;
private Statement viewAllCompany = null;
private Statement searchCompanyByKeyword = null;
private Statement selectComapnyData = null;
private Statement viewAllVacancy = null;
private Statement viewAllJobsApplied = null;
private Statement viewAllVacancyByCompany = null;
private Statement viewApplicationsByCompany = null;
private Statement getStudentVerificatioStatus = null;
private Statement viewAllApplicants = null;
```

```
private PreparedStatement createApplicantAccount = null;
private PreparedStatement createUserAccount = null;
private PreparedStatement editApplicantProfile = null;
private PreparedStatement editApplicantProfileAcademicRecord = null;
private PreparedStatement editApplicantCertification = null;
private PreparedStatement registerApplicantIDForAcademicRecord = null;
private PreparedStatement registerApplicantIDForCertification = null;
private PreparedStatement editApplicantPassport = null;
private PreparedStatement createCompanyAccount = null;
private PreparedStatement createCompanyUserAccount = null;
private PreparedStatement updateCompany = null;
private PreparedStatement editCompanyProfile = null;
private PreparedStatement createNewVacancy = null;
private PreparedStatement apply = null;
private PreparedStatement invite = null;
private PreparedStatement updateStudent = null;
```

```
private ResultSet selectUserResult = null;
private ResultSet selectApplicantDataResult = null;
private ResultSet selectApplicantAcademicDataResult = null;
private ResultSet selectApplicantCertificationDataResult = null;
private ResultSet selectCompanyUserResult = null;
```

```

private ResultSet selectAdminUserResult = null;
private ResultSet checkApplicantUsernameResultEntry = null;
private ResultSet checkCompanyUsernameResultEntry = null;
private ResultSet checkAdminUsernameResultEntry = null;
private ResultSet viewAllCompanyResult = null;
private ResultSet searchCompanyByKeywordResult = null;
private ResultSet selectCompanyDataResult = null;
private ResultSet viewAllVacancyResult = null;
private ResultSet viewAllJobsAppliedResult = null;
private ResultSet viewAllVacancyByCompanyResult = null;
private ResultSet viewApplicationsByCompanyResult = null;
private ResultSet getStudentVerificationStatusResult = null;
private ResultSet viewAllApplicantResult = null;

```

```

FileInputStream fis1;
FileInputStream fis2;
FileInputStream fis3;

```

```

public static void connectToDataBase() throws ClassNotFoundException {

    try {

        // Load (and therefore register) the Driver
        Class.forName("com.mysql.jdbc.Driver");

        // Get a Connection to the database
        connection = DriverManager.getConnection(DATABASE_URL, USERNAME,
PASSWORD);

    } catch (SQLException error) {

        System.out.print(error.getMessage());
        error.printStackTrace();
        System.out.print("What is trying to say here is that, it could not connect to the database");

    }

}

private void closeConnection(Connection connection) {
    if (connection == null) {
        return;
    }
    try {
        connection.close();
    } catch (SQLException ex) {
    }
}

```

```

    }

    public void createUserAccount(String id, String first_name, String last_name, String phone,
String email,
        String password, String role) throws ClassNotFoundException, FileNotFoundException,
SQLException {

    try {

        connectToDataBase();

        createUserAccount = connection.prepareStatement("INSERT INTO
user(id,first_name,last_name,"
            + "phone,email,password,role)VALUES(?,?,?,?,?,?,?)");

        createUserAccount.setString(1, id);
        createUserAccount.setString(2, first_name);
        createUserAccount.setString(3, last_name);
        createUserAccount.setString(4, phone);
        createUserAccount.setString(5, email);
        createUserAccount.setString(6, password);
        createUserAccount.setString(7, role);

        createUserAccount.executeUpdate();

    } catch (SQLException error) {

        error.printStackTrace();
        System.out.print(error);
    } finally {
        closeConnection(connection);
    }
}

    public void createApplicantAccount(String id, String first_name, String last_name,String
mat_no,
        String institute_name,String faculty,String department,String email,String phone,String
password,
        String jamb_exam_type,String jamb_exam_number) throws ClassNotFoundException,
FileNotFoundException, SQLException {

        String studentName = first_name + " " + last_name;
        String role = "applicant";

    try {

```

```

        connectToDataBase();

        createApplicantAccount = connection.prepareStatement("INSERT INTO
student(applicant_id_number,student_name,"
        +
"mat_no,institute_name,faculty,department,email,phone,password,jamb_exam_type,jamb_exam
_number,role)VALUES(?,?,?,?,?,?,?,?,?,?,?)");

        createApplicantAccount.setString(1, id);
        createApplicantAccount.setString(2, studentName);
        createApplicantAccount.setString(3, mat_no);
        createApplicantAccount.setString(4, institute_name);
        createApplicantAccount.setString(5, faculty);
        createApplicantAccount.setString(6, department);
        createApplicantAccount.setString(7, email);
        createApplicantAccount.setString(8, phone);
        createApplicantAccount.setString(9, password);
        createApplicantAccount.setString(10, jamb_exam_type);
        createApplicantAccount.setString(11, jamb_exam_number);
        createApplicantAccount.setString(12, role);

        createApplicantAccount.executeUpdate();
    } catch (SQLException error) {

        error.printStackTrace();
        System.out.print(error);
    } finally {
        closeConnection(connection);
    }
}

private void registerApplicantIDForAcademicRecord(String id) throws
ClassNotFoundException, FileNotFoundException, SQLException {

    try {

        connectToDataBase();

        registerApplicantIDForAcademicRecord = connection.prepareStatement("INSERT INTO
applicant_academic_record(applicant_id)VALUES(?)");

        registerApplicantIDForAcademicRecord.setString(1, id);

        registerApplicantIDForAcademicRecord.executeUpdate();
    } catch (SQLException error) {

```

```

        error.printStackTrace();
        System.out.print(error);
    } finally {
        closeConnection(connection);
    }
}

```

```

private void registerApplicantIDForCertification(String id) throws ClassNotFoundException,
FileNotFoundException, SQLException {

```

```

    try {

```

```

        connectToDataBase();

```

```

        registerApplicantIDForCertification = connection.prepareStatement("INSERT INTO
applicant_certification(applicant_id)VALUES(?)");

```

```

        registerApplicantIDForCertification.setString(1, id);

```

```

        registerApplicantIDForCertification.executeUpdate();

```

```

    } catch (SQLException error) {

```

```

        error.printStackTrace();

```

```

        System.out.print(error);

```

```

    } finally {

```

```

        closeConnection(connection);

```

```

    }

```

```

}

```

```

public String[] userLogin(String username, String password) throws ClassNotFoundException
{

```

```

    String firstName = "";

```

```

    String lastName = "";

```

```

    String phone = "";

```

```

    String role = "";

```

```

    String username1 = "";

```

```

    try {

```

```

        connectToDataBase();

```

```

        String sql = "SELECT * FROM user WHERE email = " + username + """;

```

```

        selectUser = connection.createStatement();

```

```

selectUserResult = selectUser.executeQuery(sql);

while (selectUserResult.next()) {
    username1 = selectUserResult.getString("email");
    password = selectUserResult.getString("password");
    firstName = selectUserResult.getString("first_name");
    lastName = selectUserResult.getString("last_name");

    phone = selectUserResult.getString("phone");
    role = selectUserResult.getString("role");
    //verificationStatus = selectUserResult.getString("veerification_status");
}

} catch (SQLException error) {

    error.printStackTrace();
    System.out.print(error);
} finally {
    closeConnection(connection);
}
return new String[]{password, firstName, lastName, username1, role};
}

public String getStudentVerificatioStatus(String username) throws ClassNotFoundException{

    String verificationStatus = "";

    try{

        connectToDataBase();

        String sql = "SELECT verification_status FROM student WHERE email = " + username
+ """;

        getStudentVerificatioStatus = connection.createStatement();

        getStudentVerificatioStatusResult = getStudentVerificatioStatus.executeQuery(sql);

        while (getStudentVerificatioStatusResult.next()) {

            verificationStatus = getStudentVerificatioStatusResult.getString("verification_status");
        }

    } catch (SQLException error) {

```

```

        error.printStackTrace();
        System.out.print(error);
    } finally {
        closeConnection(connection);
    }

    return verificationStatus;
}

public String[] viewApplicantInformation(String email) throws ClassNotFoundException{

    Applicant ap = new Applicant();

    String applicant_id_number = "";
    String first_name = "";
    String last_name = "";
    String dob = "";
    String gender = "";
    String address = "";
    String city = "";
    String state = "";
    String phone = "";
    String passport = "";
    String applicant_id = "";

    try {

        String sql = "SELECT * FROM student where email = '"+email+"'";

        connectToDataBase();

        selectApplicantData = connection.createStatement();

        selectApplicantDataResult = selectApplicantData.executeQuery(sql);

        selectApplicantDataResult.next();

        applicant_id_number =
ap.setApplicantIdNumber(selectApplicantDataResult.getString("applicant_id_number"));
        first_name = ap.setFirstName(selectApplicantDataResult.getString("first_name"));
        last_name = ap.setLastName(selectApplicantDataResult.getString("last_name"));
        dob = ap.setDob(selectApplicantDataResult.getString("dob"));
        gender = ap.setGender(selectApplicantDataResult.getString("gender"));
        address = ap.setAddress(selectApplicantDataResult.getString("address"));

```

```

        city = ap.setCity(selectApplicantDataResult.getString("city"));
        state = ap.setState(selectApplicantDataResult.getString("state"));
        phone = ap.setPhone(selectApplicantDataResult.getString("phone"));
        passport = ap.setPassport(selectApplicantDataResult.getString("passport"));
        applicant_id = ap.setApplicantId(selectApplicantDataResult.getString("applicant_id"));

    } catch (SQLException ex) {
        ex.printStackTrace();
    } finally {
        closeConnection(connection);
    }

    return new
String[] {applicant_id_number,first_name,last_name,dob,gender,address,city,state,phone,email,p
assport,applicant_id};
}

public String[] viewApplicantAcademicInformation(String email) throws
ClassNotFoundException{

    Applicant ap = new Applicant();

    String institution_name = "";
    String year_of_graduation = "";
    String qualification = "";
    String certificate = "";

    try {

        String sql = "SELECT * FROM applicant_academic_record where email = '"+email+"'";

        connectToDataBase();

        selectApplicantAcademicData = connection.createStatement();

        selectApplicantAcademicDataResult =
selectApplicantAcademicData.executeQuery(sql);

        selectApplicantAcademicDataResult.next();

        institution_name =
ap.setApplicantIdNumber(selectApplicantAcademicDataResult.getString("institution_name"));

```

```

        year_of_graduation =
ap.setFirstName(selectApplicantAcademicDataResult.getString("year_of_graduation"));
        qualification =
ap.setLastName(selectApplicantAcademicDataResult.getString("qualification"));
        certificate = ap.setDob(selectApplicantAcademicDataResult.getString("certificate"));

    } catch (SQLException ex) {
        ex.printStackTrace();
    } finally {
        closeConnection(connection);
    }

    return new String[]{institution_name,year_of_graduation,qualification,certificate};
}

public String[] viewApplicantCertification(String email) throws ClassNotFoundException{

    Applicant ap = new Applicant();

    String cert_name = "";
    String cert_file = "";

    try {

        String sql = "SELECT * FROM applicant_certification where email = '"+email+"'";

        connectToDataBase();

        selectApplicantCertificationData = connection.createStatement();

        selectApplicantCertificationDataResult =
selectApplicantCertificationData.executeQuery(sql);

        selectApplicantCertificationDataResult.next();

        cert_name =
ap.setApplicantIdNumber(selectApplicantCertificationDataResult.getString("cert_name"));
        cert_file =
ap.setFirstName(selectApplicantCertificationDataResult.getString("cert_file"));

    } catch (SQLException ex) {
        ex.printStackTrace();
    } finally {
        closeConnection(connection);
    }
}

```

```

    return new String[]{cert_name,cert_file};
}

public void editApplicantProfile(String applicant_id, String first_name, String last_name,
    String dob,String gender,String address,String city,String state,String phone,String
email){

    try{
        connectToDataBase();
        String sql = "UPDATE applicant SET first_name = '"+first_name+"',"
            + "last_name = '"+last_name+"",dob = '"+dob+"",gender = '"+gender+"",address =
'''+address+''''
            + ",city = '"+city+"",state = '"+state+"",phone = '"+phone+"",email = '"+email+"
WHERE applicant_id_number = '"+applicant_id+" ";
        editApplicantProfile = connection.prepareStatement(sql);
        editApplicantProfile.executeUpdate();
    }catch(ClassNotFoundException | SQLException error){

        error.printStackTrace();
    }

    public void editApplicantProfileAcademicRecord(String applicant_id, String
institution_name, String year_of_graduation,
        String qualification,String certificate) throws FileNotFoundException{

        try{
            connectToDataBase();
            File file = new File(certificate);

            String sql = "UPDATE applicant_academic_record SET certificate = ?, institution_name
= '"+institution_name+"',"
                + "year_of_graduation = '"+year_of_graduation+"",qualification =
'''+qualification+'''+WHERE applicant_id = '"+applicant_id+" ";

            editApplicantProfileAcademicRecord = connection.prepareStatement(sql);

            fis1 = new FileInputStream(certificate);

            editApplicantProfileAcademicRecord.setBinaryStream(1, (InputStream) fis1, (int)
(file.length()));

            editApplicantProfileAcademicRecord.executeUpdate();

        }catch(ClassNotFoundException | SQLException error){

```

```

        error.printStackTrace();
    }
}

public void editApplicantCertification(String applicant_id, String cert_name, String cert_file)
throws FileNotFoundException{

    try{
        connectToDataBase();
        File file = new File(cert_file);

        String sql = "UPDATE applicant_certification SET cert_file = ?, cert_name =
"+cert_name+"WHERE applicant_id = "+applicant_id+" ";

        editApplicantCertification = connection.prepareStatement(sql);

        fis2 = new FileInputStream(cert_file);

        editApplicantCertification.setBinaryStream(1, (InputStream) fis2, (int) (file.length()));

        editApplicantCertification.executeUpdate();

    }catch(ClassNotFoundException | SQLException error){

        error.printStackTrace();
    }
}

public void editApplicantPassport(String applicant_id, String passport) throws
FileNotFoundException{

    try{
        connectToDataBase();
        File file = new File(passport);

        String sql = "UPDATE applicant SET passport = ? WHERE applicant_id_number =
"+applicant_id+" ";

        editApplicantPassport = connection.prepareStatement(sql);

        fis3 = new FileInputStream(passport);

        editApplicantPassport.setBinaryStream(1, (InputStream) fis3, (int) (file.length()));

        editApplicantPassport.executeUpdate();
    }
}

```

```
}catch(ClassNotFoundException | SQLException error){  
    error.printStackTrace();  
}  
}  
}
```